MULTISPECTRAL IMAGING CAMERA

By

Isha Akella
Amartya Bhattacharya
Jason Jung

Final Report for ECE 445, Senior Design, Fall 2023
TA: Jason Zhang

6 December 2023
Project No. 06

Abstract

This paper explains the concept, design process, and results of the Multispectral Imaging Camera, pitched by Professor Gruev, which aims to achieve the ability to view visible, NIR, and UV spectra in the form of a small, handheld device to aid in faster medical diagnosis. This paper will begin with an overview of the problem the device aims to solve, followed by the design choices and changes made throughout the semester, and ending with the testing, challenges, and accomplishments of the device in tandem with the development board. Overall, multispectral imaging was accomplished through the development board, and limited functionality was demonstrated through the PCB.

Table of Contents

1. Introduction

1.1 Problem & Solution

What humans can see is subjective, inconsistent, and, most importantly, limited. In several fields, we rely solely on human vision to determine problems and solutions; however, some areas require higher resolution, wider spectral ranges, and consistent data. In a medical context, the ability to capture a variety of spectra, including those invisible to the naked eye, at a consistently accurate level can improve the assessment abilities of a medical professional, especially in surgical tasks. According to SpectralMD, limited tools are currently available to determine tissue health [1]. Multispectral imaging (MSI) can help alleviate this issue as doctors will be able to use light emitted in UV (ultraviolet) and NIR (near-infrared) spectrums to accelerate tissue diagnoses. Additionally, combining color imaging with NIR bands can help to locate and distinguish between tumors and surrounding tissues.
Existing endoscope technology typically uses RGB endoscopes, which are confined to the visible spectrum, thus limiting the number of tissue layers it can show as visible light cannot "penetrate through the surface of the skin" [2]. A multispectral camera will allow "the physician to extract intrinsic properties and structures of specific tissues which are not visible to a human eye" [2]. MSI will allow for improved real-time diagnosis rather than invasive, time-consuming procedures. As a direct result, patient care can be expedited with a faster diagnosis. MSI enhances inspection capabilities for various applications and fields but can potentially be a very beneficial tool in the medical field.

Our solution will be a handheld device with an integrated camera sensor capable of multispectral imaging across the UV, visible, and NIR spectra with real-time visualization. NIR wavelengths go much further into tissue, so the ability to capture that data is crucial. A user will be able to hold the sensor above any object for which light can pass through and see any of the three different spectra displayed on a monitor. The device will be wired for power and contain a MIPI CSI-2 (Mobile Industry Processor Interface Camera Serial Interface 2) compatible image sensor and microcontroller that is USB (Universal Serial Bus) compatible for data transfer [3]. The image will be processed to show three different spectra (UV, NIR, and visible) on a real-time video display. Our solution is novel compared to other medical handheld imaging devices, such as endoscopy cameras, due to the ability to capture multiple spectra in such a small device. It will enable physicians to use the MSI technology in closer proximity than before, making diagnosis a more straightforward and less painless process. More information when imaging will empower medical staff to make more informed clinical decisions.

1.2 High-Level Requirements

To consider our project successful, our device must:

1. Display real-time video at a rate above 20 frames per second so that the image on the screen is accurate to the location that the device is hovering over.
2. Can capture signals across the UV, visible, and NIR spectrums to achieve multispectral imaging.
3. Have the maximum dimensions of a 2.25" by 2.25" by ~7" handheld enclosure to be compact enough to be able to be used in a variety of different medical applications.

All three requirements come together to create a handheld, multispectral camera that can display information at a rate high enough so that the user has real-time imaging.

1.3 Visual Diagram

The image below (Figure 1) shows a general description of how multispectral imaging works. A multispectral camera takes in images from a sample illuminated by a light source. These images are then processed to show the different spectrums captured.

