SnapLog Camera

FINAL REPORT FOR ECE 445, SENIOR DESIGN

By

Members of Team 70
Tianshu Wei
Fei He
Shuai Huang

May 2024

Abstract

This report introduces SnapLog, a camera necklace. SnapLog is a novel solution to the repetitive nature of traditional journaling. Designed to simplify the process of recording daily activities, SnapLog offers a unique approach by generating timelapse videos of one's day at regular intervals, and then compiles these snapshots into a cohesive video story. Furthermore, all photos taken by SnapLog are securely transmitted to the user's device, such as a laptop, ensuring easy access and storage of captured memories. By eliminating the need for manual note-taking, SnapLog offers a less intrusive and more enjoyable way to document daily events. This report discusses the design process, technical aspects, design verifications, and its core features.

Contents

1. Introduction

Journaling can be repetitive and boring for people. Not only does it take time, but it also requires mental work to recall, note down, and often describe the events of the day coherently.

Our project, SnapLog Camera, is designed to be a solution to this problem. SnapLog's core feature is the ability to generate a timelapse—a series of images shot at regular intervals that, when combined, provide a dynamic overview of your day. The camera is set to take an image every few minutes, and each one is quickly and securely delivered to your smartphone. Using a dedicated operation, these photographs are creatively stitched together to produce a cohesive video story, capturing the spirit of your day in an interesting and instructive style. As the day comes to an end, the app finalizes the time lapse, encoding it into a film that can be reviewed or shared, providing a unique way to contemplate and archive your daily journey.

Besides, the product is designed to have a long-lasting battery life, small form factor, and almost hands-free experience. Our final product could boast 42 days of continuous operation in a single charge[1]. It will be designed to have a minimal intrusion in one's daily routine and the experience will be silk-smooth. Everything will transmit automatically, wirelessly, and without the user's attention, to the user's personal smart device. This experience truly will showcase the user how modern technology will make journaling so much easier and revolutionize the activity.

[1] Theoretical calculation made under operation of taking the picture with interval of 20 minutes non-stop

2. Design

2.1 Subsystem Design

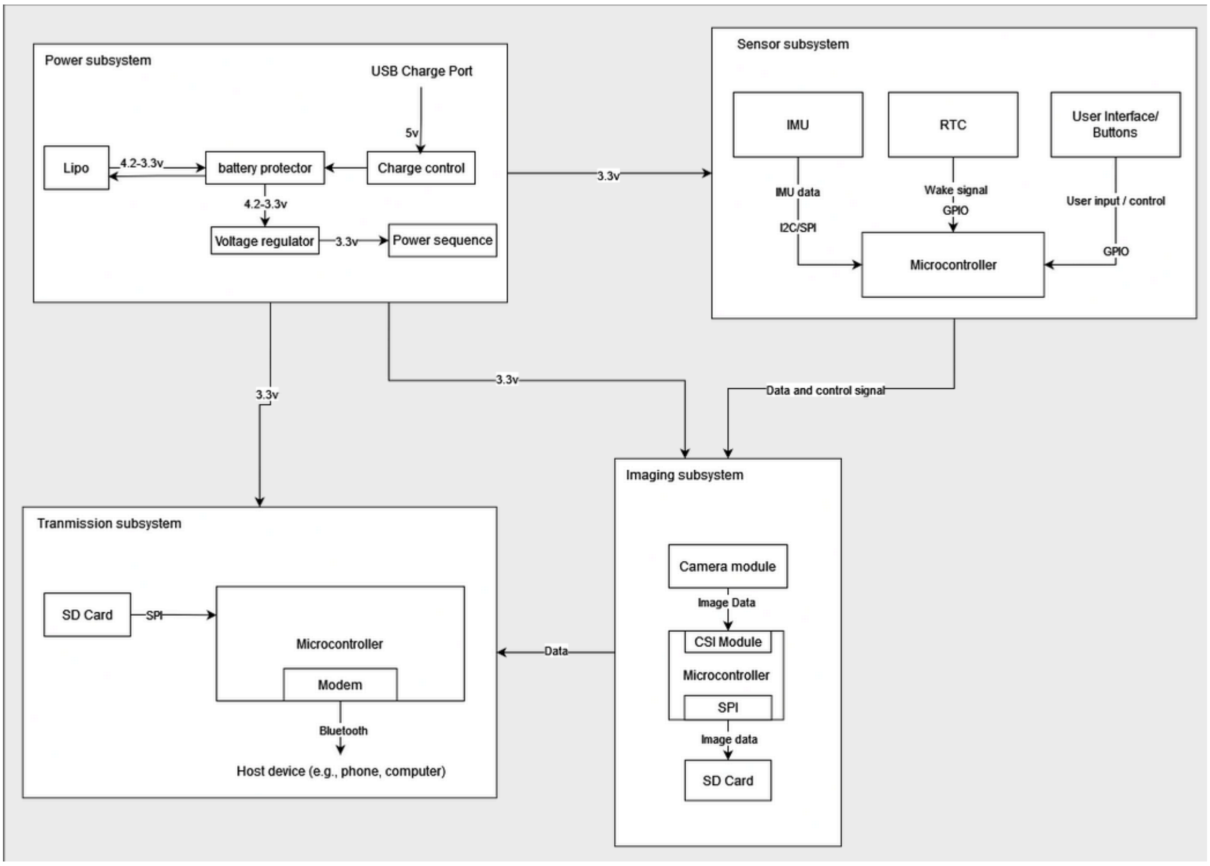

Figure 2.1.1, the initial subsystem design

We began our thought process by dividing the project into a few different subsystems that implement their own functionalities. In the design, we have 4 different subsystems: Power subsystem, Sensor subsystem, Transmission subsystem, and Imaging subsystem. The Power subsystem's role is to deliver correct voltages to each subsystem, handle power sequencing, as well as handle battery charging. The Sensor subsystem is the way our device interacts with the world, which includes an RTC to wake up the microcontroller periodically, a few buttons to take in user inputs, and an optional IMU to measure the movement of our device. The Transmission subsystem is primarily meant to send the image data over to the user's personal smart device. And finally, our imaging subsystem was designed to capture and buffer the image from the camera sensor.

