

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

Digitally-controlled LED Rotating Display System

Team #10

CHENTAI YUAN
(chentai2@illinois.edu)

GUANSHUJIE FU
(gf9@illinois.edu)

KEYI SHEN
(keyis2@illinois.edu)

YICHI JIN
(yichij2@illinois.edu)

Supervisor: Prof. Chushan Li
TA: Adeel Ahmed

May 23, 2023

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Functionality	1
1.3	Subsystem Overview	1
1.3.1	Display and Control Subsystem	2
1.3.2	Logic and Interface Subsystem	2
1.3.3	Power and Base-shield Subsystem	3
2	Design	4
2.1	Subsystem Description	4
2.1.1	Display and Control Subsystem	4
2.1.2	Logic and Interface Subsystem	7
2.1.3	Power Subsystem	10
2.2	Design Justification	12
2.2.1	Motor load analysis	12
2.2.2	Feasibility analysis of image processing	14
2.2.3	Latency analysis of image processing	14
2.2.4	Latency analysis of ESP32 program	15
2.3	Design Alternatives	16
3	Cost & Schedule	17
3.1	Cost	17
3.1.1	Labor	17
3.1.2	Parts	17
3.1.3	Total cost	17
3.2	Schedule	17
4	Conclusion	20
4.1	Accomplishments	20
4.2	Uncertainties	20
4.3	Future Work	21
4.4	Ethical Considerations	21
4.4.1	Ethics	21
4.4.2	Safety	22
	References	23
	Appendix A PCB Design (Partial)	24
	Appendix B Requirements & Verification Table	25
B.1	Display and Control Subsystem	25
B.2	Logic and Interface Subsystem	25
B.3	Power and Base-shield Subsystem	26

Appendix C	Details for Other Selected Parts	29
C.1	ESP32 Pins Configuration	29
C.2	TF card reader	29
C.3	Voltage Stabilizer	31
C.4	Switch/Connector	31
C.5	Motor	31
C.6	LED array	32
C.7	Wireless Charging Module	32

1 Introduction

1.1 Purpose

By applying Persistence of Vision (POV), many spinning Light Emitting Diode (LED) message systems are developed to display messages and images with few LEDs [1]. A higher frequency in displaying an image enhances the resolution of the image [2] and when the frequency is over 30 frames per seconds (FPS) [3], human can observe images with a great quality. With a high speed motor, many existing systems can display messages clearly.

Some limitations of the existing POV display systems on the market are to be improved. First, many of them only support limited patterns of text messages [1] and the images or messages to be displayed are pre-defined and cannot be changed in a real-time manner during the display. Second, the wired connection between some components may limit the rotation of LED array, and reduce the quality of display. Third, many other systems are not compact enough to be easily handled in hand.

Our solution is to develop a personal computer (PC) platform that can realize text, image, and video communication to our system with the help of elaborated image processing technology. Economical wireless communication technologies [4] can be applied to achieve real-time image updates. By applying a wireless charging module, the rotation of the LED array won't be limited by the wired connection. Its revolution speed can be fast and achieve good display quality. We also tried to compose our system meticulously with customized components and integrated circuits (IC) used.

1.2 Functionality

Our system can present text, images, graphics interchange format (gif), and video at a high quality. To realize this, we implemented a frame-wise, real-time communication module. The LED array is made up of 160 full-color mini LEDs that ensures a high resolution. The LED array can rotate at a speed of 1800 revolutions per minute, which allows us to display 30 frames per second. To achieve this, the printed circuit board (PCB) that we embedded our LED array is customized and light-weighted. Thanks to the nicely-designed PCB, our audiences can perceive all the contents smoothly. All our components are thoughtfully chosen. The overall system is light-weighted and portable in size. We can hold it easily while it presents all the amazing patterns.

1.3 Subsystem Overview

Our system contains 3 subsystems: Display & Control Subsystem, Logic & Interface Subsystem and Power & Base-shield Subsystem. The top-level block diagram is shown in Figure 1. This section gives the description of all subsystems and their interactions with other subsystems.

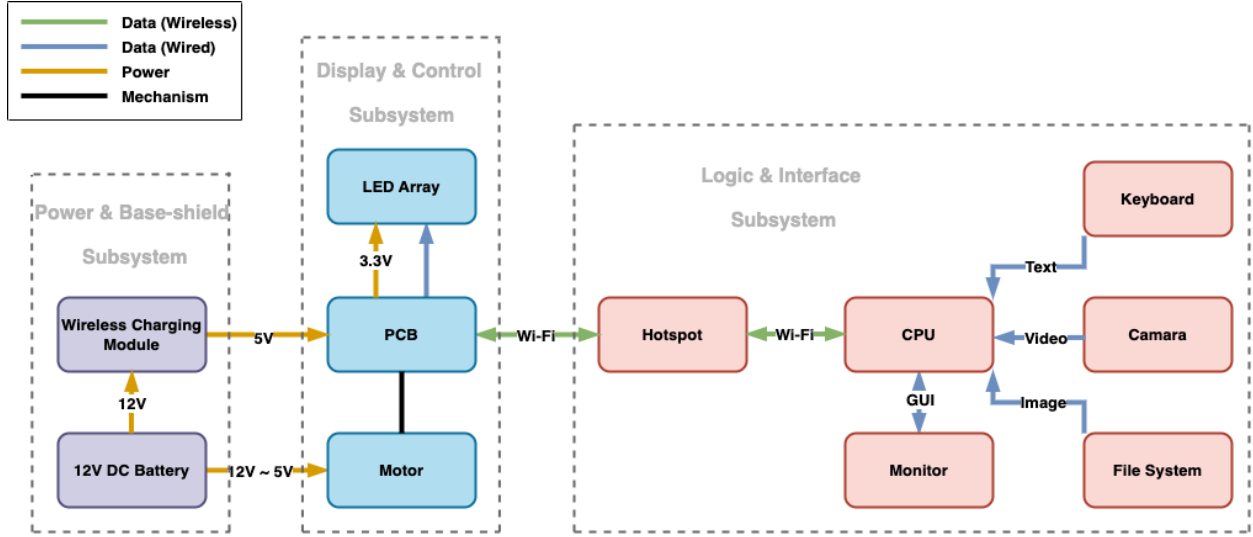


Figure 1: Block Diagram

1.3.1 Display and Control Subsystem

The *Display and Control Subsystem* is responsible for showcasing text, images, and videos on the surface of the rotating PCB. It consists of a motor and a flanged joint with an integrated PCB. The PCB houses 160 LEDs that are strategically distributed along the center line. As the motor rotates, these LEDs work together to display images at a smooth frame rate of 30 frames per second. The resulting visual output forms a captivating disk with a diameter of 170 mm. All the LEDs are meticulously controlled by our main microcontroller, the ESP32-WROOM-32E. It is securely attached to the back of the PCB and receives power from the wireless charging module in *Power and Base-shield Subsystem*. The ESP32 microcontroller not only controls the LEDs but also establishes communication with *Logic and Interface Subsystem*. Through Wi-Fi connectivity, it enables timely and seamless control over the entire system.

1.3.2 Logic and Interface Subsystem

The *Logic and Interface Subsystem* serves as the central component responsible for handling user inputs, commanding the Display and Control Subsystem, and providing feedback to the user. It consists of a general-purpose computer equipped with input peripherals and a monitor for a Graphical User Interface (GUI). Specifically, the computer receives user inputs from peripherals such as keyboards and cameras. It processes various forms of input, including text, images, and videos, with minimal latency. The processed data is then transmitted to the ESP32 microcontroller in the *Display and Control Subsystem* using Wi-Fi communication, allowing for wireless control of the LEDs. Additionally, the *Logic and Interface Subsystem* provides a GUI for visualizing the system's operation and facilitating user interaction. It's worth noting that the wireless communication between the *Logic and Interface Subsystem* and the *Display and Control Subsystem* ensures their physical

independence from each other.

1.3.3 Power and Base-shield Subsystem

The *Power and Base-shield Subsystem* plays a crucial role in supplying power to various components across different subsystems. One key element is the wireless charging module, which provides a 7.5V power supply to the ESP32 microcontroller and LED Array in *Display and Control Subsystem*. Additional details about the wireless charging module will be provided in Section 2.1 and Appendix C. To power the DC motor, a 12V DC battery is utilized, ensuring its smooth operation. Most components we choose are compacted in size, many of which are customized and highly integrated. This subsystem is packed in a 3-D printed shield, that allows our system to be portable in size.

2 Design

2.1 Subsystem Description

2.1.1 Display and Control Subsystem

PCB Design

The PCB serves the purpose of displaying images and videos during rotation. In our initial design version, the PCB incorporates several components, including heavier elements like a Li-PO battery and an HC-05 Bluetooth module. However, through a literature review, we discovered that the ESP32 chip can serve as a suitable replacement for the LED controller, offering built-in Bluetooth/Wi-Fi capabilities. This finding allows us to apply Surface Mount Technology (SMT) to the PCB, reducing its weight to 15 grams. Additionally, a symmetric component placement scheme has been implemented to enhance balance during rotation. A partial view of the PCB sketch can be found in Appendix A.