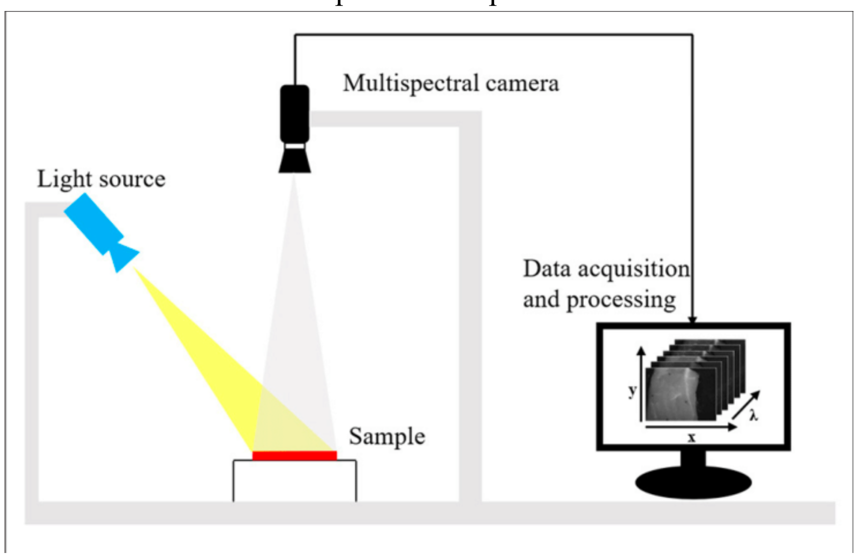

Figure 1: Multispectral imaging general design

1.4 Subsystem Overview

1.4.1 Block Diagram

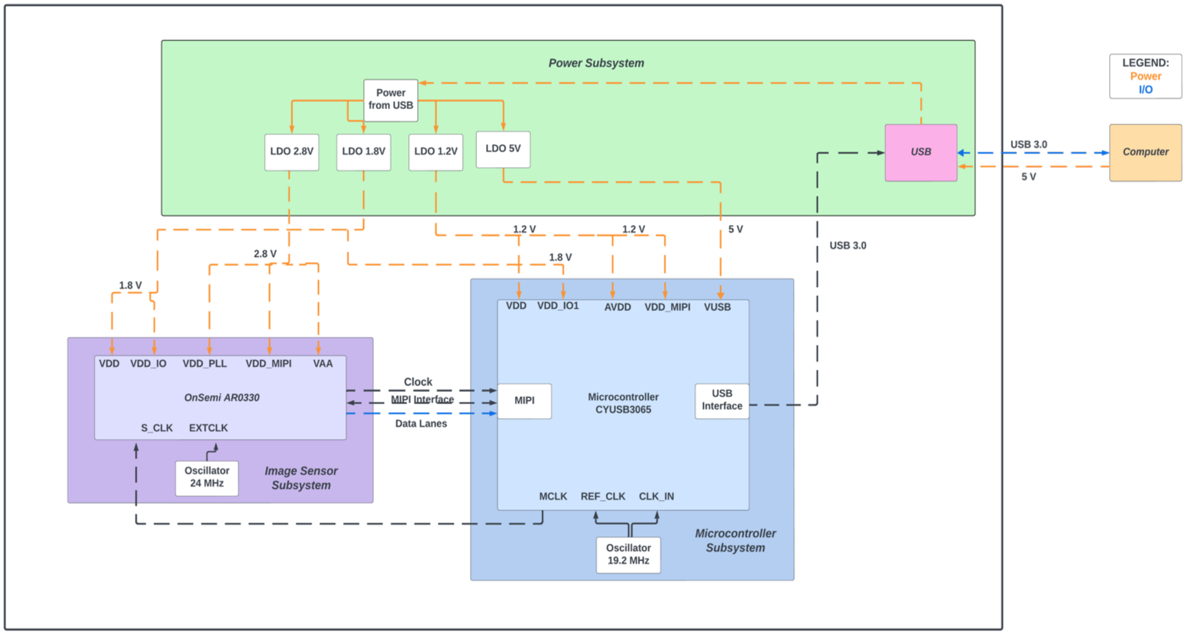

Figure 2: Block Diagram

Our project has three critical subsystems: Power, Image Sensor, and Microcontroller as shown in Figure 2 above. The power subsystem supplies power at the appropriate levels to the rest of the board using voltage regulators. The image sensor subsystem captures the incoming light to the pixels and sends it to the microcontroller subsystem, where the data will be processed and sent to the computer via USB that will display the video. No block-level changes were made to the design.

1.4.2 Power Subsystem

The power subsystem is responsible for supplying power to all other components of the project. Power is provided over USB 3.0 in the USB-C connection and distributed to the rest of the system through individual regulators per power pin. The USB-C connection will provide 5V; however, this is higher than what is necessary for most of the components on the board. A total of seven voltage regulators will be used to ensure the correct voltage requirements for each component.

1.4.3 Microcontroller Subsystem

The microcontroller subsystem processes the image data from the image sensor subsystem and prepares the video data to be sent to the monitor over the USB 3.0 interface. The MCU can run programs that exclude the image sensor; however, the sensor will not receive any clock signal without the MCU. The microcontroller used in this project will be the Infineon CYUSB3065, a BGA (ball grid array) component.

1.4.4 Image Sensor Subsystem

The image sensor subsystem is responsible for capturing image data across the UV, VIS, and NIR spectra and then sending this data to the microcontroller over MIPI (Mobile Industry Power Interface) CSI-2 (Camera Serial Interface 2) to be processed. The image sensor has a resolution above 1 MP and a quantum efficiency high enough to feasibly detect signals in the NIR, UV, and visible spectrums. The image sensor chosen is the OnSemi AR0330CM, a QFN (quad flat no-lead) component.

2 Design

2.1 PCB Design

Our PCB design was a 4-layer board in a circular shape, as shown in Figure 3. The layers were (1) Signal, (2) Ground, (3) Power, (4) Ground. The PCB has a diameter of 2" to fit into the enclosure created, and a USB-C receptacle to transmit power and data to each of the board's respective subsystems.