2.2 Core Component Selection

To achieve our goals, we believe designing around a microcontroller and supplying it with peripherals of camera sensor, RTC, and user interactable switches such as buttons would achieve this goal. The microcontroller was determined before other parts of the design since it plays an more important role. In the search of our microcontroller, we followed the requirements below to select the most suitable model:

- Has hardware for BLE, I2C, SPI, Clock generation, and GPIO
- Support big enough RAM to host at least 1 1920x1080 image (~3MB), best to support external RAM
- Has hardware to receive camera parallel data input
- Has low sleep power draw
- Cost efficient

Eventually, we landed our search on the ESP32 family of microcontrollers as they checked all the boxes. In our first few iterations, we applied a ESP32-S3FN8 microcontroller to the design, then later switched to a ESP32-D0WD-N4R8 based solution due to cost. The most important feature of this line of microcontrollers is the fact they support external PSRAM which allow us to further expand the onboard memory to a maximum of 16MB, which is plenty enough for our design.

The camera sensor was the next in the line to be narrowed down. Our requirements were as such:

- High definition, support at least 1920x1080 resolution
- Works with our microcontroller
- Advanced image processing on-board
- Has low sleep power draw
- Cost effective

This led us to the OmniVision OV5640 sensor with fisheye lens. We eventually picked a variant of the sensor packaged with all its pins fanned out using Flexible Printed Circuit to be more cost effective as we would not have issues in ordering PCB with features such as via-in-pad and Ball Grid Array footprints.

Finally, we picked a power management solution based on the requirement below:

- Support Lithium Polymer battery
- Charges the battery to 4.2V
- Takes 5V input
- Can deliver 3.3V@200mA, 1.8V@150mA, 2.8V@150mA simultaneously
- Low transient current

We first landed on a Texas Instrument power management integrated circuit TPS65014, but in a few iterations found it to be costing too much. We then switched to a different PMIC, BQ25176, combining with a few voltage regulator MCP1700 chips, to achieve the same goals, mainly due to the fact we do not need that many features on the PMIC to lower the cost.

2.3 Hardware Design

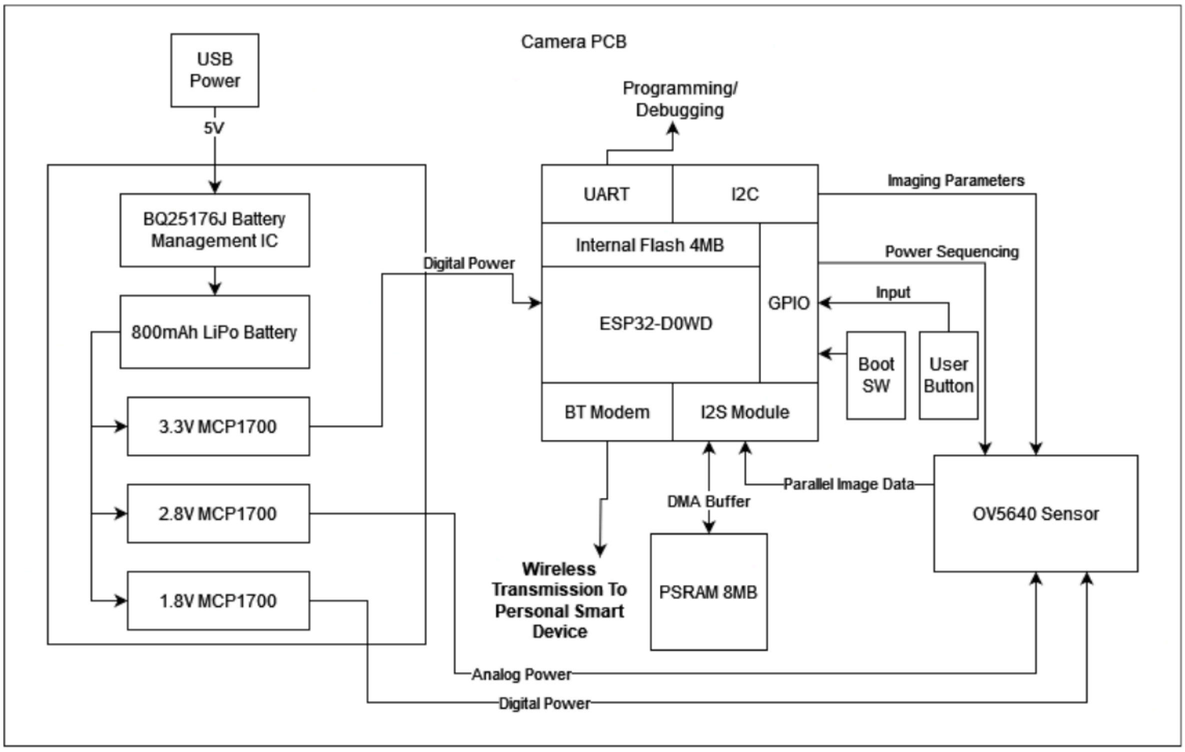

Figure 2.3.1 Final hardware design. The diagram above shows the final PCB design after 5 iterations of our PCB.

3. Prototyping Process

To create a working prototype, we have gone through 5 different revisions of different hardware designs. In this section, we will showcase our process.

3.1 Rev. A

This revision mainly focused on the bring-up of hardwares. Since our design involved high-speed signals, large amounts of parallel data, synchronized communications, RF design, VQFN and WQFN packaged chips, and ribbon cable connectors, we could barely prototype on a breadboard. The solution was to design modular bring-up boards mainly to focus on verifying the hardware design and functionality testing of the components. We designed 3 different PCBs that could be connected with cables and hand assembled them for testing.

