EE326 final project

Xu Zhao

March 2025

1 Introduction

This project centers around the design and implementation of a networked camera system capable of real-time video streaming. Under the guidance of the course instructor, we successfully integrated key hardware components including the SAM4S8B microcontroller, ESP32 WROOM WiFi module, and OV2640 camera module, and realized their efficient collaboration through a custom-designed printed circuit board (PCB) and supporting firmware.

Throughout the project, my primary responsibilities focused on the hardware development aspect. Specifically, I was in charge of designing the PCB layout, constructing and soldering the breakout board circuits, assembling and debugging all hardware components, and modeling the 3D-printed protective enclosure. My teammate, Quan Zhou, concentrated on firmware development using Microchip Studio, integrating the camera and WiFi module through embedded C programming, and constructing the interactive website interface for live video streaming.

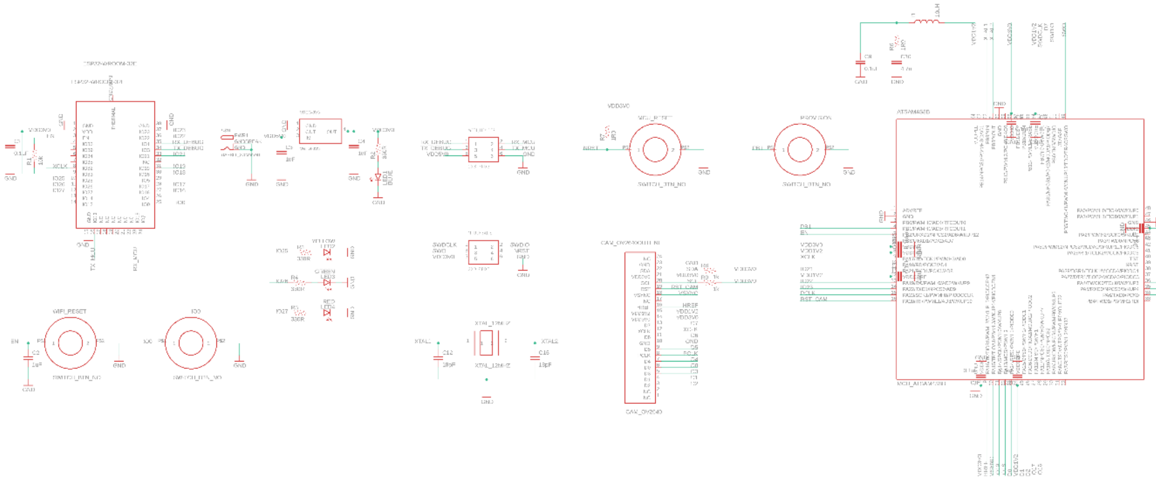
Our collaborative workflow allowed us to leverage each other's strengths effectively: I handled the entire hardware stack from design to physical assembly, while Quan Zhou managed the software and network communication layers. Together, we achieved the project's core objectives: establishing a reliable WiFi-based real-time video transmission system, mastering embedded programming and hardware-software integration techniques, completing a functional and manufacturable PCB, developing a user-friendly web interface, and delivering a structurally sound and aesthetically pleasing enclosure.

This report outlines our design approach, challenges encountered, technical solutions, and the personal learning experiences gained throughout the project development process.

2 Design Process

2.1 PCB design

The printed circuit board (PCB) design of this project was tailored to optimize component layout, signal integrity, and ease of soldering, while taking into consideration the form factor constraints imposed by the 3D-printed enclosure. To achieve a balanced and efficient design, we utilized a double-sided PCB layout, strategically distributing the various components across both sides of the board to minimize signal interference and improve thermal management.