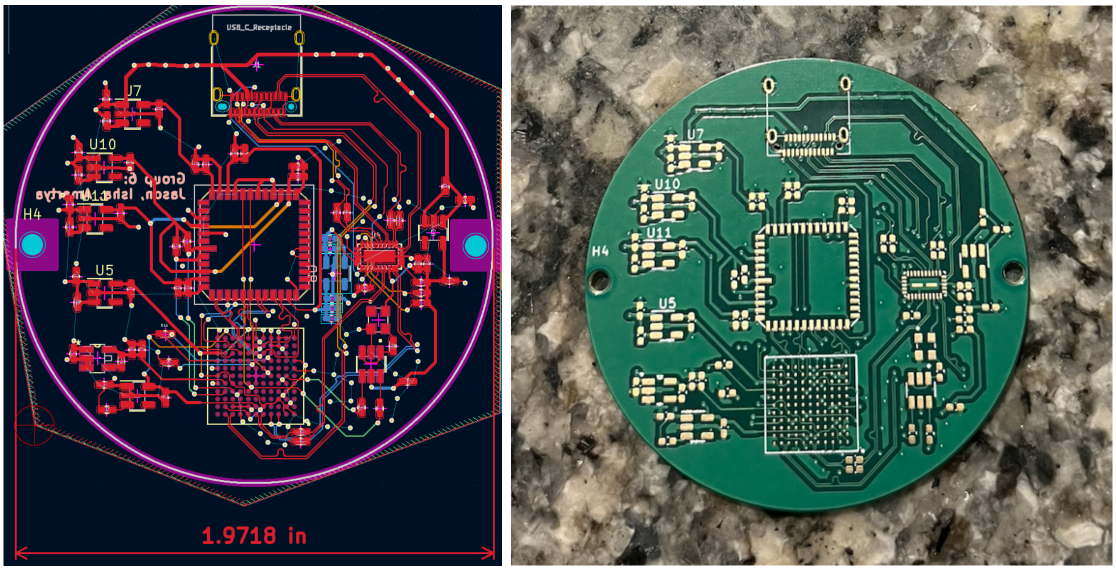

Figure 3: PCB layout design and manufactured product

2.1.1 Design and Justification

A 4-layer board was chosen due to the complexity of the integrated circuits that we were using to support the high-speed digital protocols USB 3.0 and MIPI CSI-2. The stack-up of the PCB was chosen to ensure proper grounding planes with each signal being routed to ensure signal integrity throughout the circuit. Furthermore, trace widths were calculated using the IPC 2141 standard so that impedances were matched for the standards used [4]. Equation 1 shows the formula we followed, where w is the trace width, Z is the target impedance, t is thickness of the board, and h is copper filling.

$$w \approx \frac{7.85h}{e^{Z_o \frac{\sqrt{\varepsilon_r + 1.41}}{8.6}}} - 1.25t \tag{1}$$

$$w \approx \frac{7.85h}{e^{Z_o \frac{\sqrt{\varepsilon_r + 1.41}}{8.6}}} - 1.25t$$

2.1.2 Design Alternatives

An alternative approach that could have been done is that the USB-C receptacle could have been placed on the back side of the PCB. This would have saved a lot of space on the PCB and have allowed us to get the PCB into a smaller form. However, the concern with this design choice was that more issues with signal integrity would have been introduced. For the purposes of a design that would meet our requirements, we decided to keep it on the front side and potentially place it on the back for future iterations.

Another design choice was that we could have also used more layers in our PCB design. This would have made routing easier to handle and would have minimized any potential forms of cross-talk and parasitic capacitance by having separate layers for analog voltages, digital voltages, and digital signals. However, considering that we were able to execute a design using 4 boards, adding more layers would have resulted in diminishing returns for an initial prototype.

2.2 Enclosure Design

Our design uses a custom 3D printed enclosure with a handheld design and space for the USB connection and image sensor. Figure 4 shows the CAD model we designed, and Figures 5 and 6 demonstrate different physical aspects of the printed model. The USB port is contained within the lid of the enclosure, and the wire, once connected, will wrap around the handle, threading out to a computer to interface with the PCB. Additionally, an opening in the lid was made to expose only the image sensor.

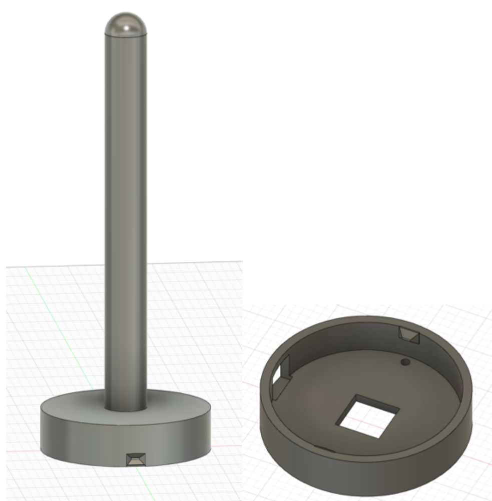

Figure 4: Enclosure CAD Design

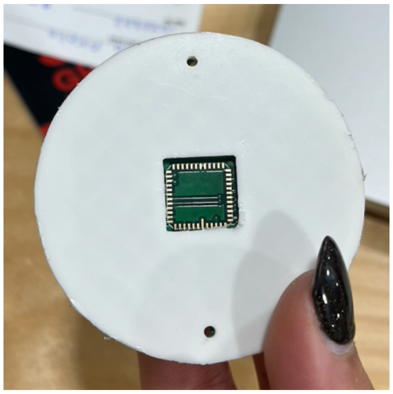

Figure 5: Enclosure Lid Exposing Image Sensor

Figure 6: Final Printed Enclosure

2.2.1 Design and Justification

3D printing of the enclosure was chosen due to the desired size constraints and intended shape. For the device to be handheld and comfortable, a thin radius handle was chosen. Additionally, due to the size of the enclosure, we were unable to use threads to secure the base to the lid as they would not print in high enough definition. We adjusted our design to use snap-fit components that would allow us to remove the PCB when necessary and easily snap into place. Since our device was not being tested in a medical setting, it was printed with a PLA filament, a commonly available filament used for 3D printing that is durable and highly functional for everyday handling.