Surface Mount Technology (SMT) has been implemented for the new PCB design, as shown in the render view of Figure 2. The front face of the PCB features the LED array, Trans Flash (TF) card, and resistors. On the back face of the PCB, you can find the Hall switch and the ESP32 microcontroller. Specifically, we have utilized the ESP32-WROOM-32E variant, which is equipped with a 4 MB Serial Peripheral Interface (SPI) flash. This microcontroller serves as the control unit for the LED array on the PCB. It operates on a 3.3V DC power supply and is capable of receiving data from a Trans Flash card and Bluetooth signals. By running the pre-burned program, the ESP32 converts the input data into control signals for the LED array using a clock signal.

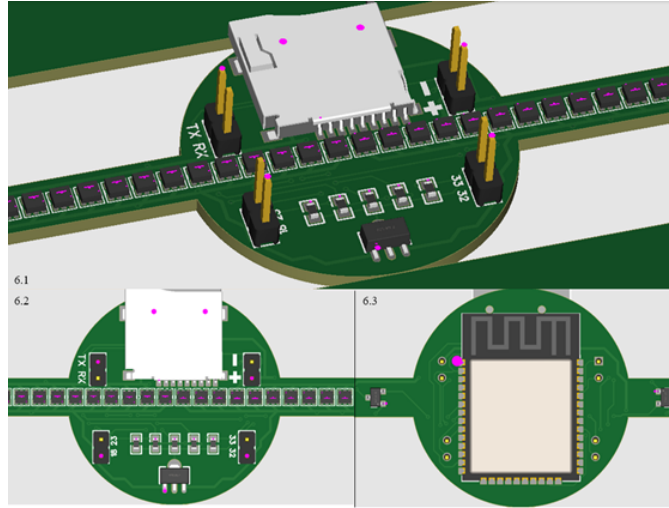


Figure 2: SMT on final version PCB

To fulfill the requirements of LED array control, Wi-Fi functionality, TF card reading, and compact size, we have selected the ESP32 microcontroller. Specifically, we have chosen the ESP32-WROOM-32E variant. This microcontroller will be powered by a +3.3 V source through a voltage stabilizer (AMS1117-3.3). The voltage source will be a wireless voltage

supply, which will be discussed further in the Power Supply section. Details for the pin layout and connection details, please refer to Appendix C, Figure 17 and Table 8, and the embedded design for ESP32 is detailed sketched in Appendix A.

The microcontroller's primary role is to process all input data, which includes data received from the PC through Wi-Fi module communication and data from the TF card. It then outputs the processed data to the LED array. The microcontroller requires 520 KB of Static Random Access Memory (SRAM) to store all the necessary instruction data, which will be pre-burned. The physical dimensions of the ESP32-WROOM-32E on the PCB layout measure 18.0 mm \times 25.5 mm \times 3.1 mm. Details for TF card reader, the Voltage Stabilizer, Switch, Connector and Motor we used in the project is shown in Appendix C.

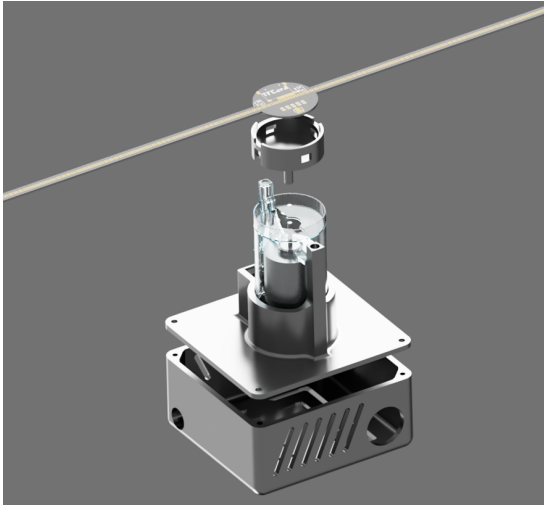


Figure 3: Explosive view of system

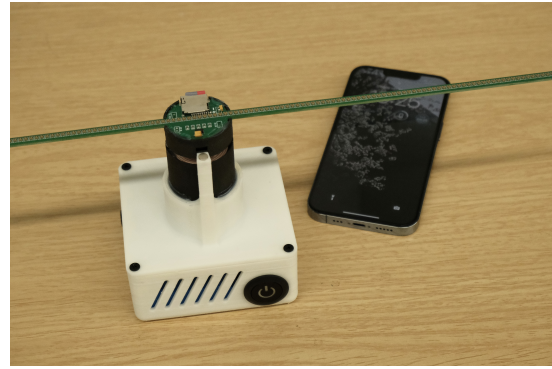


Figure 4: Assembled system

System Assembly

As illustrated in Figure 3, the assembly steps are as follows. Begin by assembling the 385-motor into the motor protector. Next, attach the wireless charging module onto the PCB holder, place the PCB onto the holder and secure it in place. Then proceed to assemble the motor protector and the outer shield together, with the PCB holder attached to the motor shaft. Finally, the assembled system is shown in Figure 4.

ESP32 Program

As the processing unit in the *Display and Control Subsystem*, ESP32 is mainly responsible for receiving converted image data from Wi-Fi connection, decoding to appropriate data format and triggering LED arrays to display. However, as we target at supporting real time video display, we need to guarantee a appropriate latency for the whole process in ESP32. Considering the limited hardware resources, the optimization of our program is necessary.

The optimization scheme primarily focuses on maximizing the utilization of hardware resources. Firstly, since we are utilizing a dual-core ESP32 chip, we can achieve parallel

execution of different tasks. In my design, the *loop()* function will be pinned to *core₀*. Since *loop()* is an infinite loop function required by ESP32, it will be responsible for map pixel color to LED and trigger the LED arrays to display. Another task *receive()* will be assigned to *core₁*. It is a self-defined function for receiving image data from Wi-Fi client, decoding data and storing to local RAM.

The second point involves the design of a local FIFO (First-In-First-Out) buffer to handle video data. Since real-time video data is continuous and can vary in size for different frames, the processing time required for each frame can significantly differ. If certain frames take an extended amount of time to process, it can adversely affect the display quality, resulting in noticeable lags between frames. By implementing a FIFO buffer to store fully processed video data before displaying it, we can manually generate a time slice for processing certain frames without affecting the display.

LED array

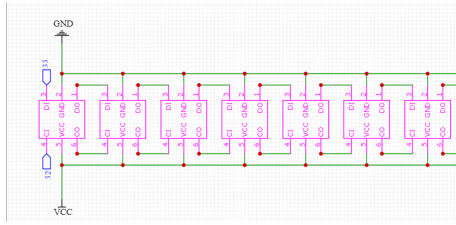


Figure 5: PCB Sketch of LED Array

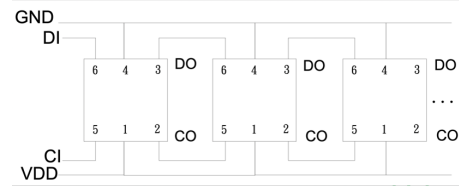


Figure 6: LEDs layouts

Single LED arrangement is detailed described in Appendix C. A series of LEDs will be a connected in a line and their pin connection is shown in Figure 6. To control LED array, our input data signal is arrange in a way shown in Figure 7. As it is shown in Figure 5, for each LED chip, data and clock of input pins is provided by microcontroller, while the data and clock output pins are connected with the next LED input.

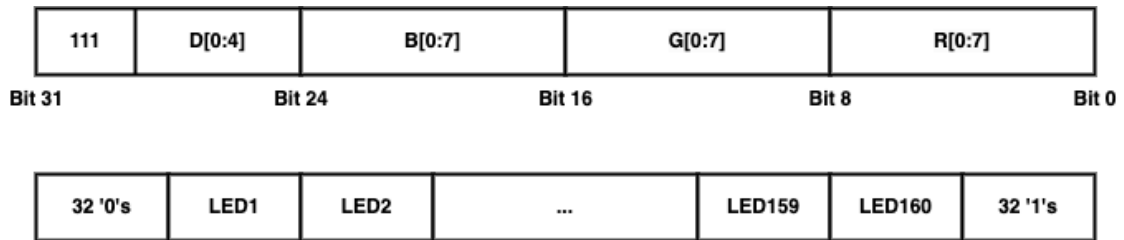


Figure 7: Signal format for a single LED and a serial of LEDs

When all the 160 LEDs shine on and give white light at the highest intensity the power consumption will be

$$160 \times 7.5V \times 18mA = 21.6W. \quad (1)$$

The control signal will consume

$$160 \times 5V \times 0.9mA = 0.72W. \quad (2)$$

Considering power dissipation for all the 2 serials of LEDs which is

$$2 \times 350mW = 0.7W, \quad (3)$$

LED module will consume power up to

$$21.6W + 0.72W + 0.7W = 23.02W \quad (4)$$

For a single LED state, we will need 32 bits of data. Our system will require 30×180 frames per second. The input will also include starting bits and the ending bits. The total transfer rate will be

$$\frac{1}{30}s \times 180 \times (160 + 2) \times 32 \times \frac{1}{8 \times 1024 \times 1024} Mb = 3.337Mb/s. \quad (5)$$

2.1.2 Logic and Interface Subsystem

Image processing

The central processor performs image processing on various types of input, including text, images, and videos, before transmitting the data to the ESP32. Text input from the keyboard is packaged as images that contain the text message before undergoing image processing. Single static images can be processed directly. Recorded videos and live video streams captured by the camera are divided into a series of images, which are processed and transmitted sequentially.