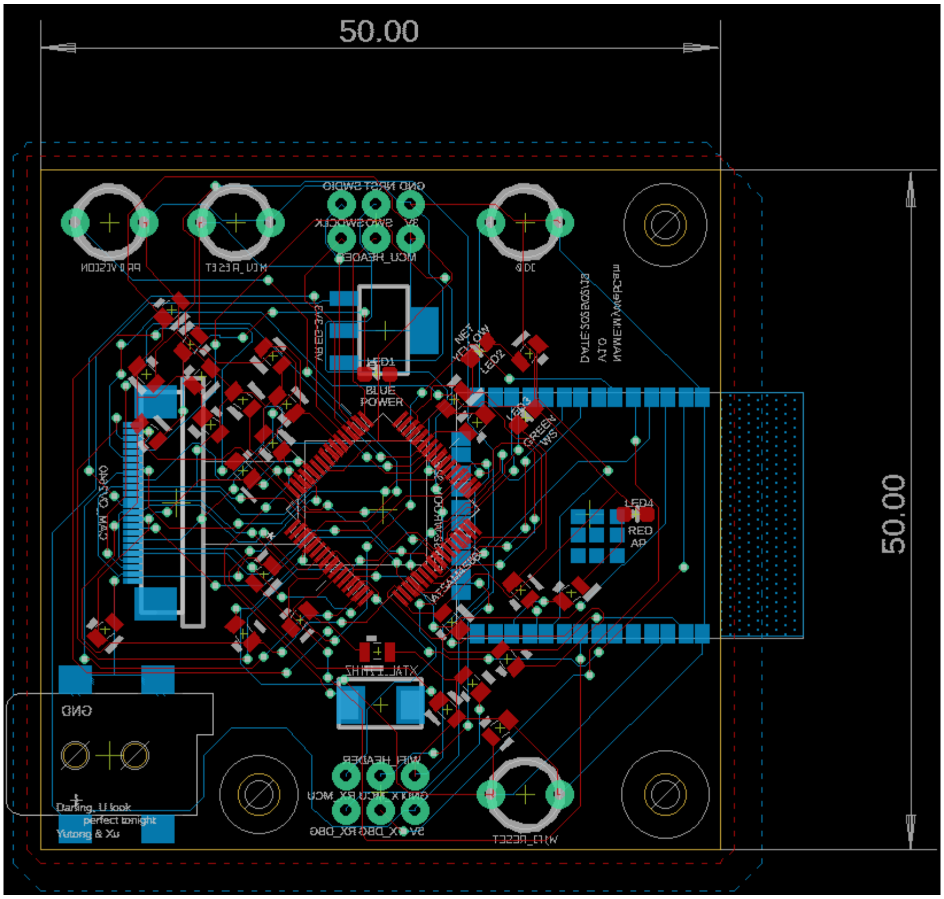


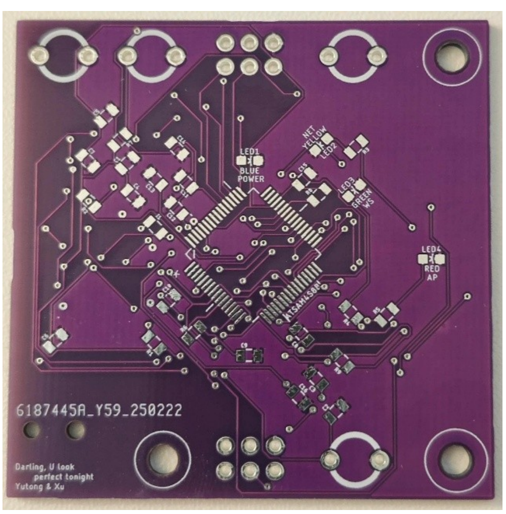
One of the distinguishing features of our PCB design is the intentional 45-degree rotation of the central microcontroller unit (MCU), the SAM4S8B. This rotation was adopted not only as an aesthetic choice but also to streamline the routing of peripheral traces and facilitate clearer separation between power, control, and data lines. On the same side as the MCU, we populated all the passive components, including resistors, capacitors, inductors, and LED indicators. This clustering of small components around the MCU allowed for minimized trace lengths and reduced parasitic effects, ensuring stable operation of high-frequency control signals.