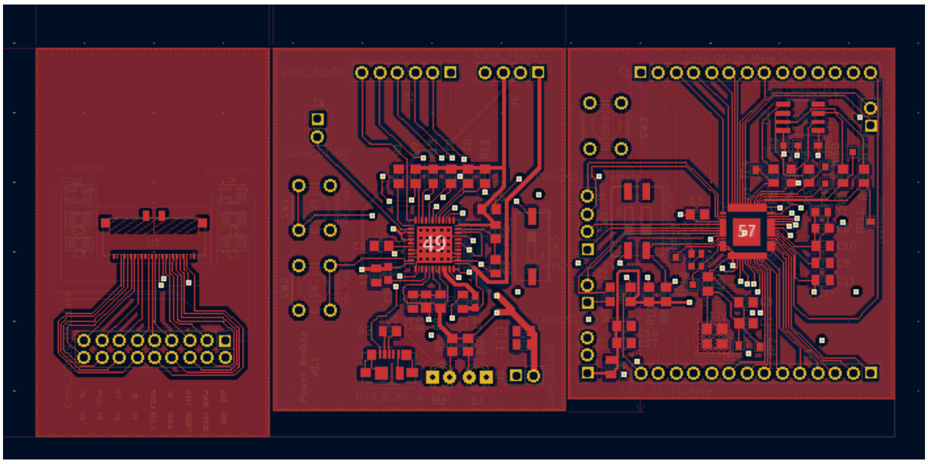

Figure 3.1.1, PCB design of Rev A combined, To lower production costs, we applied V-cuts between the PCBs so we could separate them when they arrive.

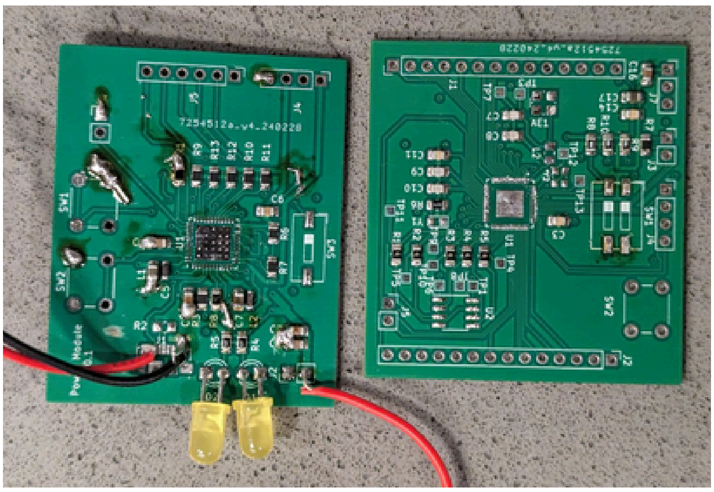

Figure 3.1.2, 2 Physical PCBs. Chips were removed and reused in the next revision to lower costs.

The PCB in the first revision had some major issues, such as the ground network was disconnected causing the circuit unable to function. The intentional test points and the usage of 0805 packaged components proved to be a great design consideration in this iteration; they helped us to make hardware debugging much easier. In the image above, many solder bridges and wires were applied to correct those mistakes and proceed with testing. These experiences greatly assisted us to improve our design on the next iteration.

3.2 Rev. B

In this revision, we focused on correcting the issues from the previous revision. The goal of this revision was to get a piece of simple camera demonstration code provided by the manufacturer to run on the hardware.

A major hardware swap happened during this revision, so that 2 versions were produced in this stage.