2.2.2 Design Alternatives

For future iterations of the enclosure design, the handle could be made more ergonomically for people with limited mobility in their fingers or wrists. Additionally, rather than using snap-fit components, which can be fragile when printed on a low-quality printer, having a base/lid structure that screws into place may be a better option. Finally, since the motivation is to use this device as a medical-grade electronic, usage of this device in a clinical setting will require a sterile equipment cover that meets FDA Title 21 (21CFR820) & ISO 13485 standards. See Section 5.3 for more details.

2.3 Power Subsystem

As mentioned in section 1.4.2, the power subsystem is responsible for supplying appropriate power through voltage regulators in the USB-C connection. See Figures 14-16 in Appendix C for the specific schematics.

2.3.1 Design and Justification

Due to the nature of the USB-C connection, a mux was required to determine the proper orientation of the data pins. Unlike USB-A or Micro-B connections, USB-C can be plugged into a device in two separate directions, so the mux determines the orientation and passes the relevant data lines through to the rest of the board. USB-C was chosen over the other options as it is a smaller, future-proof USB version.

Additionally, since USB supplies 5V of power, which was higher than the required voltage for almost all components on the board, several voltage regulators were added. There were four different required voltages for the entire board, however, for the larger components, such as the MCU and image sensor, rather than placing all the burden on one LDO (low dropout voltage regulator), we chose to use two, so that each LDO would have its own current draw that did not exceed its thermal capacities. As a result, we had a total of seven voltage regulators.

Finally, we initially chose to use USB 3.0 due to the higher data transfer speed compared to USB 2.0. For future iterations of our project, we believe the high-speed transfer of 3.0 would be more compatible.

2.3.2 Design Alternatives

Although USB 3.0 and USB-C are preferred options for the final iteration of this project, for the purposes of ECE 445, and the initial version, USB 2.0 and USB Micro-B or USB-A could have been used. As discussed in further detail in 3.2.1, our group was able to enumerate our PCB as a USB 2.0 device rather than a USB 3.0 device. USB 3.0 is built off USB 2.0 but is more complex as there are more pins and data lines that must be wired. For the sake of prototyping, USB 2.0 has a high enough data transfer rate and would have allowed for a much simpler design. Additionally, the mux required by USB-C added complexity that, combined with USB 3.0, resulted in a board that did not function as intended and enumerated as a USB 3.0 device. The use of these two alternatives would decrease the complexity of the layout, increasing the likelihood of a successful prototype.

2.4 Microcontroller Subsystem

The microcontroller subsystem is responsible for communicating between the image sensor and the display over MIPI CSI-2 and USB. It is comprised of the Infineon CYUSB3065 microcontroller and voltage regulators supplied by the power subsystem. See Figure 18 in Appendix C for the MCU Schematic.

2.4.1 Design and Justification

The microcontroller that we used was the Infineon Cypress CX3 bridge controller [5]. The microcontroller had many attractive features and capabilities that were suitable for the requirements of this project. First off, it was compliant with the USB 3.0 Superspeed standard, and the MIPI CSI-2 interface. This allowed us to use the microcontroller as a bridge between the image sensor and the host PC at speeds that would support a large stream of image data. Furthermore, the microcontroller had a powerful CPU, GPIOs, and multiple debugging options. The latter was important to us because it gave us options to debug our firmware through a development board. Considering size as a constraint in combination with all the features that it provided, this microcontroller was an attractive option for the design. It would be suitable for future iterations and higher demands on computational power.

2.4.2 Design Alternatives

Design alternatives would have included using a less capable microcontroller, the advantage to this approach would be a reduction in cost as well as less complexity in the overall design. If more of a priority was taken to create a minimum viable product, there should have been a higher level of consideration for this option. However, replacing this with a less capable microcontroller would have inevitably resulted in another iteration with different standards and protocols and an overall redesign of our board. Furthermore, having a less capable microcontroller would have made developing the firmware more difficult. The reason for this is that the MIPI CSI-2 standard was designed with the intent of making image data easier to stream, there is a large system of support for this standard, and made the process of development easier overall.

2.5 Image Sensor Subsystem

The image sensor subsystem is responsible for capturing image data in the UV, visible, and NIR spectra. It is solely comprised of the OnSemi AR0330CM imager and voltage regulators supplied by the power subsystem. See Figure 17 in Appendix C for the image sensor schematic.

2.5.1 Design and Justification

The OnSemi AR0330 CMOS image sensor was chosen because it met the minimum requirements for multispectral sensitivity that we had set out for ourselves. Crucially, this included a 10% quantum efficiency at both 380 nm and 780 nm shown in Figure 7 [6], the critical ranges for ultraviolet and infrared wavelengths, respectively. On top of the quantum efficiency requirement, we also selected our imager based on its resolution and maximum frame rate capability, to ensure a clear, real-time video output. The AR0330CM can display images up to 3.15 megapixels in resolution, and stream video at up to 60 fps. Finally, we chose an image sensor that would interface with our microcontroller over the CSI-2 standard, a widely adopted, high-speed protocol for the transmission of still and video images from image sensors to application processors, developed by the MIPI Alliance. This protocol is the de-facto embedded camera and imaging interface and has support for a broad range of features tailored for high performance and low power data transfer, as well as low electromagnetic interference, making it perfect for our use case.