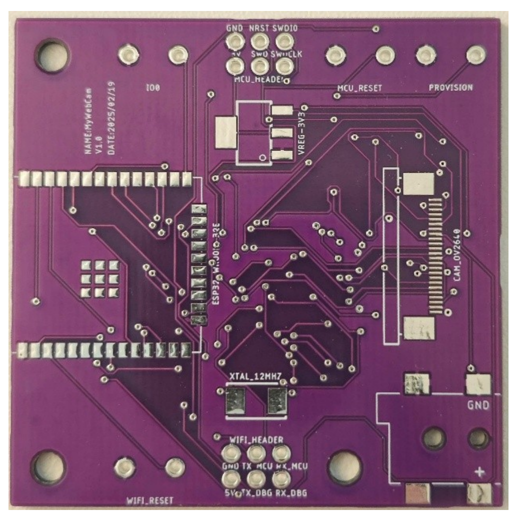
Conversely, the opposite side of the PCB was reserved for larger, more interference-prone modules: the ESP32 WROOM Wi-Fi module, OV2640 camera connector, the external 12 MHz crystal oscillator, and the power management circuitry. This deliberate separation of high-frequency communication modules from sensitive control circuitry mitigated potential electromagnetic interference (EMI) and facilitated a more organized, modular assembly process. Additionally, dedicated decoupling capacitors were placed near both the MCU and ESP32 power pins to suppress voltage fluctuations and enhance system reliability.

During the post-fabrication phase, while conducting continuity testing and functional debugging, I identified an unintended wiring error: the free general-purpose input/output (GPIO) pin on the MCU had been mistakenly routed to the XCLK input pin of the OV2640 camera module. This misconnection had the potential to disrupt camera operation by sending unintended signals to the clock line. To rectify this issue without necessitating a full re-fabrication of the PCB, I carefully severed the erroneous copper trace using a precision hobby knife under magnification. This manual correction successfully isolated the XCLK line, allowing the camera to receive a properly configured clock signal without interference. This incident highlighted the importance of thorough schematic verification and post-production inspection in PCB development, and it further underscored the value of maintaining flexible debugging strategies to address unforeseen hardware faults.

Throughout the design process, EAGLE was utilized for schematic capture and layout design. Design rule checks (DRC) and electrical rule checks (ERC) were rigorously performed to ensure adherence to manufacturing tolerances and signal integrity standards. Special attention was paid to the routing of differential pairs and the isolation of analog and digital







grounds to minimize crosstalk and noise.

In summary, the PCB design phase combined careful planning, practical layout considerations, and real-time problem-solving to produce a functional, reliable hardware platform for the networked camera system. The design decisions—including the unique MCU orientation, component distribution strategy, and precise corrective actions—demonstrate a methodical approach to embedded hardware development.

2.2 C programming

Our firmware development was performed in the Microchip Studio environment using the C language, and the firmware serves as a key communication bridge between the SAM4S8B microcontroller, the ESP32 WROOM Wi-Fi module, and the OV2640 camera. We initially referenced several example projects provided by Microchip Studio as a guide to the infrastructure and to ensure best practices for integration of peripherals and modules.

We started with basic MCU functionality configuration, using sysclk_init() to set the system clock, wdt_disable() to disable the watchdog to prevent accidental resets, and ioport_init() to initialize the necessary input/output ports. This phase is critical, and accurate peripheral initialization ensures that the entire project will run stably in subsequent phases.

Next, we focus on the integration of the OV2640 camera module. Our goal is to initialize the camera via the TWI (Two-Wire Interface) interface, configure the TWI connection using configure_twi(), and set the camera's JPEG output mode by calling configure_camera(). In addition, we use start_capture() to let the MCU capture the image data stream. To understand and properly configure the camera's registers, we refer to the given datasheet for register configuration examples and combine it with existing sample project code to ensure that the camera is able to correctly capture 320x240 resolution JPEG images.