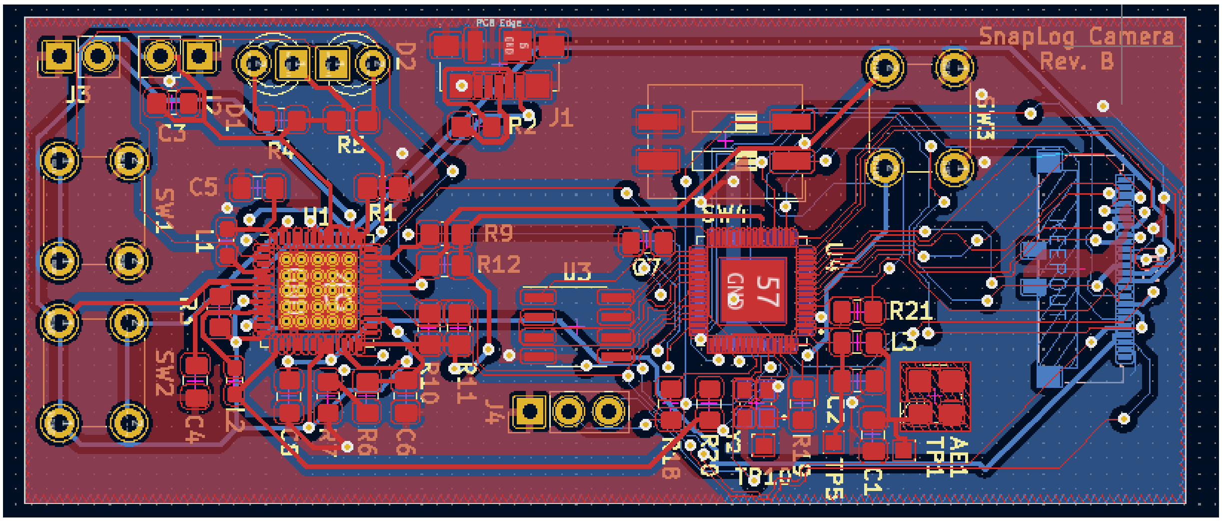

Figure 3.2.1, Rev. B.1 PCB design

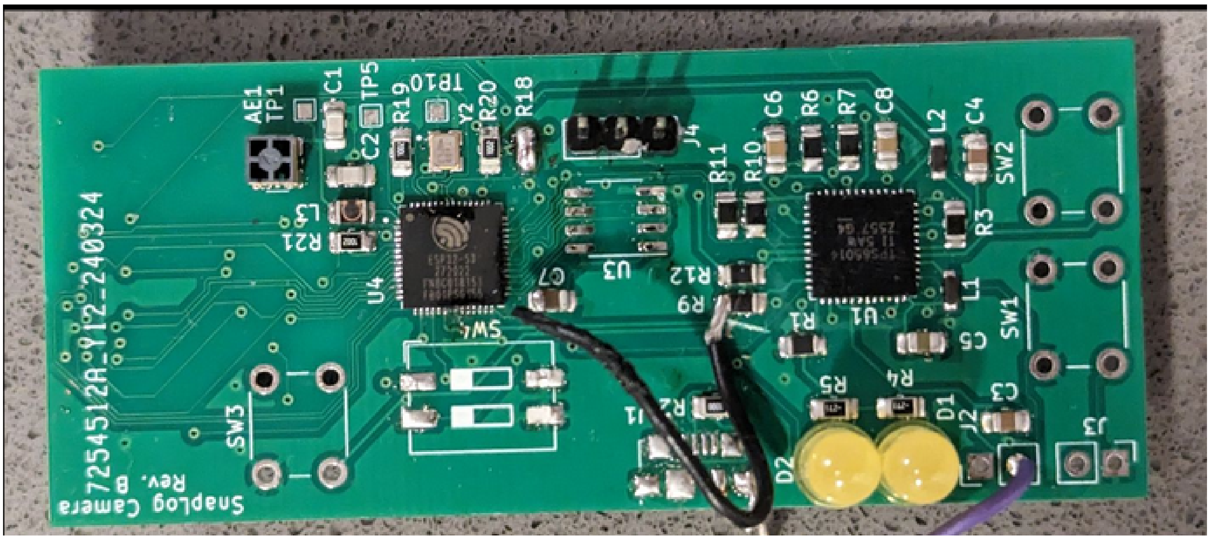

Figure 3.2.2, Rev. B.1 PCB

The first version of the revision was similar to the combination of 3 bring-up boards. This greatly improved the form-factor of the board and corrected plenty of mistakes from the previous revision. This board was produced and debugged in the same fashion as Rev. A, and minor issues still exist in this revision. Due to cost, we had to swap out the ESP32-S3 and TPS65014 to the other solution, which will be discussed in the next paragraph.