For an image of arbitrary size, our main focus is to apply resizing, rotating, coordinate frame transformation, and brightness correction to convert the original image into a format suitable for display with rotating LEDs. The resulting output image should define the state of each LED within one circumference.

Considering that we have 2 LED strides with 80 LEDs on each stride, and assuming the circumference is evenly divided into 360 pieces, the total number of LED states within one circumference would be 80×360 . Additionally, each LED state includes the values of the RGB channels, resulting in a complete representation of the LED display.

The first resizing operation is relatively straightforward. It resizes the image to a height of 160, which is the total number of LEDs, while maintaining the original image's height-to-width ratio to avoid shape distortion. Let H_i and W_i represent the height and width of the input image, respectively. The output height and width, H_o and W_o , can be defined as follows:

$$W_o = 160 \quad (6)$$

$$H_o = \lceil \frac{W_i}{H_i} 160 \rceil \quad (7)$$

After resizing the image, the next step is to rotate the resized image 90 degrees clockwise. This rotation is done to align the direction of zero degrees in the polar coordinate system and prepare for the subsequent coordinate frame transformation.

To obtain the state of each LED in the polar coordinate frame, we can perform a coordinate frame transformation from the Cartesian coordinates of the rotated image. Let's consider a pixel with position information (x, y) in the rotated image. To convert this pixel position to a polarized position, the following formulas can be utilized:

$$r = \sqrt{x^2 + y^2} \quad (8)$$

$$\phi = \arcsin \frac{y}{r} \quad (9)$$

where r and ϕ are polar radius angle.

To define the coordinate frame transformation, let's assume that H_i and W_i represent the height and width of the input image after rotation. Here, we consider W_i as a fixed value of 160.

The transformation center should be located at the center of the rotated image. In Cartesian coordinates, the position $P = (x, y)$ of the transformation center can be calculated as follows:

$$P = (x, y) = (80, \lceil \frac{H_i}{2} \rceil) \quad (10)$$

To determine the radius R of the bounding circle for the transformation, we take the minimum value between the height and width of the rotated image and divide it by 2. This ensures that the display is filled with valid pixel values, even if there is a significant difference between the height and width of the image. However, it's important to note that when there is a large difference in dimensions, a considerable amount of content may be ignored as it falls outside the bounding circle.

$$R = \min(80, \lceil \frac{H_i}{2} \rceil) \quad (11)$$

The output height and width H_o and W_o are fixed as 360 and 80 as stated before to simplify data access on ESP32.

The last operation is brightness correction. There are two issues regarding brightness on a rotating LED display:

First, when defining the state of LEDs with the same degree of central angles, the density of valid pixels is not evenly distributed along the radius. This is due to the varying rotating radius of the LEDs. In other words, if every LED shines with the same brightness, the center of the display will appear brighter than the edges. This is because there is a longer time interval between two adjacent LEDs passing the same position at the edge, while the center always has a shining LED.

Second, human perception of brightness is not linearly related to the actual physical intensity of light. For instance, a doubling of physical light intensity does not appear to us as a doubling of perceived brightness. However, many imaging systems and display devices, such as computer monitors and cameras, operate based on a linear relationship between the input signal and the output intensity. This can lead to a mismatch between displayed images and human visual perception.

We employ two strategies to address these two issues. Firstly, we utilize a linear transformation to normalize the brightness of LEDs along the rotating radius. To achieve this, we introduce a global brightness factor, denoted as B_g , which corrects the overall brightness, and a central brightness factor, denoted as B_c , which controls the brightness of the display center. Both B_g and B_c are assumed to range from 0 to 1.

Given an initial brightness value, B_i , we calculate the resulting brightness, B_l , using the following equation:

$$B_l = B_g(B_c + \frac{(1 - B_c)r}{R})B_i \quad (12)$$

Here, the constant R represents the number of LEDs per stride (which is 80 in this case), and r denotes the index of the LED on the stride ($0 \leq r < 80$). This equation allows us to adjust the brightness of each LED along the rotating radius based on the global and central brightness factors.

Additionally, we incorporate a nonlinear Gamma correction to further enhance our image processing. After obtaining the brightness value B_l through linear correction, we apply a power function with a parameter γ to the input brightness, resulting in the corrected brightness B_n . The formula for this correction is as follows:

$$B_n = cB_l^\gamma \quad (13)$$

In this equation, c represents a constant, typically set to 1, and γ is a parameter that controls the nonlinear correction. By applying this Gamma correction, we can better match the perceived brightness with the actual physical intensity of light.

In Section 2.2, we will demonstrate the feasibility of our image processing techniques.

Graphical User Interface

Graphical User Interface (GUI) is the front end of our design. It abstracts and hides the complex inner logic in our project, and hence provides a direct method for user to operate. There are two interfaces in our design. One in *Logic and Interface Subsystem* and another in *Display and Control Subsystem*.

Considering our final scheme, the interface in *Logic and Interface Subsystem* should be able to provide the following functionalities.

1. The ability to choose different type of images or videos to upload, convert and transmit to display subsystem.
2. The ability to enter user customized text and transmit to display.
3. The ability to choose between different mode. As we target at supporting text, image, video and real time video display, the interface should provide entry to change the mode.

The interface in *Display and Control Subsystem* should be able to provide the following functionalities.

1. The ability to retrieve, show and choose available medias existing in the ESP32 local storage like RAM and external storage like SD card.
2. The ability to cooperate with interfaces in *Logic and Interface Subsystem*.

2.1.3 Power Subsystem

Outer-Shield

The figure below, Figure 8 and Figure 9, illustrates the basic structural design of Computer-Aided Design (CAD) models for our project. In the Figure 8.1, you can see the motor, which is powered by a 12V power supply. The upper-right sub-figure displays the LED integrated Printed Circuit Board (PCB). All of these components will be securely mounted and protected by an outer shield, as depicted in the Figure 8.3.

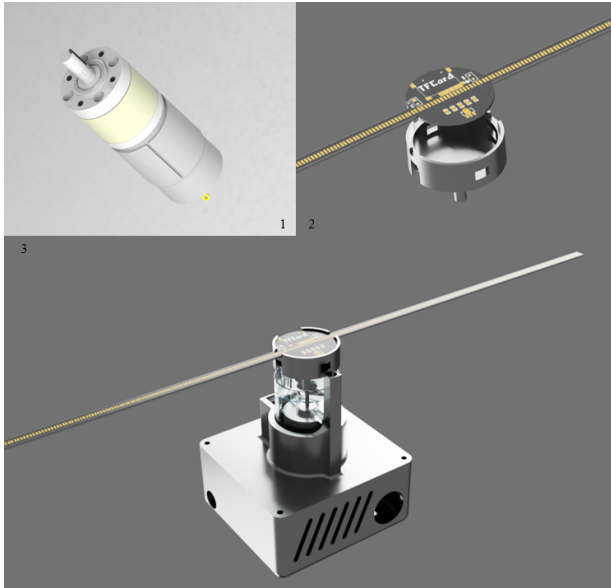


Figure 8: Outer Shield Design Overview

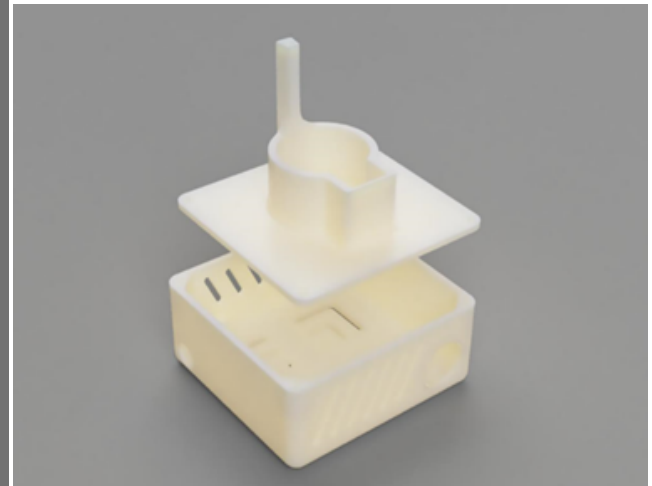


Figure 9: Outer Shield Design Details

The objective of system design is to create an integrated, compact, and wireless LED display system that can establish real-time communication with computer web-page. To achieve this, I have developed a conceptual CAD model with a projected volume of 100 cm^3 . The LED array is capable of displaying real-time images within a circular area with a radius of 16 cm . The outer shield is constructed using 3D printed parts, providing protection for the motor. By carefully arranging the interior space of the shield, we can accommodate a 385 DC motor and a 12 V power supply.