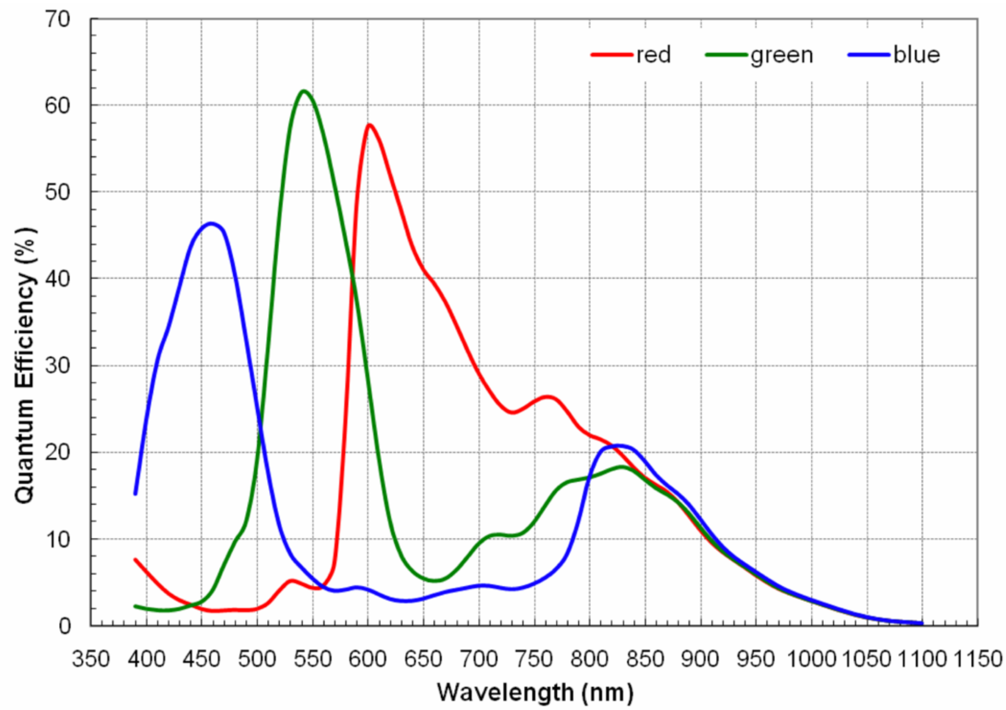

Figure 7: AR0330 Quantum Efficiency [6]

2.5.2 Design Alternatives

An alternative that we initially considered was the Sony IMX487 BSI (backside-illuminated) CMOS image sensor, Sony's latest multispectral product offering, released in September 2021. It has the industry's highest effective pixel count of 8.13 MP and is specifically developed for high UV and IR sensitivity, resulting in a resolution and quantum efficiency that far exceeds the capabilities of the AR0330, as shown in Figure 8 [7]. Unfortunately, due to how recently this sensor was released, it is still in the prototype phases for the various manufacturers and was not available for purchase from Sony.

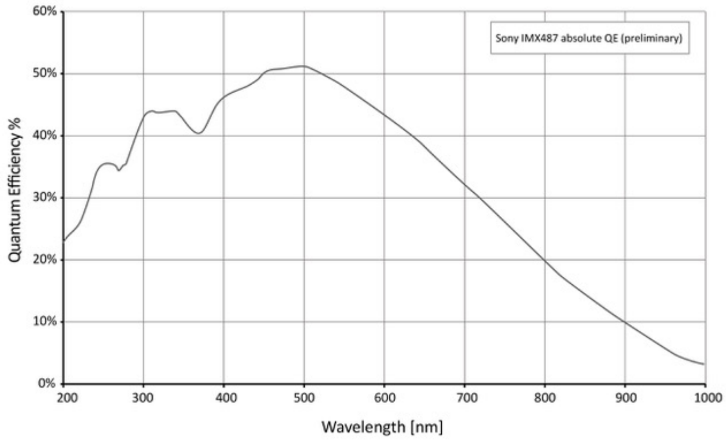

Figure 8: Sony IMX487 Quantum Efficiency [7]

3. Requirements and Verification

In this section, we will be discussing critical portions of our project as well as the requirements and verifications. The full requirements and verifications table can be viewed in Appendix A under Table 2.

3.1 Image Sensor

3.1.1 Frames Per Second

Our first high-level requirement necessitates that the device operates at above 20 frames per second to support real-time video. Although we were unable to test this requirement on our PCB, we were able to conduct the test on the development board. Figure 9 shows an image of a Webcam FPS Checker which we used on the dev board to determine that it operates at 60 frames per second, which is well above the required 20 fps.