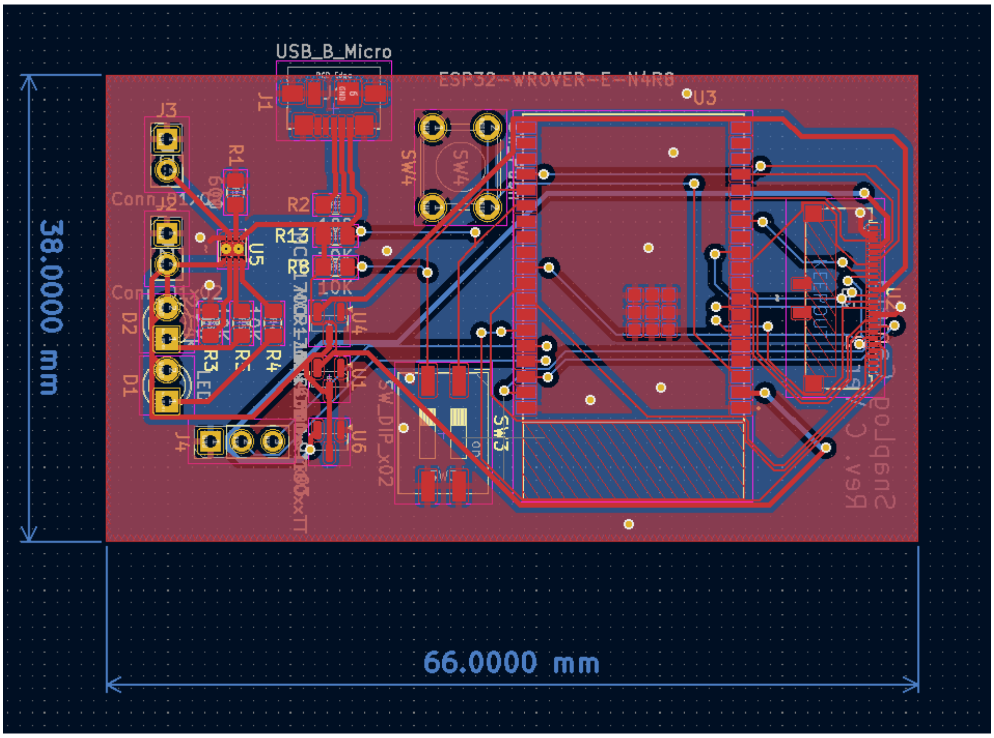

Figure 3.2.3, Rev. B.2 PCB design

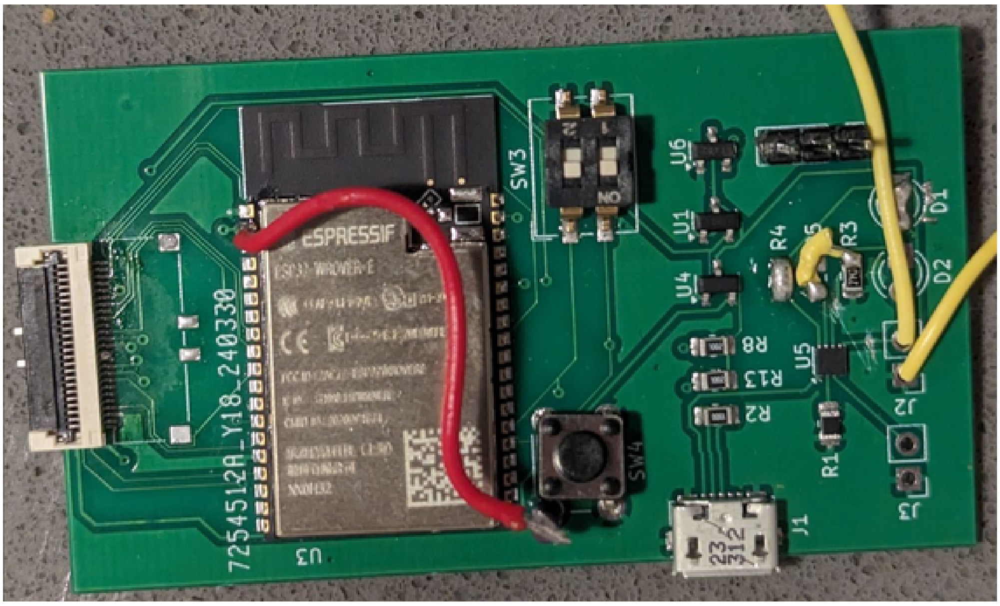

Figure 3.2.4, Rev. B.2 Physical PCB

This design we switched the microcontroller to the older ESP32-D0WD-N4R8 which costs half the price as ESP32-S3 and BQ25176 power management IC that costs one-seventh the price of a TPS65014. Beside the cost, another advantage of this design is that it further improved the form-factor of our hardware, since these chips require less external passive components. We intentionally left some blank spaces to help us make debugging the hardware easier.

3.3 Rev. C

This revision corrected all the issues we ran into in all the previous revisions. The final revision performed as we intended, and is able to perform all required functionalities as intended.

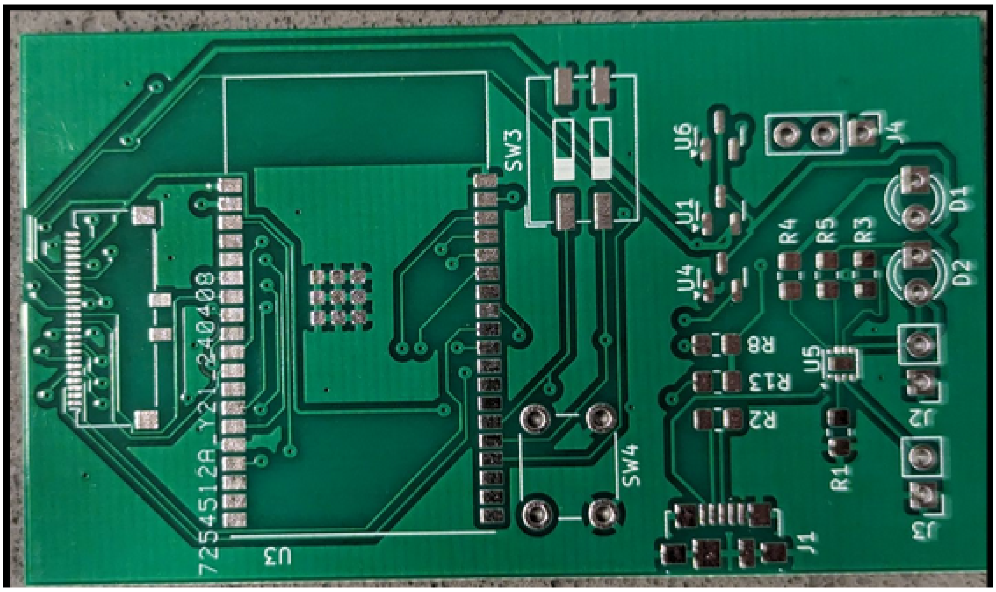

Figure 3.3.1, Blank design of Rev. C PCB

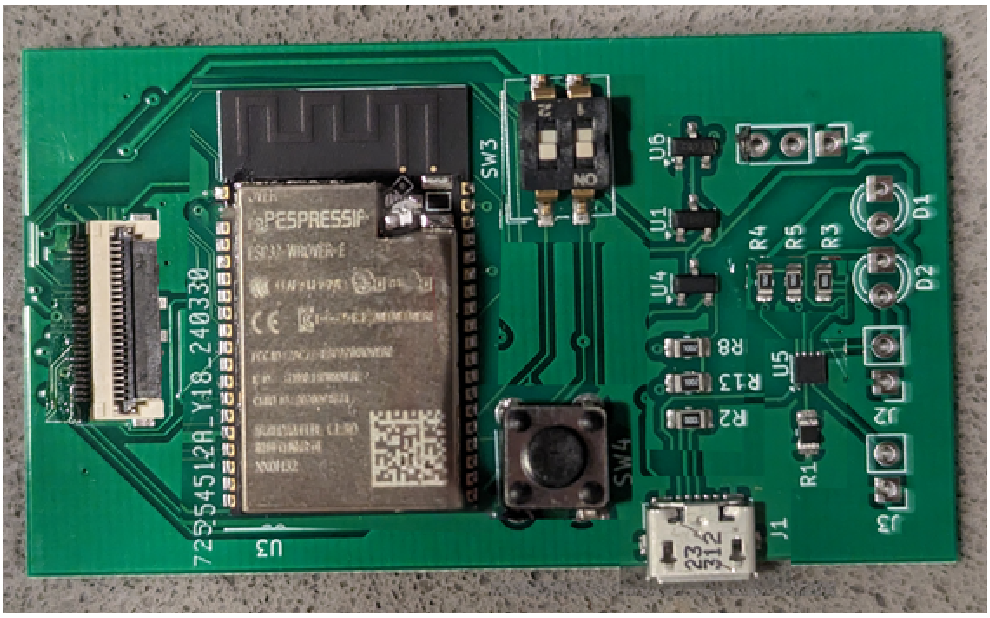

Figure 3.3.2, A concept image of what it looks like after fully assembled.

Due to the time limitations, we enclosed our design before preserving any media involving the physical PCB after soldering. The image above was produced as a demonstration resembling the physical PCB after assembly.