However, having a single PCB mounted directly on the motor without any protective shield poses a risk of damage. To address this concern, I have made modifications to the

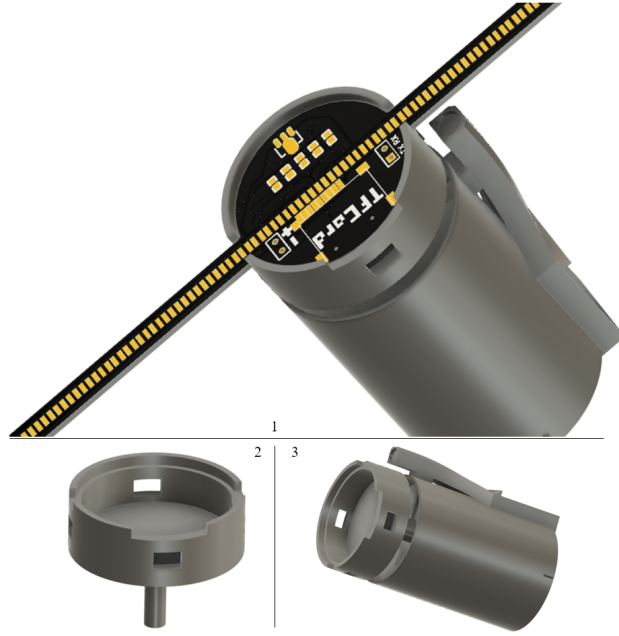


Figure 10: Motor Case and PCB Holder

case design for the 385 motor and the PCB, as shown in Figure 10.1. This updated design ensures a secure and tight fit for both the motor and the PCB, effectively minimizing the impact of vibrations during the display rotation. For a more detailed perspective, Figure 10.2 and Figure 10.3 showcase the specific features of the PCB holder.

Wireless Charging Module

The wireless charging module is used for transfer power from the base part to the upper PCB that carry voltage stabilizer, signal receiving module and the LED array. It will be located at the rotation axis of the base motor part and PCB. The voltage stabilizer requires a current input larger than 700mA. The wireless will supply a current larger than 1000mA in 500ms once it is on. Detailed reference for connectors is shown in Appendix C, Figure 21. The power is transferred from electricity to magnetic field, and then to electricity again, part of the power is dissipated at a rate of 16.7%.

The emitter part will transfer the input DC power to an AC power to its coil and a magnetic field is produced. The module we use includes an integrated protection circuit and a light will shine on if there is anything wrong with the emitter, but we still need to verify that the output is within the desired range when an input at the possible range is given. A $7.5V \pm 0.5V$, $3A \pm 0.5A$ input is what our voltage stabilizer can withstand and a power of 28W input is required to supply to the PCB.

2.2 Design Justification

Regarding to high-level requirements we proposed, we prove functions of critical components through mathematical analysis or numerical simulation. As the visualized and critical part of our system, we do want to do a better visual effect analysis, but it is more subjective and harder to analyze mathematically and simulate. We will improve our analysis after finding more formal materials and do experiments after we get all components.

2.2.1 Motor load analysis

In this analysis, we want to figure out the maximum load for the motor when we set the system vertically and horizontally and keeping a 30 FPS display of images.

With parameter got from datasheet, we know that the motor has a rotation rate of 5000 RPM when no load. And it will be stalled under a torque of 750 g·cm. Based on the relationship between rotation rate and torque, we will have an idealized function:

$$N = -\frac{20}{3}\tau + 5000 \quad (14)$$

As we are aimed at forming a display with image refresh rate larger than 30 fps, which means,

$$N_{aim} \geq 30 \frac{circle}{second} = 1800RPM \quad (15)$$

Constrains has shown below in Figure 11(a).

By substitution, we will have a maximum torque shown below.

$$\tau_{aim} \leq 480g \cdot cm \quad (16)$$

When we place the system vertically, we can calculate the force on the output shaft of motor by the equation,

$$F_{shaft} - mg = F_{centripetal} \quad (17)$$

We have the equation for centripetal force,

$$F_{centripetal} = m \times w^2 \times r \quad (18)$$

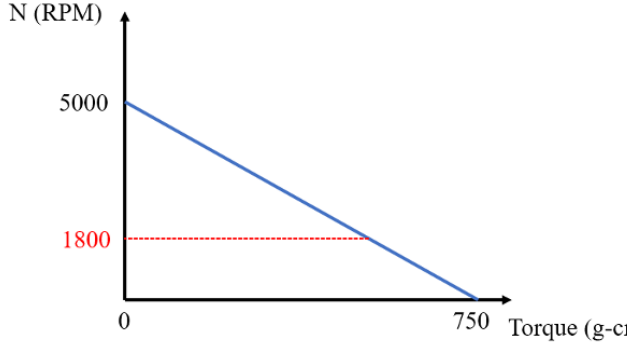
where m is the equivalent weight for whole PCB, w is the angular velocity, and r is the mass offset on PCB.

The angular velocity w and rotation speed N have the relationship,

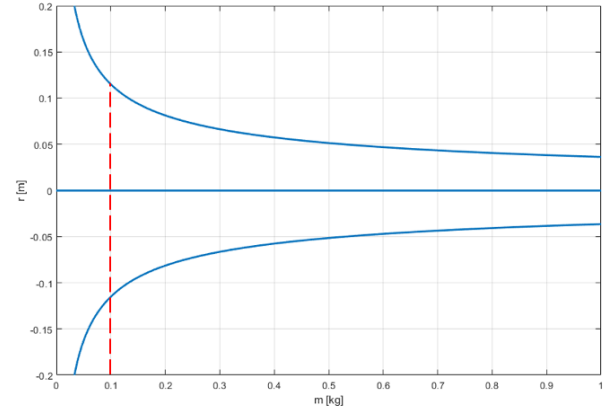
$$w = \frac{N \times \pi}{30} \quad (19)$$

With a minimum rotation speed 1800 RPM, we can calculate the relationship between mass offset on PCB and the PCB weight, shown below.

$$188.55^2 \left[\frac{rad}{s} \right] \times r^2 + 9.81 \left[\frac{N}{kg} \right] \times r - \frac{47[N \times m]}{m} \leq 0 \quad (20)$$



(a) Motor N-Torque Constrains



(b) Relationship between mass offset and the PCB weight

Figure 11: Data Analysis

Also shown on the Figure 11(b) below, where m is PCB weight and r is mass offset on PCB.

For example, we designed the overall PCB weight $m = 100g$, and by substitution, the mass offset on PCB can be calculated (Shown in red line on Figure 11(b)), which is,

$$-0.1151m \leq r_{offset} \leq 0.1148m \quad (21)$$

With a mechanical simulation, Finite Element Analysis is placed on our main PCB. During the rotation, the board will be bearing a torque of $480 \text{ g} \cdot \text{cm}$ for maximum, together with a pressure from all the components, which weight 30 g for maximum. The result is shown in Figure 12 below. The maximum displacement at two edges of the PCB will within 1.5mm , which meets the requirement.

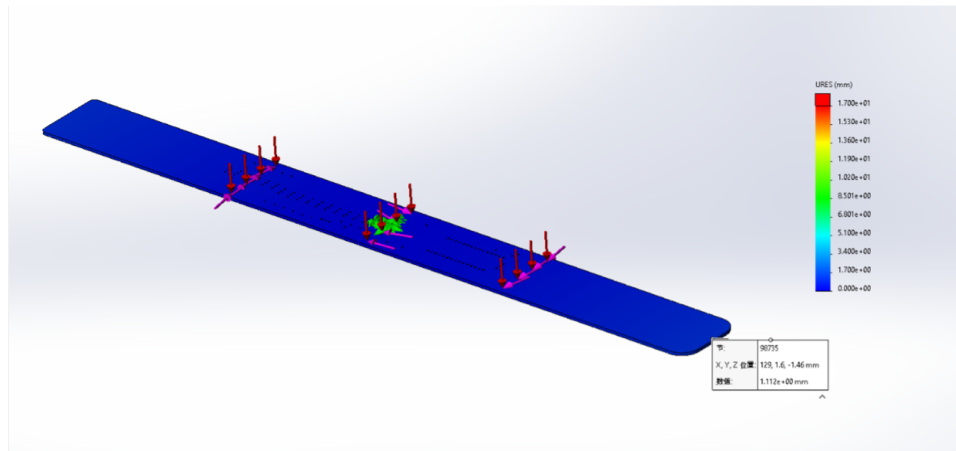


Figure 12: FEA Result for the main PCB

2.2.2 Feasibility analysis of image processing

In order to validate the feasibility of our image processing operations, we have selected a vibrant dynamic image in the form of a .gif file as an example. We have chosen one frame from this image for further explanation. Figure 13 provides the original image, the processed image, and the displayed image for reference.

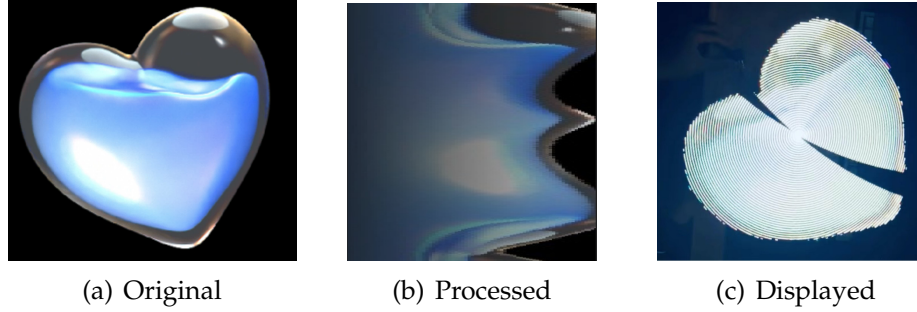


Figure 13: Images Display

Upon examining the processed image in Figure 13(b), it is readily noticeable that the heart pattern from the original image undergoes significant deformation as a result of the coordinate frame transformation from Cartesian to polar coordinates. Additionally, the processed image exhibits a transition from dark to light, gradually increasing from left to right. The left side appears dimmer compared to the right side, which can be attributed to linear brightness correction. The effect of Gamma correction may not be visibly apparent without a direct comparison.