For communication with the ESP32 Wi-Fi module, we implemented UART serial communication handlers such as configure_usart() and write_wifi_command() to send commands and receive responses. During the initial debugging, we encountered a synchronization problem where the ESP32 did not immediately respond to the commands sent by the MCU. To solve this problem, we implemented an interrupt-based UART handler (wifi_usart_handler()), which allows the MCU to listen to the return data from the ESP32 in real-time, and ensure that the commands are executed successfully before proceeding to the next step.

One of the core functions of the firmware is to capture images from the webcam and stream them via WebSocket. The logic is as follows: when a web page triggers a capture request, the camera captures an image and stores it in a buffer, and then sends the data to the Wi-Fi module by calling write_image_to_web() through the UART interface. Initially, we found that there are frame loss and delay problems during data transmission. After detailed debugging and optimization, especially in the timing mechanism (configure_tc()) and inter-module synchronization strategy, we significantly improved the stability and frame consistency of the video stream.

```c
while (true) {
    if (provisioning_flag) {
        write_wifi_command("provision", 1);
        provisioning_flag = false;
        } else if (ioport_get_pin_level(WIFI_CLIENT_PIN_MASK)) {
        start_capture();
        if (len_success) {
            write_image_to_web();
        }
    }
}
```

Throughout the development process, we frequently used Tera Term for debugging to monitor UART communications in real-time to verify various stages of initialization, camera image capture, Wi-Fi connectivity, and data transfer. This iterative debugging approach allowed us to quickly locate and resolve small faults and improve development efficiency.

Overall, our structured approach to firmware development - drawing on example projects, step-by-step implementation, and rigorous testing - allowed us to overcome the initial challenges of integrating hardware and software, and ultimately succeed in implementing reliable real-time video streaming capabilities.

2.3 Website Development

The web interface is an important part of this project, providing users with a platform for live video streaming that can be accessed through a browser. To build this interface, we used standard Web technologies, including HTML, CSS, and JavaScript, to ensure compatibility and ease of use.

We developed the site primarily using core Web technologies without relying on additional server-side frameworks. HTML was used to build the content of the pages (homepage.html, webcam.html, and info.html), CSS (homepage_styles.css, webcam_styles.css, and info.html), and JavaScript (homepage_styles.css, info_styles.css, homepage_styles.css, homepage_styles.css, homepage_styles.css, homepage_styles.css, and info.html), info.html), CSS (homepage_styles.css, webcam_styles.css, info_styles.css) for styling and visual layout, and JavaScript (homepage_functions.js, webcam_functions.js, info_functions.js) for dynamic client-side interaction.

Our website contains a clean navigation bar that allows users to easily access different functional pages, including the home page, camera video streaming, info page and Wi-Fi file management section. The overall design focuses on clarity, ease of use, and responsive layout to ensure that users can smoothly switch between functional pages. The homepage (homepage.html) is designed to welcome the user and briefly introduce the functions and usage of the system. The page is designed with soft colors and intuitive navigation to enhance user experience. The webcam.html page is the core functional module responsible for presenting the real-time video stream captured by the OV2640 camera.

In the webcam.html page, we use WebSocket for real-time video transmission. In the JavaScript logic (webcam_functions.js), we establish a WebSocket connection to handle the live image stream from the ESP32 Wi-Fi module.

The core logic is controlled around a button (streamButton) that allows the user to start and stop the video stream and dynamically display timestamps and image updates in the browser. During the initial development phase, we encountered connection stability issues. To address this issue, we optimized WebSocket state management, including operations such as connecting (onopen), disconnecting (onclose), and data processing (onmessage) to ensure a smooth user experience. The processing of image data uses JavaScript's FileReader API to make the reading and updating of video streams smoother, thus providing a stable real-time video viewing experience.

To ensure the stability of the system, we had done several rounds of comprehensive testing to verify the functionality of all UI components and the reliability of video streaming with different network conditions and devices. To address the latency and connection stability of the streaming service, we optimized the timing mechanism and added user feedback indicators to further improve the overall user experience.