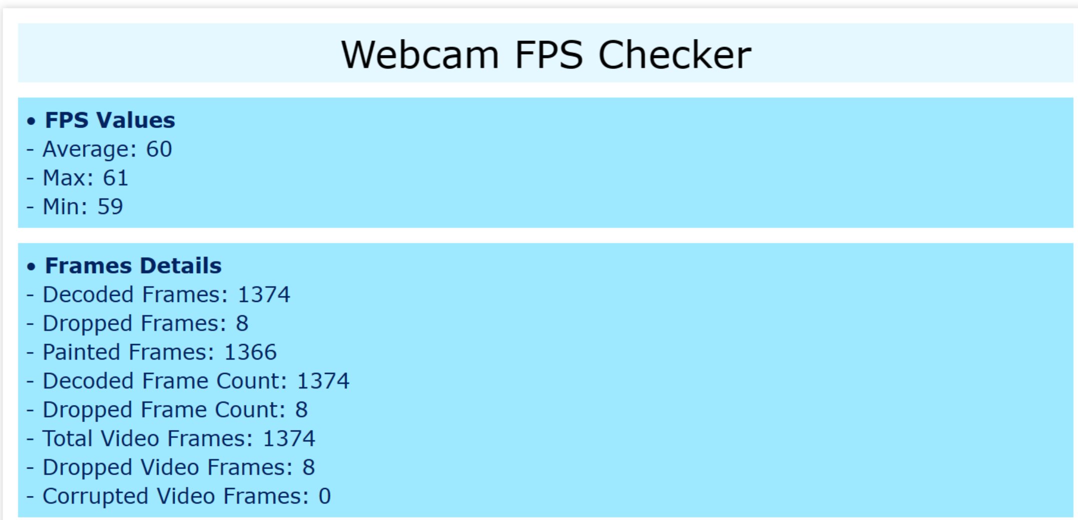

Figure 9: FPS Test Results

3.1.2 Multispectral Sensitivity

Our second high-level requirement was for the image sensor to be able to display a sensitivity to all three of the UV, visible, and NIR spectra to achieve multispectral capability. While we were unable to successfully program our image sensor to output video, we were able to verify the infrared sensitivity of the camera on our development board in Professor Gruev's Biosensors Lab. Our first step was to physically remove the lens that came with the development board, as it contained an 'IR cut' filter, which was blocking light input from the infrared spectrum. Manufacturers often include such a filter to protect the sensor, but in our case, it was blocking us from achieving the results we were looking for. Once we had removed the lens and replaced it with one provided by Professor Gruev (without an IR cut), we replicated the diagram shown in Figure 1 to create our testing setup, consisting of a high-powered 30 mW laser pointed towards a fluorophore sample, which, when excited by the laser, will emit photons in the infrared spectrum. We then pointed our development board camera towards the sample and introduced a visible cut filter, which blocked light input below the infrared wavelength, allowing us to isolate the spectrum in which we were trying to see a result. As evident in Figure 10 below, we were able to achieve an observable degree of sensitivity to the infrared spectrum.

In terms of ultraviolet sensitivity, considering the quantum efficiency chart in Figure 7, given the fact that the QE percentage is the same in both the infrared and ultraviolet regions (10% at 780 nm and 380 nm, respectively), theoretically, the sensor should be able to display just as much sensitivity to the ultraviolet spectrum as it did in the infrared spectrum test. Due to the significantly higher dangers of ultraviolet radiation and time constraints in finding safe ways to conduct ultraviolet wavelength testing, we are currently unable to show real-world results for the UV spectrum.

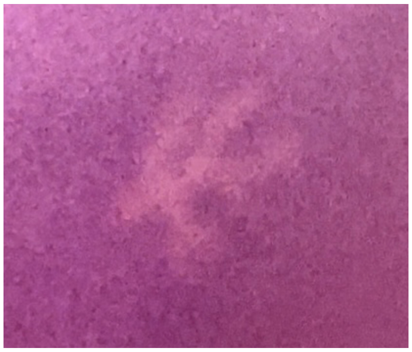

Figure 10: IR Detected by Dev Board

3.1.3 Field of View

As you can see in Table 2 in Appendix A, one of our original requirements that we wanted to verify for our image sensor subsystem was the ability to obtain a specific field of view. While the functionality of the sensor was not dependent on the exact visible angle, the requirement was more to ensure that we were able to obtain an image that was not distorted beyond legibility. We later realized that the field of view could be adjusted simply by using a different lens. Our final design accounted for this fact and included the ability to attach and detach lenses in accordance with the Canon C mount specification [8], with a flange back distance of 17.53 mm and a thread diameter of 1 inch. Professor Gruev's Biosensors lab had a wide array of lenses we were able to borrow from, and in this way, we were able to support many different fields of view. We chose not to verify our original requirement since we were able to find multiple lenses that had legible, usable, and distortion-free fields of view.

3.2 MCU

3.2.1 Device Enumeration

A necessary requirement for our PCB to work is for it to enumerate as a device on the PC. For our device, we planned to enumerate the device as a USB 3.0 device. After realizing that USB 3.0 is built on top of USB 2.0, we developed a bare minimum driver code that only enumerated the PCB as a USB 2.0 device and displayed a pattern of data (see Figure 11) that was independent of the image sensor. The output from the PCB was then verified with the development board. The purpose of this was to isolate the functionality of the microcontroller circuit and ensure that we were able to get a PCB to enumerate as a device.