Upon observing the displayed image in Figure 13(c), it becomes evident that the shape of the original heart pattern is recognizable. Although there may be some color distortion, the flow of liquid within the heart is still clearly visible. Additionally, a small highlight area can be observed at the top right of the heart in the displayed image. Thanks to the effect of linear brightness correction, the overall brightness of the image appears uniform, without any noticeable decrease in brightness from the center to the edges.

The presence of the two cracks in the heart pattern can be attributed to the limitations of the camera equipment and the shoot mode used. However, it's important to note that these cracks are not visible to the human eye when observing the heart pattern in real-time. Due to the high-speed rotation of the LEDs at 1800 RPM, the heart pattern appears complete and undamaged. Furthermore, the rotation speed is appropriately adjusted, resulting in a natural and smooth flow of the liquid within the heart pattern.

2.2.3 Latency analysis of image processing

We have chosen the real-time video mode to assess the latency of image processing. This selection is based on the fact that the images captured by the built-in camera have high resolution, with diverse and irregular pixel patterns in complex environments, as opposed to cartoon images. Processing such images takes more time, and when encoding

them, a lower compression ratio is required. Additionally, achieving real-time display at 30 frames per second (FPS) places particular emphasis on minimizing the latency of image processing.

Here, we executed the real-time video mode for a duration of 10 seconds, following the procedures outlined in the R & V Table 6, on the computer used for the final demonstration. We measured the latency required for the image processing algorithm to process each frame. Since the camera outputs video at a rate of 30 FPS, we obtained approximately 300 frames during this duration. The results are illustrated in Figure 14.

It is evident that the latency of image processing varies between 1 millisecond and 3 milliseconds, with an average latency of less than 2 milliseconds. Considering that a frame takes approximately 33 milliseconds to generate in 30 FPS mode, the observed latency falls within an acceptable range.

2.2.4 Latency analysis of ESP32 program

The verification of the optimization of ESP32 program is direct. I start a real time video for 10 seconds and measure the latency it takes for the ESP32 program to execute and display a single frame.

As shown in the Figure 15, the latency is around 10ms and 25ms with some frames take more than 50ms to process. Consider the local RAM limitation (520KB) and maximum frame size ($320 \cdot 80 \cdot 2 = 50KB$), I design a video FIFO buffer that can store 5 frames stored before displaying, which means the program allows latency more than 150ms. By applying the FIFO buffer, the real time video display in rotating LED arrays can be stable in 30fps, which satisfied our requirements.

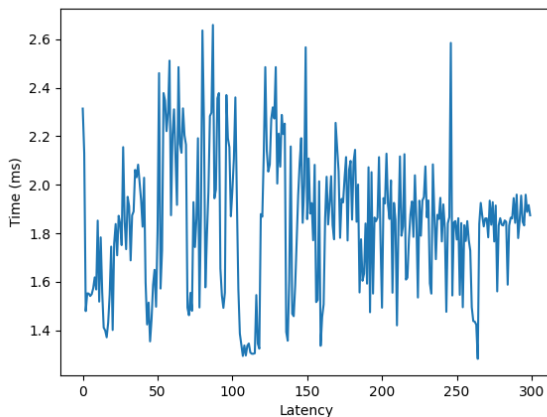


Figure 14: Image processing latency

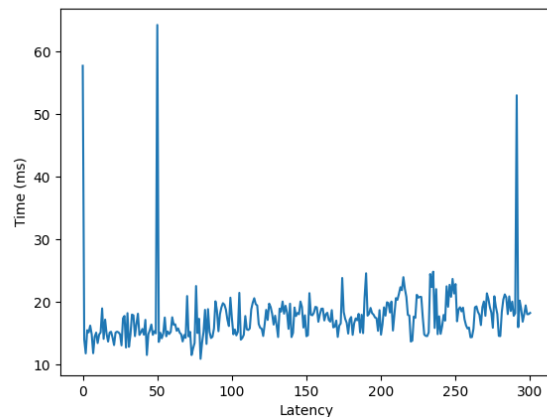


Figure 15: ESP32 program latency

2.3 Design Alternatives

We had initially planned to use an FPGA board as the central processing platform in our system to handle various types of input data, perform image processing algorithms, and transmit processed data to the ESP32 microcontroller. However, after careful consideration, we have decided to replace the FPGA board with a computer for the following reasons:

Our project sponsor, Prof. Li, has emphasized that as a display system, the primary criterion for success is achieving a comprehensive display effect that encompasses brightness, fidelity in terms of color and shape, and fluency in dynamic display. The selection of the central processing platform and other components should prioritize flexibility, ensuring that the components can fulfill their specific functionalities and be seamlessly integrated together.

When we reviewed our design, we realized the importance of usability for our system, in addition to display capabilities. We expect the display device to be compact and handheld, rather than fixed in place. So, we plan to use wireless communication to control the display device and simplify the design of the display device. Moreover, if we can further simplify the design of the logic and interface subsystem, it will significantly increase the usability of our system. In this regard, a personal computer has an incomparable advantage over FPGA.

Besides, during the development phase on FPGA, we discovered that it necessitates extensive usage of floating-point arithmetic for coordinate frame conversion from Cartesian to polar coordinates. However, the limited precision of floating-point arithmetic on FPGA can negatively impact the accuracy of the coordinate frame conversion, potentially compromising the fidelity of shapes displayed. Additionally, FPGA introduces additional latency when executing image processing tasks.

3 Cost & Schedule

3.1 Cost

In this section, we list our cost of labor and parts in development.

3.1.1 Labor

The estimated labor cost is listed in Labor Cost Analysis Table 1. We refer to the sample and past senior design projects to get the hourly salary and estimate total working hours based on our project workload and course credits.

Table 1: Labor Cost Analysis. The table shows the partner name in the project (Partner), the hourly salary of partner (Hourly Salary), the working hours in the project per person (Working Hours) and the total salary per person (Total).

Partner	Hourly Salary	Working Hours	Total
Chentai Yuan	\$40	150	$\$40 \times 150 \times 2.5 = 15000$
Guanshujie Fu	\$40	150	$\$40 \times 150 \times 2.5 = 15000$
Keyi Shen	\$40	150	$\$40 \times 150 \times 2.5 = 15000$
Yichi Jin	\$40	150	$\$40 \times 150 \times 2.5 = 15000$
Sum			\$60000

3.1.2 Parts

The approximated cost of parts is listed in Parts Cost Analysis Table 2. Five sets of materials were used for developing the system. We include cost of standard or non-standard parts, lab equipment, shop services in table. So, the estimated cost to reproduce our single system is about \$60, although some parts are not available for single purchase.

3.1.3 Total cost

The total cost of labor and parts is

$$\$60000 + \$300.448 = \$60300.448.$$

3.2 Schedule

The complete schedule is given in Table 3 below. Schedule is broken down by week and team member.

Table 2: Parts Cost Analysis. The table shows the part name and short description (Description), the manufacturer of the part (Manufacturer), the quantity needed in the project (Quantity), the unit cost of the part (Cost/unit) and the total cost of the the part (Total Cost).

Description	Manufacturer	Quantity	Cost/unit	Total Cost
TF card reader (TF-SMD.HYC77-TF09-200)	Huayuchuang	5	0.198	0.99
TF card	Sandisk	5	2	10
ESP32-WROOM-32E (4MB)	Espressif Systems	10	1.48	14.8
HDR-F-2.54_1x2	LCSC	8	0.031	0.248
Voltage Stabilizer (AMS1117-3.3 SOT-89 L4.5-W2.5-P1.50-LS4.2-BR)	Youtai Semiconductor Co., Ltd.	5	0.052	0.26
HALL (C6102 SOT23-3)	LCSC	10	0.73	7.3
10k resistor (R0603)	Jiufeng	30	0.01	0.3
LED (APA102)	Zhengpin	600	0.13	78
Motor (RS-385A)	YouxinDianzi	5	3.15	15.75
PCB printing	LCSC	5	8.20	41
12V DC Power supply	Chenke	5	3.21	16.05
Wireless charging module (24V45C)	Chenke	5	5.15	25.75
Labor cost SMT for LED	LCSC	5	15	75
3D Printing for PCB holder	LCSC	5	3	15
Total				300.448

Table 3: Weekly schedule table. The table shows the duration of weeks (Week) and individual schedule for every group member (Chentai, Guanshujie, Keyi and Yichi).