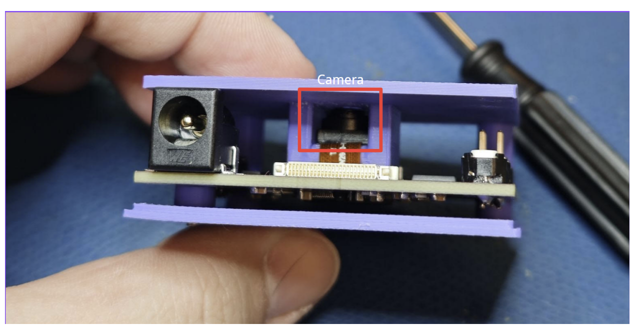
In conclusion, through our optimizations and rigorous testing, we finally implemented a stable real-time video streaming web interface, which successfully met the technical require-

```javascript
websocket.onopen = function(evt) {
    streamButton.textContent = "Stop Webcam";
    streamButton.title = "stop webcam";
    writeToScreen("Connected");
    buttonClicked = false;
};

websocket.onclose = function(evt) {
    streamButton.textContent = "Start Webcam";
    streamButton.title = "start webcam";
    if (buttonClicked) {
        writeToScreen("Webcam stopped.");
    } else {
        writeToScreen("Disconnected unexpectedly. Error: " + evt);
        doConnect();
    }
    buttonClicked = false;
};

websocket.onmessage = function(evt) {
    const now = new Date();
    timestampDiv.textContent = "Last Update: " + now.toLocaleTimeString();

    const reader = new FileReader();
    reader.onload = function(e) {
        webcamImage.src = e.target.result;
    };
    reader.readAsDataURL(evt.data);
};
```

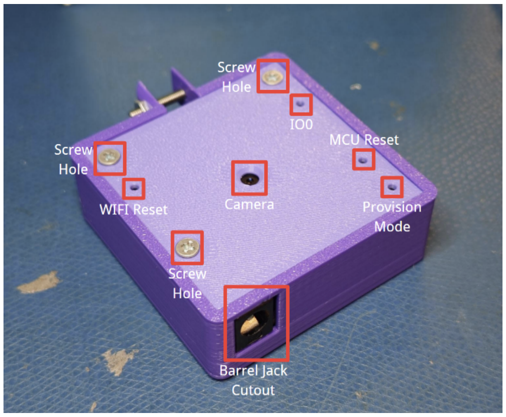
ments and user expectations.

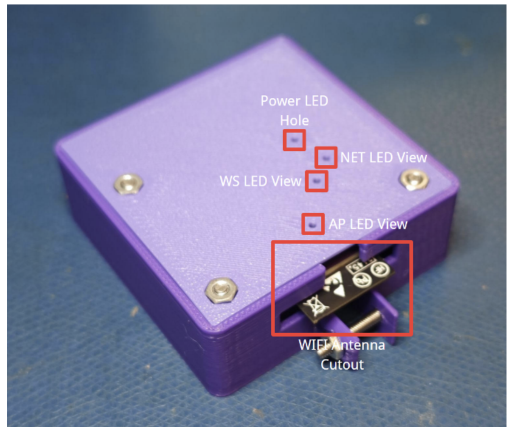
2.4 3D Enclosure Design

In this project, the design of the 3D-printed protective enclosure posed both functional and structural challenges, particularly due to the vertical height discrepancy between the power module and the camera module on the PCB. To address these challenges and fulfill the specific requirements outlined in Task 7, I departed from the conventional two-part enclosure design—typically comprising a container and a lid—and instead developed an innovative three-part enclosure structure, which I term the "sandwich structure." This design concept draws inspiration from modern smartphone architectures, where multi-layered stacking improves both stability and spatial utilization.



The primary objective of this sandwich structure is to securely house the PCB while minimizing internal movement and ensuring precise alignment of interactive components such as the camera, buttons, and indicator LEDs. In this design, the PCB is sandwiched between a top cover and a bottom cover, with an external frame encasing the assembly. The frame not only encloses the sides of the enclosure but also integrates critical access openings and structural features.