3.4 Professional Enclosure

Our team designed an enclosure to enhance the aesthetics and functionality of SnapLog using industry-standard Computer-Aided Design (CAD) software. The enclosure was fabricated using state-of-the-art 3D printing technology, ensuring precision and durability in the final product. We assembled the enclosure using consumer electronic grade adhesives, adhering to industry standards for quality and reliability.

By employing professional design and fabrication techniques, we have not only enhanced the visual appeal of our product but also ensured its longevity and ease of maintenance, ultimately delivering a high-quality and user-friendly solution.

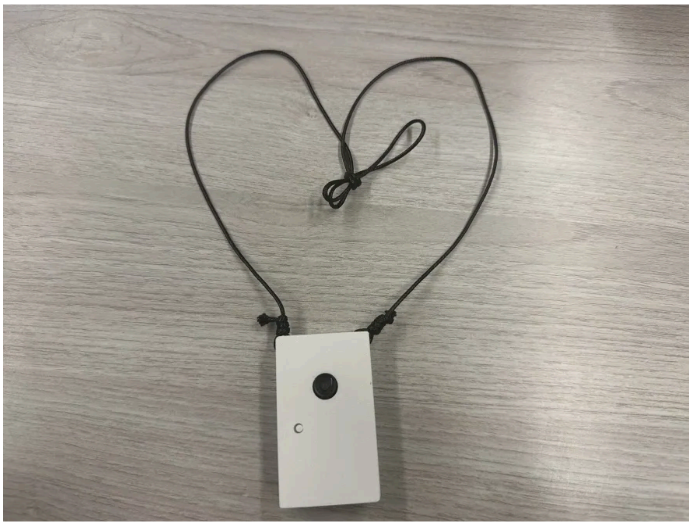

Figure 2.4.1 Final design of the camera necklace

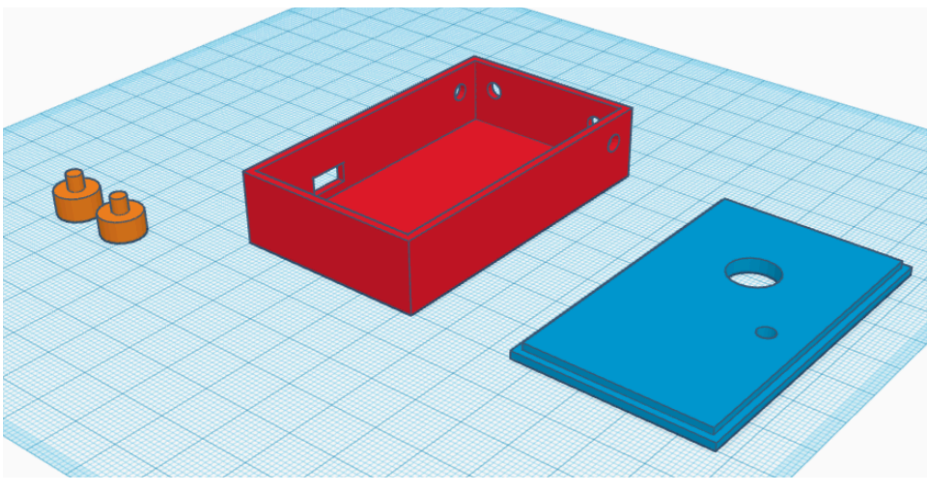

Figure 2.4.2 CAD model of enclosure

3.5 Software

3.5.1 Transmission Protocol Design

SnapLog involves two ways of transmission between itself and the host machine, such as a laptop. To facilitate seamless communication, we self-designed a robust transmission protocol.

1. **SnapLog to Host Machine**: Utilizing Bluetooth Low Energy (BLE) technology as the underlying communication framework, SnapLog wirelessly transmits images from its cache to the host machine. Building upon BLE, we have developed a self-designed transmission protocol tailored to the specific requirements of SnapLog. This protocol enhances data reliability and efficiency during transmission, ensuring that images are delivered securely and promptly to the host machine.

2. **Host Machine to SnapLog**: Our graphical user interface (GUI) empowers users to set the desired image capture interval, providing flexibility and customization options. To maintain data integrity and minimize transmission errors, the host machine reports any packet loss encountered during communication with SnapLog. Moreover, our transmission protocol facilitates synchronization between the host machine and SnapLog, ensuring seamless data exchange and optimal performance throughout the image capture process.