Week	Chentai	Guanshujie	Keyi	Yichi
2/13-2/19	Select project	Select project	Select project	Select project
2/20-2/26	RFA	RFA	RFA	RFA
2/27-3/5	Individual survey	Individual survey	Individual survey	Individual survey
3/6-3/12	Project Proposal	Project Proposal	Project Proposal	Project Proposal
3/13-3/19	Design PCB	Design GUI	Plan communication	Order parts
3/20-3/26	Order parts	Explore Verilog implementation	Order parts	Update visual effect design
3/27-4/2	Solder PCB, test components	Implement SDRAM I/O module	Develop image processing on PC	Test signal on the LED test series in static
4/3-4/9	Implement the Rainbow test code on ESP32, unit test on the LED array	Implement image decoding and pixel mapping in ESP32	Implement Wi-Fi on ESP32	Test the visual effect in motion
4/10-4/16	Test the system with LED array and the motor	Explore optimization of ESP32 program	Implement UDP	improve display, implement GUI
4/17-4/23	Implement the outer shell design and render the CAD	Design FIFO buffer on ESP32	Wireless test fails on FPGA	Integrate display system with control part
4/24-4/20	Finish the outer shell assembly	Merge subsystems	Develop image and video modes	Test and refine GUI
5/1-5/7	Integrate, finalize decoration	Test and debug	Develop text mode, integrate	Circuit debugging
5/8-5/14	Integrate all	Integrate all	Integrate all	Integrate all
5/15-5/21	Final Demo	Final Demo	Final Demo	Final Demo
5/22-5/28	Final Report	Final Report	Final Report	Final Report

4 Conclusion

4.1 Accomplishments

In this groundbreaking project, our team has successfully developed and realized a Rotating Display System that boasts an impressive array of features, all packed into a compact and easily portable size. The primary functionality of this system lies in its ability to wirelessly transmit text, images, and videos using Wi-Fi technology, opening up a world of possibilities for dynamic and engaging visual displays.

The display itself takes the form of a disk, measuring 160 pixels in diameter. With a pixel spacing of just 2 millimeters, the clarity and sharpness of the visuals are truly remarkable. Operating at a smooth 30 frames per second, the display ensures seamless transitions and fluid animations. Furthermore, the system offers the convenience of color temperature adjustment, allowing users to fine-tune the RGB three primary colors to achieve their desired visual effect.

To facilitate easy management of content, we have implemented a user-friendly graphical user interface in local computer and web page interface in ESP32. These interfaces allow users to effortlessly retrieve and select images or videos from the computer or those stored in the external storage of the ESP32. With the ability to abstract the overall system, users can browse through their collection of media files and choose the ones they wish to display on the rotating screen.

In addition to its impressive image and video capabilities, our system also includes a camera mode, which further enhances its versatility. With this mode activated, users can showcase real-time images captured by a connected camera directly on the display with a frame rate up to 30 FPS. This feature opens up exciting possibilities for live demonstrations, interactive experiences, and captivating visual presentations.

The combination of its compact design, wireless connectivity, and rich functionality makes our Rotating Display System an innovative and practical solution for a wide range of applications. Whether used for advertising, information display, art installations, or entertainment purposes, this system offers a captivating and immersive visual experience that is sure to leave a lasting impression.

4.2 Uncertainties

Despite the remarkable achievements in our final design, it is important to acknowledge the presence of certain uncertainties that may impact the overall performance of the system. One area of concern relates to the image quality in real-time video mode, particularly when processing frames that require more than $30ms$ to complete.

In scenarios where the video being displayed contains intricate and color-rich imagery, there is a possibility that the image quality may be compromised. The intricate nature of such visuals demands more processing time, and if the frames exceed the allotted time ($30ms$), it may result in a degradation of image quality. While the system is designed to

operate at a smooth 30 frames per second, the processing requirements of complex images can potentially strain the system's capabilities, leading to potential inconsistencies in the visual output.

Another aspect that may introduce uncertainties is the stability of the Wi-Fi connection. The quality of the Wi-Fi signal plays a vital role in the overall performance of the system, and a weak signal can impact the transmission latency, subsequently affecting the display quality. In instances where the Wi-Fi signal strength is compromised, the stability of the connection may suffer, leading to potential disruptions or delays in the transmission of data. These disruptions can directly impact the display quality, potentially resulting in visual artifacts or inconsistencies.

By recognizing these uncertainties, we can proactively explore mitigation strategies to minimize their impact. Through continued testing, monitoring, and iterative improvements, we aim to enhance the overall reliability and performance of the Rotating Display System, ensuring a consistent and high-quality visual experience for our users.

4.3 Future Work

We currently adjust multiple parameters in image processing manually for many different images and finally fix the parameters so that our system can display images with appropriate brightness in most cases. However, if the given image is too dim or too bright, the display effect might be less than ideal. When an image is too dim, we are unable to see details, especially if the ambient light is bright. Conversely, when it is too bright, intense light may cause discomfort. Moreover, the power consumed by the LED will increase significantly, potentially draining power supplied by the wireless charging module. Consequently, the ESP32 microcontroller may shut down. To address these issues, we can develop a more intelligent brightness correction algorithm that evaluates the brightness distribution of images and adjusts the brightness for better display.

In addition, we can develop some interesting applications based on our system. One idea is to add APIs that connect our system to popular artificial intelligence programs such as ChatGPT and Midjourney. This way, users can input text messages into our system, which can then display the text response from ChatGPT or the image generated by Midjourney. Another possibility is for the system to accept real-time sound or music streams and visualize the sound on the display. This would require analyzing the sound using technologies like Fourier transform.

4.4 Ethical Considerations

4.4.1 Ethics

Regarding to the IEEE [5] and ACM Codes [6] of Ethics, we summarize the following ethical concerns that apply to this project:

1. Consider the impact of our project on society: We should have a responsibility to consider the impact of our work on society as a whole, including the social, economic, and environmental consequences. This requires taking a holistic view and considering not only the immediate benefits, but also the potential long-term consequences.
2. Uphold the principles of fairness and justice: We should strive to ensure that our work is fair and just, and that it does not discriminate against individuals or groups based on factors such as race, gender, religion, or nationality. This requires being aware of potential biases and taking steps to mitigate them.
3. Be honest and transparent: We should be honest and transparent in the professional conduct, including the communication with team members, teaching assistant, instructors and the public. This requires being forthright about potential risks and uncertainties, and disclosing any conflicts of interest.

4.4.2 Safety

Regarding to the safety guidelines provided in course website and our project development flow, we summarize the following safety concerns that should be considered:

1. Electrical safety: The power system involved in our project could pose risks of electrocution or electrical fires if proper safety measures are not taken. This includes ensuring that all wiring is properly insulated and grounded, that circuits are appropriately sized and protected, and that appropriate safety equipment is available for handling and testing electrical components.
2. Mechanical safety: The rotating motor involved in our project could pose risks of injury or damage to equipment if not properly installed or operated. This includes ensuring that the motor is securely mounted and that all moving parts are properly guarded to prevent contact with users or other objects.
3. Lab safety: The design and testing of our LED display system involves working with various tools, equipment, and materials that can pose hazards to our members. This includes ensuring that all of us are trained in proper safety procedures, that appropriate safety equipment is available, and that all testing and assembly is performed in a designated and controlled laboratory environment.

References

- [1] L. Yong, M. F. Zakaria, and H. Hashim, "Design and development of spinning led message display system," Mar. 2012.
- [2] H. Demirel and G. Anbarjafari, "Image resolution enhancement by using discrete and stationary wavelet decomposition," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1458–1460, 2011. DOI: 10.1109/TIP.2010.2087767.
- [3] T. Morris, P. Blenkhorn, and F. Zaidi, "Blink detection for real-time eye tracking," *Journal of Network and Computer Applications*, vol. 25, no. 2, pp. 129–143, 2002, ISSN: 1084-8045. DOI: <https://doi.org/10.1006/jnca.2002.0130>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S108480450290130X>.
- [4] A. Sarode, D. Salunke, A. Shukla, and M. Sonawadekar, "Spinning led display using radio frequency," *International Journal of Computer Science and Informatics*, pp. 280–283, Apr. 2014. DOI: 10.47893/IJCSI.2014.1160.
- [5] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 02/08/2020).
- [6] ACM. ""ACM Code of Ethics and Professional Conduct"." (), [Online]. Available: <https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct> (visited on 02/08/2020).