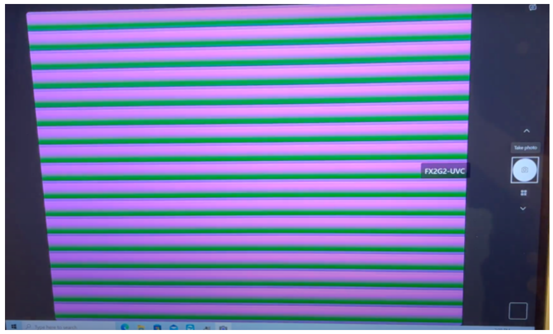

Figure 11: USB 2.0 Enumeration Pattern

3.2.2 Interfacing with Image Sensor

When it came to programming the PCB, we were unable to interface with the image sensor due to difficulties getting the MCU to enumerate as a USB 3.0 device. However, we were able to interface with the image sensor over the MIPI CSI-2 standard by using the development board. The development of this firmware allowed us to transmit image data back to the PC and display it successfully.

3.3 USB-C

3.3.1 Power Delivery

There was a total of 9 voltage regulators placed in our design, so it was imperative that every integrated component worked successfully. Furthermore, since we have a 4-layer PCB design, we had a dedicated 5-volt power plane to ensure ease of routing. To verify the functionality of each voltage regulator, a multimeter was used, where one end was on a ground pin, and the other end was on the output. To test the functionality of the power plane, we used a breakout board for the USB-C connection and tested the supply voltage to the board, as well as the output of an overvoltage protection device using a multimeter.

3.4 Enclosure

3.4.1 Dimensions

Our third high-level requirement constrained the total size of the device to 2.25” x 2.25” x 7” to be compact enough to hold and use in a variety of different applications. In its final form, the device is situated inside a 2-part enclosure whose dimensions are shown below. Figure 12 shows a height of around 5.5”, and Figure 13 shows the diameter of the enclosure lid to be slightly less than 2.25”; therefore, the total size of the device is less than the maximum specified dimensions.

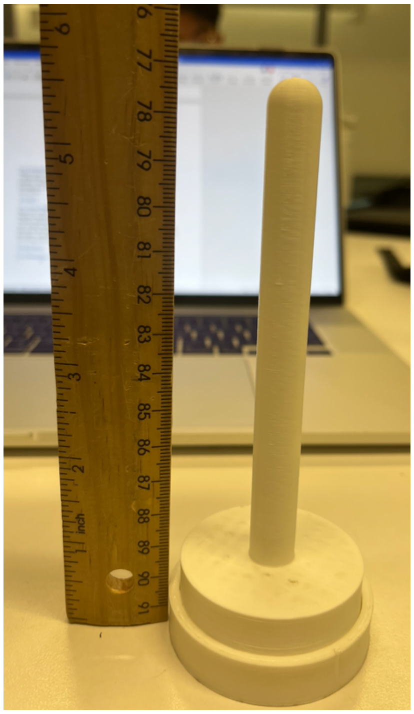

Figure 122: Height of Enclosure

Figure 13: Diameter of Enclosure Lid

4. Costs & Schedule

4.1 Parts

Table 1 details the breakdown of our ordered parts for PCB assembly, development board, and enclosure costs for one PCB.

Table 1: Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Total Cost ($)
CYUSB3065-BZXC-ND	Infineon Technologies	25.96	25.96	25.96
AR0330CM1C00SHAA0	On Semi	24.96	24.96	24.96
NCP361SNT1G	On Semi	0.75	0.75	0.75
ECS-2520SMV-192-FP	ECS Inc	2.23	2.23	2.23
SN74LVC2G07DBVR	Texas Instruments	0.45	0.45	0.45
HD3SS3220IRNHT	Texas Instruments	4.06	4.06	4.06
12401832E402A	Amphenol Inc	1.66	1.448	1.448
AP2127K-1.2TRG1	Diodes Incorporated	0.308	3.08	3.08
AP2127K-3.3TRG1	Diodes Incorporated	0.308	0.308	0.308
AP2127K-1.8TRG1	Diodes Incorporated	1.11	0.84	0.84
AP2127K-2.8TRG1	Diodes Incorporated	0.74	0.56	0.56
RC0201FR-075K1L	Yageo	0.81	0.81	0.81
CC0402ZRY5V7BB104	Yageo	0.46	0.46	0.46
RC0402JR-074K7L	Yageo	0.49	0.49	0.49
RC0402JR-07100KL	Yageo	0.49	0.49	0.49
CL05A105MQ5NNNC	Samsung Electro Mechanics	0.35	0.35	0.35
RC0402JR-0710KL	Yageo	0.49	0.49	0.49
RC0402JR-071K5L	Yageo	0.49	0.49	0.49
PCB Total				**$68.23**
CAD Enclosure	Siebel Center of Design			$0.60
Per Unit Cost				**$68.83**
Development Board	e-con Systems	299	299	299
Total				**$367.23**

4.2 Labor

The average salary for an electrical engineering graduate at Illinois is $87,769, and the average salary for a computer engineering graduate at Illinois is $109,176 [9]. Since we have a mix of both electrical and computer engineering majors, we will be using the average of the two, which is $98,472.5. For 50 work weeks at 40 hours per week, this comes out to $49.24/hour.
Once the design and assembly of the project begins, the project is estimated to take 10 hours of work per person per week, keeping in mind that a lot of work will be done in pairs or as a group. Designing began in week 5 of this class, and accounting for breaks and the due date of the project, we have roughly 9 weeks to complete the project resulting in 10 * 9 = 90 total hours per person. We will not be utilizing the machine shop for our project. We will also multiply our costs by 2.5 to account for any overhead for the development of this project.