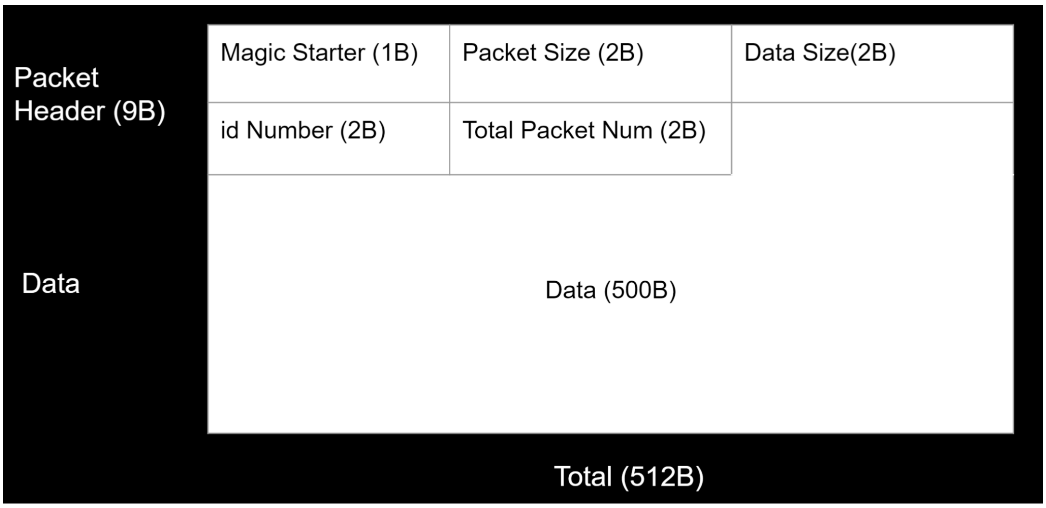

Figure 3.5.1.1 Packet Structure of Our Self-Designed Transmission Protocol

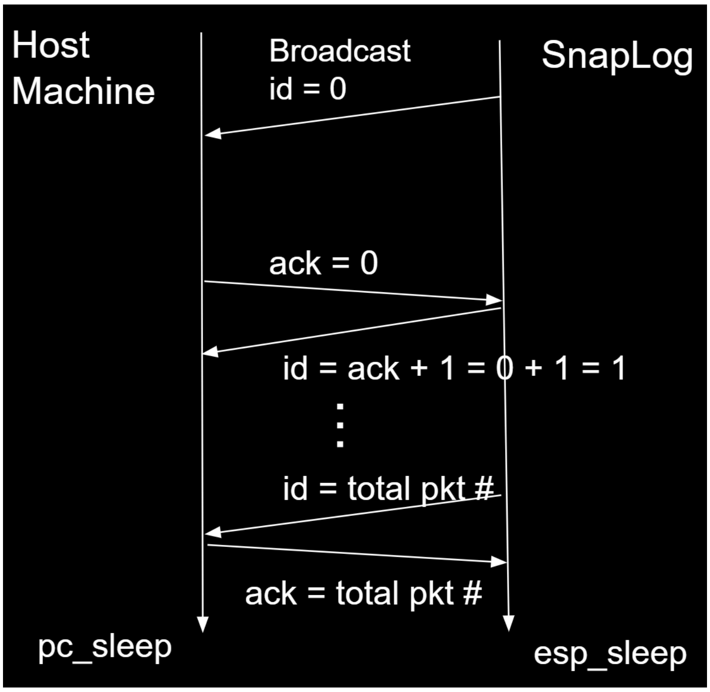

Figure 3.5.1.2 Handshaking of Our Self-Designed Protocol

3.5.2 Graphic User Interface

We programmed a user-friendly graphical interface (GUI) that allows users to easily input time intervals for picture taking and seamlessly create captivating timelapse videos using the photos captured by SnapLog.

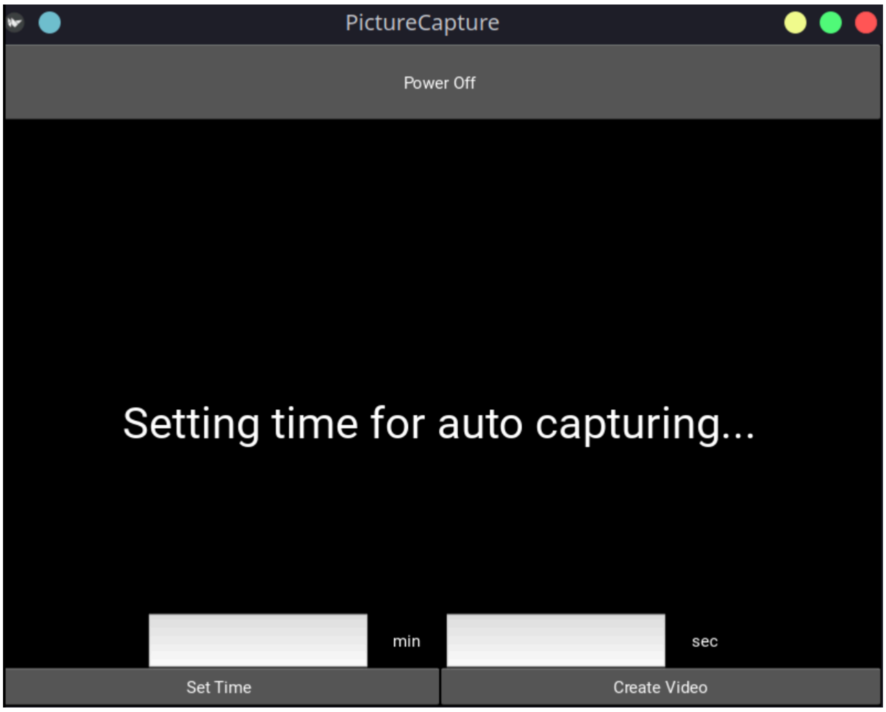

Figure 3.5.2.1 GUI for Setting Camera Parameters

4. Design Verification

4.1 Camera Subsystem:

1. **Camera Operation Requirements and Verification**:

- Requirement: The camera must be capable of taking pictures and sending them to the host device within 5 minutes.
- Verification: Test the camera's image capture and transmission time to ensure that images are received by the device within the specified time.

2. **Automatic Image Capture**:

- Requirement: The camera should automatically take pictures at a configurable interval not exceeding 5 minutes.
- Verification: Configure the camera to take pictures at intervals of 1 minute, 3 minutes, and 5 minutes, and verify that it functions as set without user intervention.

3. **Manual Image Capture Capability**:

- Requirement: The camera must have an immediate picture-taking function triggered manually by the user.
- Verification: Manually trigger the camera to take a picture using a button and verify that the image is captured and processed instantly.

All requirements were achieved.

4.2 Power Subsystem:

1. Power Requirement:

- Requirement: The power subsystem should supply at least 3.3v 200mA to other subsystems.
- Verification: Measure the voltage and current across the resistor connected to the output of the power subsystem to ensure it meets these requirements.

2. Manual Power-Off Function:

- Requirement: The camera must include a manual power-off function that can be activated by the user.
- Verification: Test the manual power-off functionality by using the power-off button and confirm that the camera powers down immediately without issues.

All requirements were achieved.