Specifically, openings for the power supply barrel jack, WiFi module, reset and provision buttons, and various indicator LEDs have been carefully dimensioned and positioned according to the Task 7 specifications to ensure usability and compliance. The external frame is further enhanced by a set of precisely engineered mortise and tenon joints, which interlock with the top and bottom covers, effectively securing all three parts together and preventing relative displacement.

A key innovation in my design is the inclusion of a camera support bracket integrated into the top cover. Given that the power module's height significantly exceeds that of the camera module, this bracket elevates the camera to match the height of the power module, ensuring both modules are positioned on a common plane. This eliminates any potential obstruction in the camera's field of view and stabilizes the camera against shaking during operation. Additionally, the camera opening on the top cover is chamfered to reduce any risk of visual occlusion and improve the aesthetic smoothness of the front face.

Attention was also paid to fastening mechanisms. I designed countersunk screw holes with surrounding recesses to accommodate hexagonal nuts, allowing the screws to sit flush with the enclosure's outer surface after assembly. This ensures a sleek exterior without protrusions, preventing mechanical interference and enhancing overall handling safety. The enclosure wall thickness is maintained at 1 mm throughout, adhering to the required minimum thickness for mechanical strength and printability, while keeping the enclosure lightweight.

For mounting purposes, my design adopts a non-traditional approach by cleverly integrating the mounting mechanism directly into the structural interplay between the external frame and the top cover. Instead of explicitly adding separate mounting holes, I utilize the protrusion of the screws used to secure the external frame. These screws intentionally extend beyond the surface of the top cover, and together with the contour of the top cover, naturally form mounting holes through the resulting gaps. This innovative design eliminates the need for additional mounting features, maintaining the enclosure's clean aesthetic while ensuring structural simplicity. Furthermore, by leveraging the existing fastening hardware, this solution enhances the enclosure's overall cohesiveness and reduces unnecessary complexity. The resulting mounting holes effectively allow for wall installation while preserving the integrity and unified appearance of the enclosure, fully satisfying functional requirements in an elegant, integrated manner.

In summary, the three-part sandwich structure improves upon traditional designs by enhancing internal stability, accommodating height discrepancies between components, and facilitating assembly without compromising accessibility or aesthetics. Through meticulous attention to component positioning, mechanical tolerances, and fastening details, this design successfully meets the functional requirements and provides a robust, user-friendly protective enclosure for the networked camera system.

2.5 Challenge and Solution

Throughout the course of the project, we faced a variety of technical and practical challenges that needed to be overcome which trained our capability of careful analysis and problem solving.

2.5.1 PCB Design

During the PCB development phase, we encountered a serious hardware problem: the ESP32 Wi-Fi module was incorrectly connected to the wrong GPIO pin on the MCU expansion board. After detailed troubleshooting, I found this wiring error and promptly removed the incorrectly connected wires and restored the PCB to its intended connection. This adjustment got our PCB back up and running again and emphasized the importance of careful verification during the hardware design process.

2.5.2 Firmware Development

Synchronizing the MCU, Wi-Fi module, and camera was a challenging task during firmware development. One of the prominent problems is unstable data transfer between modules, resulting in intermittent data loss. To diagnose this problem, we conducted several rounds of iterative testing and utilized PUTTY for serial port monitoring. Through in-depth analysis of the UART communication signals, we found that the root of the problem lies in the imperfect communication interrupt handling logic. By optimizing and improving the interrupt handling mechanism, we successfully solved the synchronization problem and improved data stability.

2.5.3 Website Development

In the process of web development, we encountered an unexpected behavior: after we uploaded the customized web page to the file management system of the Wi-Fi module, the browser still showed the default streaming page of the ESP32 module, which resulted in a confusing layout and affected the normal operation of our web functions. After detailed debugging, we found that there is a conflict between the default index.html page that comes with the ESP32 module and our customized web page. In the end, we successfully resolved this conflict by renaming and reorganizing the structure of the HTML file so that the web page we designed could be displayed correctly.