Appendix A PCB Design (Partial)

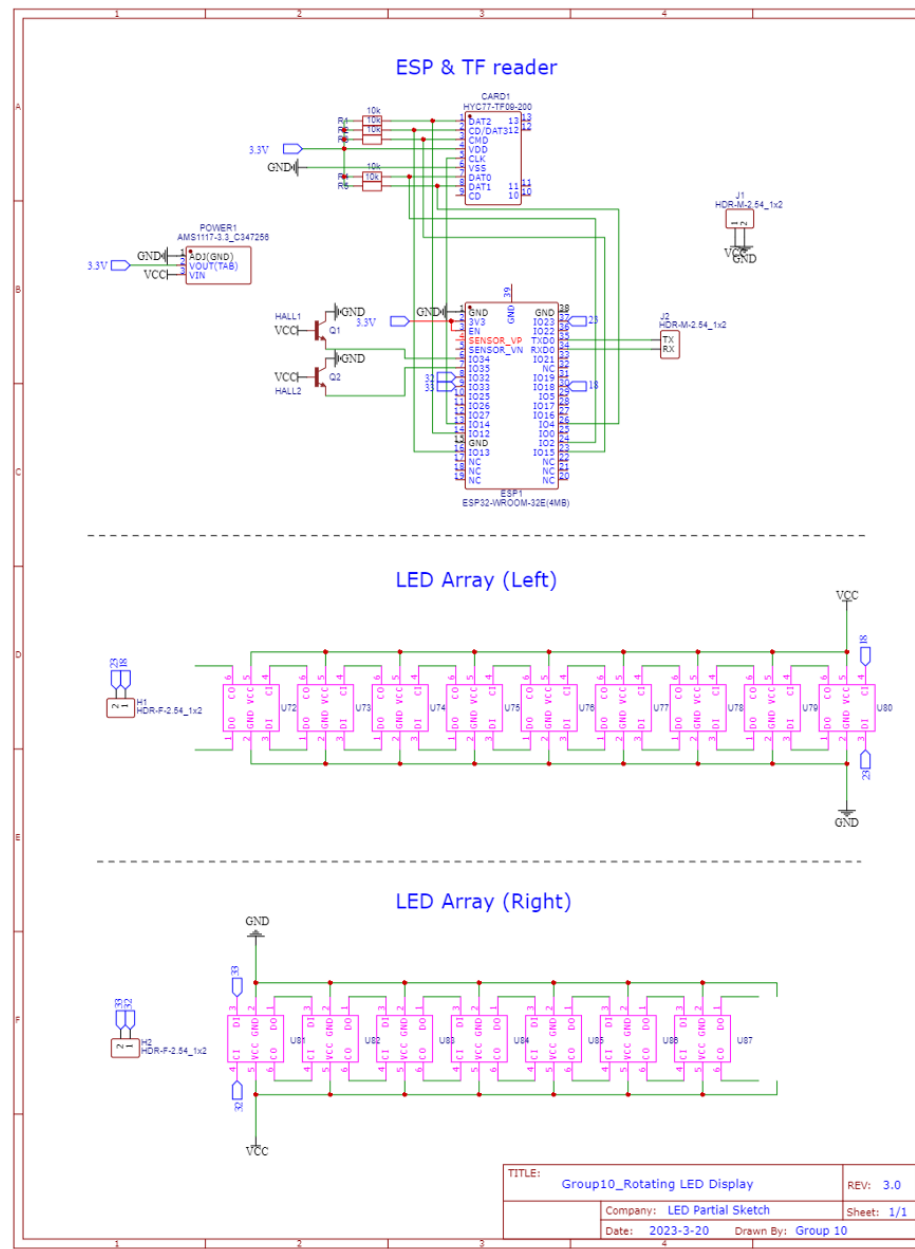


Figure 16: PCB Sketch (Partial)

Appendix B Requirements & Verification Table

B.1 Display and Control Subsystem

The following two tables 4 and 5 show the requirements and verification procedures of display and control part.

Table 4: R&V Table of display part

Requirements	Verifications	Status (Y/N)
The display is bright enough and the audience can observe the illuminance of over 50 lux	<ol style="list-style-type: none">1. Turn on our system with our test image, adjust the brightness to half of the maximum.2. Use an illuminometer to measure the illuminance of our system perpendicularly to the surface at a distance of 2m. The measured illuminance reached over 90 lux on average.3. Use the illuminometer to measure the illuminance of our system 45 degrees to the display surface, 2m away. The measured illuminance reached over 70 lux on average.	Y
The display is balanced, the whole surface is evenly luminous, which has a measured difference in illuminance within 10%	<ol style="list-style-type: none">1. Use a paper cylinder to cover the test probe of our illuminometer, so that it can only perceive light right from its front.2. Turn our system with a set of pure-color test images (white, red, green, and blue). Adjust the illuminance to half of the maximum.3. Turn on the illuminometer and place the test probe perpendicularly to the center and every 42.5mm from the center (the end of the cylinder is 40cm to the display surface), read, and record the result. Repeat the process six times.4. Compare all the result we get from step 3, calculate if the maximum difference of illuminance is 3.35% of the average value over all test colors.	Y

B.2 Logic and Interface Subsystem

The following Table 6 shows the requirements and verification procedures of Logic and Interface Subsystem. We use test programs to verify requirements. They were created by us during development.

Table 5: R&V Table of control part

Requirements	Verifications	Status (Y/N)
The motor will output a rotation speed 1800 RPM under a 12V power supply and torque of 480 g·cm	<ol style="list-style-type: none"> 1. The motor will connect with +12V power supply, and it will rotate with the speed of 5000 RPM. 2. An empty PCB will attach to the motor using flange. Applied a weight of 10 gram at the distance of 6 mm place on the PCB (It is a high-load test). 3. Use slow motion camera and record the rotation speed. It will rotate more than 30 circles per second. 	Y
The PCB have a displacement within 1.5 mm with the rotation speed of 1800 RPM	<ol style="list-style-type: none"> 1. Set a horizontal line (Line1) at the back of the PCB, which horizontal to the PCB. Set a line (Line2) parallel to the horizontal line, but 1.5 mm lower. 2. Repeat step 1 and step 2 above. 3. Use slow-motion camera and capture the movement of the upper PCB. The maximum displacement point (the edge of PCB) will within the Line2, which shows the maximum displacement less than 1.5 mm. 	Y
The Hall switch (SOT23-3) changes the status under the influence of magnet. (Type for magnet is 2 mm height, 6mm diameter. Distance with hall switch is 20-40mm)	<ol style="list-style-type: none"> 1. Connect hall switch with ESP32 chip on the firing tool, check the status is working for the chip. 2. Put the magnet at 40mm distance with the hall switch, the hall switch will not change the status. 3. Move the magnet 1 mm as a step, closer to the hall switch. Record the status for each step the switch will change the status at the distance of 28 ± 0.5 mm 	Y

B.3 Power and Base-shield Subsystem

The following Table 7 shows the requirements and verification procedures of Wireless Charging Module.

Table 6: R&V Table of Logic and Interface Subsystem

Requirements	Verifications	Status (Y/N)
<p>The transmission bandwidth between PC and ESP32 is large enough to handle 30 images per second. All images after processing should be over 1450 bytes and under 1550 bytes.</p> <p>In development, 1450 bytes is tested as the maximum size of images after processing.</p>	<ol style="list-style-type: none"> 1. Reset and start ESP32. 2. Check if two images 'test.jpg' and 'test2.jpg' are in the folder 'img'. Both images are close to maximum size after processing. 3. Run python program 'wifi_test.py' in the folder 'src'. It first connects to ESP32 Wi-Fi server. If it succeeds, you can see 'connected' in terminal. 4. Then it sends two images alternatively in 30 FPS in 10 seconds. It sleeps 30 ms after sending every image to simulate 30 FPS. 5. You can see size of every image in terminal. After 10 seconds, it prints the total number of images sent. It should be larger than 300. 6. ESP32 also prints size of every image received and the total number of images received in serial port output. If you connect ESP32 to PC with a serial port in Arduino application, then you can see result. 	Y
<p>The overall delay of image processing algorithm is less than 30ms for a single frame to guarantee the continuous display of video stream.</p>	<ol style="list-style-type: none"> 1. Reset and start ESP32. 2. Make sure that at least one camera is available to be used by PC. 3. Run python program 'image_test.py' in the folder 'src'. It first opens a camera connected to PC. If it succeeds, you can see 'connected' in terminal. 4. Then it starts to capture real-time video in 30 FPS, runs image processing for 10 seconds. 4. After 10 second, you can see the total number of images processed in terminal. It should be about 300. And it also shows a plot about the delay of image processed on every image. 	Y

Table 7: R&V Table of Power and Base-shield Subsystem

Requirements	Verifications	Status (Y/N)
Input voltage $23V \pm 1V$ Input current $1A \pm 0.05A$ Output voltage $7.5V \pm 0.5V$ Output current $3A \pm 0.5A$	<i>Verification process for input:</i> 1. Attach variable power supply to V_{in} . 2. Sweep from 10.0 V to 12.0 V and ensure input current remains within the desired value. 3. Ensure the emitter will not give any warning and work normally. Its temperature is below $70^{\circ}C$. <i>Verification process for output:</i> 1. Attach voltmeter and parameter to receiver output pin. 2. Ensure the output voltage and current is within the tolerance range	Y

Appendix C Details for Other Selected Parts

C.1 ESP32 Pins Configuration

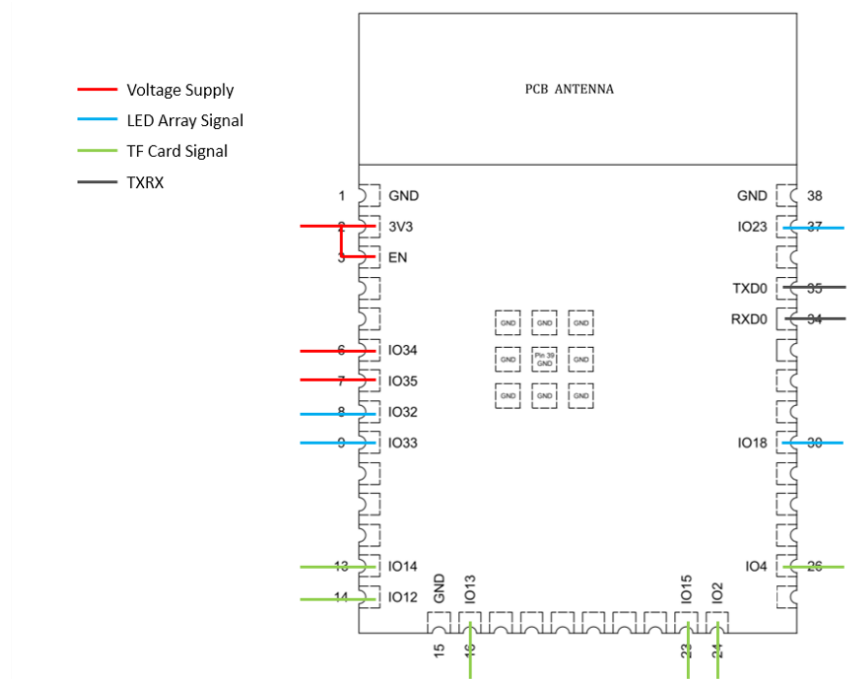


Figure 17: ESP32-WROOM-32E Pin Layout

C.2 TF card reader

The HYC77-TF09-200 TF card reader on PCB is supplied by 3.3V power, receive TF card push and output TF card data.