4.3 Wireless Transmission Subsystem:

- Requirement: The imaging subsystem should generate raw imaging data that can be read by the microcontroller.
- Verification: Verify that the data output from the imaging subsystem is readable and accurate by the microcontroller.
- Requirement: The transmission subsystem should wirelessly transmit images to other devices using Bluetooth.
- Verification: Check the receiving program on the host device to confirm that it correctly displays images transmitted from the camera via Bluetooth.

All requirements were achieved.

4.4 Software Subsystem:

- Requirement: The software on the host device must convert pictures taken in the last 24 hours into a video that can be viewed on the device.
- Verification: Use the host device's software to process images from a 24-hour period and check if it successfully creates a video.

All requirements were achieved.

5. Costs

5.1 Parts

We bought all the parts from mouser. The following table lists the cost of components of a single unit of our final design.

Table 5.1 Parts Costs

Part Mfr#	Retail Cost ($)
ESP32-WROVER-E-N4R8	$2.82
MCP1700T-3302E/TT	$0.42
MCP1700T-1802E/TT	$0.51
MCP1700T-2802E/TT	$0.50
BQ25176JDSGR	$1.32
RTS103C1R2M4L151	$3.19
CR0805-FX-2742ELF	$0.10
CRCW0805600RFKEAHP	$0.46
47346-0001	$1.07
OV5640-AF-FPC	$4.71
F34G-1A7Q1-E8C24	$0.47
560112120004	$0.14
RCS08051M00JNEA	$0.137
2454985-2	$0.85
Total	**$16.697**

It is worth mentioning the cost above only reflect the cost of a single unit of our final design, <u>not the cost of the entire R&D.</u> We mainly paid out-of pocket for this project, and for each revision we need to at least multiply the cost by a factor from 3 to 5. The total cost, involving

PCB, is around $600. This also excludes the cost of our recycled parts, such as the battery was recycled from our personal drone's battery pack.

5.2 Labor

The primary cost for this project is labor. The engagement of three highly skilled individuals significantly contributes to the overall labor cost. Based on survey-reflected industry standards, the labor rate is estimated at $200 per hour. With each team member dedicating 40 hours per week, and considering a project duration of 12 weeks with an additional 2 weeks allocated for paid time off, the total labor cost is calculated as follows:

$$\begin{aligned} Labor\ Cost &= ideal\ salary(hourly\ rate) \cdot actual\ hours\ spent \cdot 3 \\ &= 200\ \$/h \cdot 40\ h/week \cdot 12\ weeks \cdot 3 people \end{aligned}$$

This leads to $96,000 pre-tax cost for labor.

6. Conclusion

Overall, the project is successful. We not only met but exceeded all high-level requirements outlined in the design document. Our innovative solution successfully addresses the core objectives, delivering a product that fulfills its intended purpose and offers additional value to users.

6.1 Accomplishments

1. **Efficient Wireless Transmission**: The camera successfully captures and wirelessly transmits photos to the host device within a 5-minute delay. To achieve this, our team implemented BLE for efficient transmission. Additionally, we developed a robust data transmission protocol to ensure reliable data transfer. In the end, the transmission delay is approximately 30 seconds surpassing the initial proposed 5 minutes The microprocessor efficiently wakes up from hibernation to communicate with sensors and transmit parameters over SCCB, achieving remarkable battery life of up to 533 hours with an 800mAh battery.

2. **Automatic Picture Capture**: The camera autonomously captures pictures at configurable intervals, meeting the requirement of a maximum 5-minute interval. To enhance user experience, we integrated an intuitive GUI for seamless interaction with the hardware.

3. **Video Compilation Software**: The software on the host device seamlessly converts pictures taken within the last 24 hours into a video format. This feature enables users to easily view and enjoy a timelapse of their daily activities. By integrating these functionalities, our project successfully simplifies the process of documenting and reliving daily moments, offering users a convenient and enjoyable tool to preserve and cherish their memories.

6.2 Future work

In the next iteration of SnapLog, several enhancements and developments are planned to further refine and expand the capabilities of the device.

1. **Size Optimization**: One key aspect of the future work involves decreasing SnapLog's size by half. This optimization is feasible as many designs on the final version of the PCB were for testing purposes, and we only utilized one side of the PCB.We aim to create a more compact and portable device without compromising functionality.

2. **Mobile Application Development**: Currently, SnapLog's GUI is only compatible with Mac, Windows, and Linux desktops. As part of future work, we will develop a dedicated mobile application for smartphones, enhancing the accessibility of SnapLog.

3. **Market Opportunities**: Additionally, we will actively explore opportunities to bring SnapLog to market. This involves conducting market research to identify potential target demographics, assessing competitors, and developing a comprehensive marketing strategy.

References

[1] "ESP32S3 Series." Mouser Electronics, https://www.mouser.com/datasheet/2/891/esp32_s3_datasheet_en-2946743.pdf.

[2] "ESP Hardware Design Guidelines - ESP32-S3 - — ESP Hardware Design Guidelines latest documentation." Technical Documents, https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32s3/index.html. Accessed 1 May 2024.

[3] ESP-WROVER-KIT V4.1 Getting Started Guide - ESP32 - — ESP-IDF Programming Guide v5.2.1 documentation, https://docs.espressif.com/projects/esp-idf/en/stable/esp32/hw-reference/esp32/get-started-wrover-kit.html. Accessed 1 May 2024.

[4] "TPS65014 Power- and Battery-Management IC for Li-Ion Powered Systems" Texas Instrument, https://www.ti.com/lit/ds/symlink/tps65014.pdf?ts=1714624541300.

[5] "BQ25176J: 800-mA JEITA-Compliant Linear Battery Charger For 1-Cell Li-Ion and LiFePO4 With VINDPM" Texas Instrument, https://www.ti.com/lit/ds/symlink/bq25176j.pdf?ts=1714564471961.

[6] ESP32 Camera Driver, https://github.com/espressif/esp32-camera. Accessed 1 May 2024.