2.5.4 3D Enclosure Design

One of the main challenges posed by the PCB design was the central layout of the camera module, which made it difficult to securely fasten it to the edges of the housing. In addition, the limited length of the camera's connecting wires resulted in restricted mounting locations. By redesigning the 3D housing model, I cleverly added internal support structures and optimized the dimensions of the housing so that the camera could be securely mounted at the right viewing angle and distance. At the same time, he precisely opened a small aperture in the housing to ensure that the camera could capture images properly.

Looking back at the whole project, we deeply realize the importance of careful pre-planning during the hardware integration and testing phase. At the same time, we also realize that we still have deficiencies in the details, for example, in handling data transmission and segmentation, we still have some incomplete understanding of some of the processes, but thanks to the communication with the TA and other team members, we still completed this part of the design. In the future, we hope to have a deeper understanding of signal processing.

2.6 Improvement for Future

In reflecting on the results and performance of the current design, we have identified two main directions for improvement to enhance the overall experience and stability of the system. First, our current video streaming frame rate and bit rate are low, resulting in suboptimal picture quality and affecting the user experience. In future releases, we plan to focus on optimizing the firmware and network transmission protocols to improve the overall streaming quality. Specific measures include optimizing image encoding and transmission efficiency, reducing data loss during transmission by adjusting the camera's JPEG encoding parameters, and compressing image data to increase transmission speed. In addition, we will improve the data processing capability of the Wi-Fi module, adjust the Wi-Fi transmission buffer of the ESP32, and optimize the slicing and transmission strategy of the data stream to reduce frame loss.

Secondly, during the testing, we found that the problem of image loading failure occurs occasionally on the web page side, which affects the reliability of the system. To solve this problem, we will improve the JavaScript logic, optimize the WebSocket event handling mechanism to ensure smooth data reception, and adjust the JavaScript code to better parse and render the image data. At the same time, we will enhance the robustness of WebSocket communication and implement better error handling and data validation mechanisms, such as automatic retry when data reception fails, to ensure the stability of WebSocket connections. In order to further improve the performance of the web page side, we are also thinking about whether we can optimize the web page caching strategy, and introduce a more appropriate caching mechanism on t he browser side, so that we can reduce repeated loading and improve the image refresh speed.

Through these optimizations, we hope to improve the quality of video streaming in future releases to make it smoother and clearer, as well as to improve the stability and reliability of the system to ensure that users can have a better experience.

3 Teamwork

Throughout the development of this project, my teammate Quan Zhou and I collaborated closely, each contributing our strengths to ensure the successful completion of the networked camera system. I believe our division of labor was clear and efficient, with each member focusing on distinct yet complementary aspects of the project.

In particular, I took the lead in all hardware-related tasks. I was responsible for the design of the printed circuit board (PCB), carefully considering the component layout, signal integrity, and mechanical constraints to ensure optimal functionality and manufacturability. I also designed and modeled the 3D-printed protective enclosure, paying special attention to structural stability, component alignment, and aesthetic appearance, which involved several iterations to accommodate the physical discrepancies between modules such as the power supply and camera. Additionally, I handled the breakout board circuit construction, ensuring that all connections between the MCU, WiFi module, and camera were correctly implemented. All soldering work—both surface mount and through-hole components, including the precise soldering of the MCU and WiFi module breakout boards—was completed by me, requiring steady hands and thorough inspection to guarantee no short circuits or loose connections.

Meanwhile, Quan Zhou primarily focused on the software development side. He was responsible for the firmware programming in Microchip Studio, including configuring the MCU peripherals, integrating the camera driver, and establishing communication protocols with the WiFi module. He also developed the interactive website interface, implementing the navigation structure, WebSocket streaming functionality, and ensuring a smooth user experience across the homepage, webcam page, and information page.