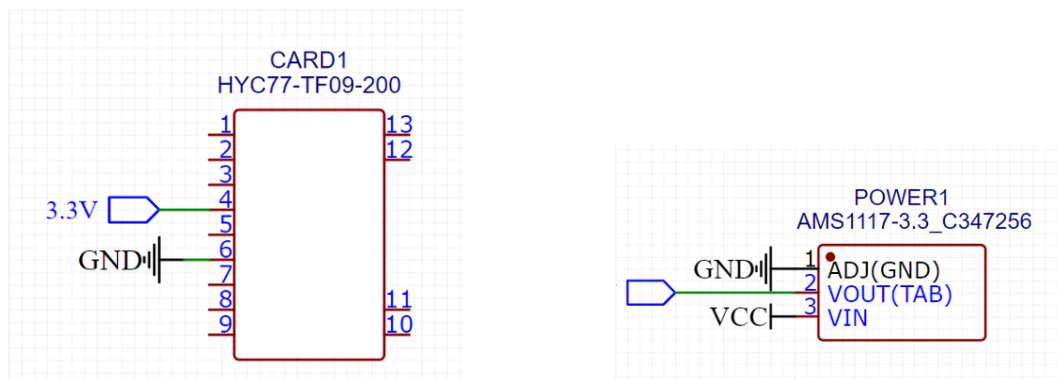


Figure 18: PCB Sketch of TF card reader Figure 19: PCB Sketch of Voltage stabilizer

HYC77-TF09-200 is a common TF card reader attached on PCB. It can realize the function of reading data stored in TF card. We will transfer these data to microcontroller (ESP32-WROOM-32E). Shown on Figure 18, HYC77-TF09-200 will be powered by a +3.3V source

Table 8: ESP32-WROOM-32E Pin connection

Pin	Function	Connection
2, 3	Voltage input, keep High on Pin3 to enable the chip	With voltage stabilizer (AMS1117-3.3)
6, 7	RTC_GPIO, awaken chip from Deep-sleep mode	VCC, through a Hall switch (SOT23-3)
8, 9 & 30, 37	CI/DI, as clock/data input for LED array	With LED array (Left and right)
13, 14, 16, 23, 24, 26	Receive data from TF card	Through TF card reader (HYC77-TF09-200) connect with TF card
34, 35	TX RX, receive and transmit data	Through programmer writer connect with computer
Built-in Bluetooth	Receive and transmit data with IEEE 802.11 wireless standard	Through WI-FI with PC

through voltage stabilizer (AMS1117-3.3). The pin connection table is shown in Table 9.

Table 9: HYC77-TF09-200 Pin connection

Pin	Function	Connection
1	SD_DATA2	GPIO 12, ESP32-WROOM-32E Pin 14
2	SD_DATA3	GPIO 13, ESP32-WROOM-32E Pin 16
3	SD_CMD	GPIO 15, ESP32-WROOM-32E Pin 23
4	+3.3 V voltage input	/
5	SD_CLK	GPIO 14, ESP32-WROOM-32E Pin 13
6	GND	/
7	SD_DATA0	GPIO 2, ESP32-WROOM-32E Pin 24
8	SD_DATA1	GPIO 4, ESP32-WROOM-32E Pin 26

C.3 Voltage Stabilizer

The AMS1117-3.3 Voltage Stabilizer on PCB accepts unstable power supply in range of 3V and +8V from the wireless voltage supply module which will be introduced in power subsystem, then output a 7.5V stable power supply. The PCB sketch is shown in Figure 19.

C.4 Switch/Connector

A self-locking switch is placed as a cutting off for the +12V voltage source. Detailed function for connectors is shown in Table 10 below.

Table 10: Switch and Connector Functions

Device	Requirement	Function
LA22-A2 Switch	Size: 22mm diameter; Type: Self-locking; electric parameter: bearing current should larger than 700mA (motor rated current)	Turn ON/OFF the connection of +12 V voltage source. Connect with +12V Li-PO battery and motor.
HDR-M-2.54_1x2	Connector bearing +8 V voltage	Connect with ESP32 (for TX/RX).
HDR-F-2.54_1x2	Connector bearing +8 V voltage	Connect with LED array (for LED test).

C.5 Motor

We choose RS-385A motor, which has 5000 RPM without load. We list some other parameters in Table 11.

Table 11: Motor Requirements

Function	Requirement
Physical dim	Shell diameter 29 mm. Output shaft diameter 2.3 mm, 11 mm length
Voltage supply	+12V DC
Rated current	550 mA
No load RPM	5000 RPM
Stall condition	Torque = 750 g·cm

C.6 LED array

Our LED array is made up of 160 square LC8822-2020 LEDs, whose diagram is shown in Figure 20. The LEDs have an integrated circuit inside and with a single sequence of data signal we can control a serial of LEDs, this greatly reduce the complexity of our PCB configuration Table 12. LED array supplied by 7.5V DC Power and controlled by ESP32-WROOM-32E through pins 8, 9 and 30, 37. For a single LED, the control signal is formatted as 24-bit RGB, 5-bit brightness (D[0:4]) and 3-bit padding. For the whole LED array, the control signal starts with 32-bit starting "0"s, follows with 160 32-bit control signal for every LED and end with 32-bit ending "1"s.

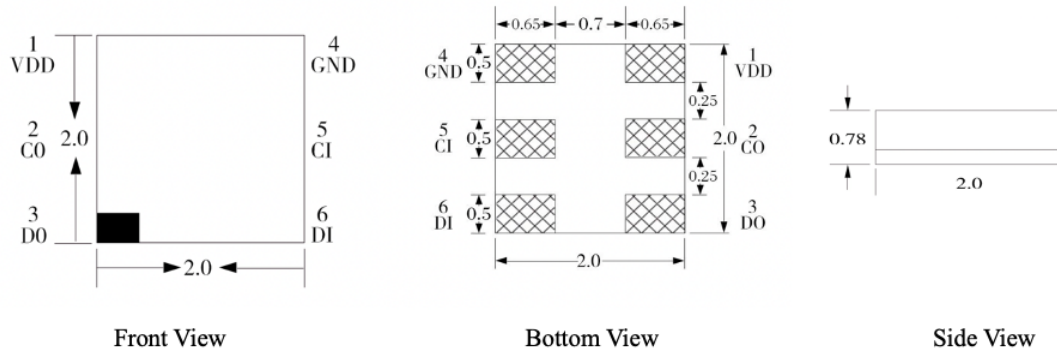


Figure 20: Three-view drawing of LED

Table 12: Pin Configuration of LED

Pin	Function	Connection
VDD	Supply Voltage	VDD input
CO	Clock output	Next LED's CI
DO	Data output	Next LED's DI
GND	Ground	Common ground
CI	Clock input	From clock signal/previous LED's CO
DI	Data input	From input signal/Previous LED's DO

C.7 Wireless Charging Module

In the following formula N corresponds to the number of turns of the coil and r is the distance where we measure from the coil that produce the field.

$$B = \frac{N\mu_0 I}{2\pi r} \quad (22)$$

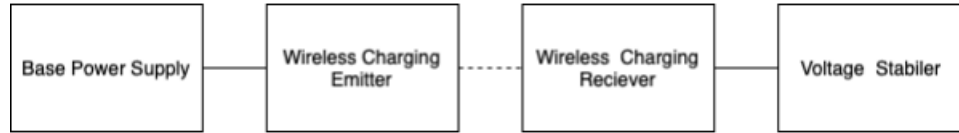


Figure 21: Wireless Charging Connection

The power is transfer back to electricity from the magnetic field follows the following formula:

$$E = n \frac{\Delta \Phi}{\Delta t} \quad (23)$$

The receiver will be connected to the voltage stabilizer and its pin configuration is shown in Table 13.

Table 13: Connection for the wireless charging module

Device I/O	Requirement	Function
Input Vcc on the emitter	Input Anode. Supply 24V, 1A input to the emission part	Connected to a 24V, 1A power supply in the base part
Input GND on the emitter	Input cathode	Connected to the ground
Output high on the receiver	Output 7.5V, 3-4A	Connected to the voltage stabilizer high input
Output low on the receiver	Output low	Connected to the voltage stabilizer low input