In retrospect, I believe the workload between us was well-balanced and fair. While my responsibilities concentrated on the hardware design, assembly, and debugging, Quan Zhou's tasks revolved around the software implementation and optimization. Both roles were essential and interdependent, and we maintained constant communication to ensure seamless integration between hardware and software components. Our collaborative efforts allowed us to leverage each other's expertise, resulting in a fully functional, robust, and user-friendly embedded camera system.

4 Learning

I enrolled in EE 326 with the expectation of deepening my practical skills in embedded system design, especially in areas such as PCB development, hardware debugging, and system integration. Having prior experience in embedded systems, I hoped this course would allow me to apply and refine these skills in a real-world, multidisciplinary project environment.

Throughout the quarter, I gained substantial hands-on experience, particularly in the design and implementation of hardware components. The PCB design phase significantly enhanced my proficiency with EAGLE, and I learned to carefully balance component placement, signal integrity, and manufacturability constraints. Moreover, through the process of creating a breakout board, I became adept at reading datasheets, designing custom library components, and translating abstract circuit requirements into a physical board layout.

One of the most valuable skills I acquired was surface mount soldering. While I had some basic soldering experience prior to this course, the complexity and density of the components used in this project—especially the SAM4S8B MCU and the ESP32 WiFi module—required me to develop a much higher level of precision and control. Reworking solder joints, identifying bridges, and ensuring strong mechanical and electrical connections became second nature to me.

Additionally, designing the 3D-printed enclosure offered me insights into mechanical design considerations. I had to think beyond the circuit board, considering how the physical assembly, component heights, and user interaction (buttons, LEDs, camera alignment) would affect the final product's functionality and usability. This broadened my perspective on the importance of integrating electrical and mechanical design processes cohesively.

While the course exceeded my expectations in terms of hardware exposure and practical challenge, I found that I had less opportunity to delve deeply into certain aspects of the software and networking stack, particularly the low-level configuration of the WiFi firmware. In future iterations of the course, it might be beneficial to allocate more lecture time or optional workshops focusing on WiFi module internals and optimization techniques.

Overall, I believe the workload of this course was appropriate. The project was complex and time-consuming, especially with the meticulous hardware design and debugging, but every stage contributed meaningfully to my learning. That said, the deadlines occasionally clustered tightly around other course obligations, so perhaps minor adjustments to pacing could make the experience even smoother without reducing the challenge.

In conclusion, EE 326 provided me with a comprehensive, hands-on learning experience that bridged theoretical knowledge with real-world embedded system development. I leave the course with improved technical skills, greater confidence in hardware debugging, and a deeper appreciation for the interdisciplinary nature of electronic system design.

5 Conclusion

Reflecting on the entire development process of this project, I feel a strong sense of accomplishment and growth. From the initial schematic design to the final hardware assembly, firmware integration, and real-time video streaming demonstration, this project encapsulated the core principles and challenges of embedded system design. Each phase demanded not only technical proficiency but also meticulous attention to detail, effective problem-solving, and close coordination with my teammate.

Personally, I am particularly proud of the hardware aspects I contributed to the project. Designing and fabricating the PCB, constructing the breakout board circuit, soldering all components, and developing the custom 3D-printed enclosure allowed me to consolidate and expand my practical skills. I encountered unforeseen hardware challenges, such as wiring errors and mechanical alignment issues, but through careful debugging and iterative adjustments, I was able to resolve them and ensure the system's stability.

Additionally, working in tandem with my teammate reinforced my appreciation for interdisciplinary collaboration. While I focused heavily on hardware design and implementation, seeing the seamless integration with the firmware and web interface highlighted the importance of cohesive teamwork in realizing a fully functional embedded system.

Overall, EE 326 has been one of the most rewarding courses in my academic journey. It provided a comprehensive platform for applying theoretical knowledge to tangible outcomes and gave me invaluable experience in navigating real-world engineering constraints. Moving forward, I feel better prepared and more confident to tackle complex hardware-software integrated projects, both in academic research and future professional endeavors.