$$49.24 \left[\frac{\$}{hr}\right] \times 2.5 \times 90 = 11{,}079 \left[\frac{\$}{person}\right] \times 3 = \$33{,}273 \text{ in labor costs for the project.}$$

4.3 Total

Summing up the total labor costs and costs of parts, the development of one PCB will cost:

$$\$367.23 + \$33{,}273 = \$33{,}640.23$$

4.4 Schedule

Please see Table 3 in Appendix B for our full schedule.
Our group remained on track until mid-way through the semester, at which point, we encountered a manufacturing lead time issue that altered the remainder of our schedule. Our original plan was to do a turnkey assembly with PCBWay, however, that would have taken too long, considering our deadlines. As a result, we had to hand-solder our PCB, which delayed our schedule by many weeks. We had not accounted for component delivery and potential hand-soldering issues in our original timeline. We had to solder our board multiple times, which contributed to our group being behind schedule and unable to achieve the full intended functionality for our project on the PCB itself.

5. Conclusion

5.1 Accomplishments

Looking back on the progression of this project, we are proud of the successes we were able to achieve. Given that we designed our PCB with PCBWay full turnkey assembly in mind, the fact that we were able to hand-solder the BGA and QFN components onto our board was a feat in and of itself. We are glad that our PCB was able to get recognized by the computer and enumerate, even if only as a USB 2.0 device, as that confirmed the fact that our MCU was soldered correctly and functioning as expected and that the real issue resided in our USB mux. Finally, we are proud that we were able to replicate the infrared test results of Professor Gruev's existing multispectral imaging setup on our development board camera – a far more portable, efficient, and inexpensive design, albeit to a lesser degree of sensitivity. This showed us that not only was multispectral sensitivity possible with a much less expensive sensor but also that our firmware developed for USB 3.0 was written correctly and worked as intended.

5.2 Uncertainties

The main reason why our PCB design did not work with full functionality was because our device did not enumerate as a USB 3.0 device but as a USB 2.0 device instead. This indicated that we were able to configure the device. We were able to diagnose this as a hardware issue because the traces that handle the USB 3.0 protocol are independent of the USB 2.0 traces. The traces that we used for USB 3.0 went through additional integrated circuits that introduced new sources of error in our design. Considering that this board was hand-soldered, if a simpler USB connection was chosen, or if we were able to get our PCB assembled by professionals, or if we developed code for the device to read image data through a USB 2.0 protocol, we are confident that our design would have worked as intended.

5.3 Ethical Considerations

5.3.1 Development

Faulty development can lead to complications in the operating room. (e.g., misdiagnosis or incorrect incisions) which will be a detriment to patient health and safety. To avoid harm, before a potential clinical usage, a well-documented and thorough review of the fidelity and technical quality of the project by various experts within the field will be performed. Areas where the team's developers are not competent will seek consultancy from appropriate experts.

5.3.2 Misuse

We envision accidental misuse as the result of not knowing the product's limitations when used in a clinical setting. This is an imperative factor to consider when we are concerned about patient safety. Aspects such as the device's longevity and where it can be used in a patient setting will need to be standardized to avoid any potential complications. Additionally, the attributes necessary to determine diagnosis must be tested with the device so patients are not misdiagnosed.

5.3.3 Safety and Regulatory Standards Industry Standards

Within the medical device industry, regulations will be determined by the intended use case of the technology. For instance, if the desire is to use it as a preliminary tool for a patient diagnosis of skin cancer, it could potentially qualify as a Class II device and follow the FDA's guidelines for further development in a clinical setting. However, if there were a demand to use such a product as a small surgical tool, the product would undergo a stringent regulatory review known as Premarket Approval as it likely qualifies as a class III device. Furthermore, since it is a medical device, we need to ensure a degree of sterility during application. According to FDA guidelines, usage of medical devices should be done in a manner that will "prevent contamination of equipment or product by substances that could reasonably be expected to have an adverse effect on product quality" [10]. Therefore, in clinical applications, it is imperative that the device is used while wrapped in a sterile plastic bag, which is a common practice for medical electronics.

5.4 Future Work

Due to the timeline of the course and budget constraints, the number of iterations on the fabrication of our hardware design was limited to one order. Given an opportunity to continue this work, design changes could be made. To start, establishing a less complex USB connection, such as USB type A or USB Micro-B, could have simplified our design and allowed us to establish fully functioning hardware. Doing this would give us a baseline to develop firmware from and allow for more detailed image processing.

Keeping with this theme, additional iterations would give us the opportunity to further decrease the size of the design. Redesigning the board so that components are placed on the back side would save space and allow us to design a smaller device. Furthermore, when designing a new enclosure, using different materials for our 3D prints would give us the opportunity to create a final product that is more ergonomic and capable of being used in a surgical setting.

Another thing to keep in mind is the choice of USB 3.0. Research into whether a higher data transfer speed is required for this purpose should be conducted to minimize additional and unnecessary complexity.

5.5 Acknowledgements

We would like to thank to Professor Arne Fiflet and the senior design course staff for supporting the development of this project throughout this semester. Also, special thanks to Professor Viktor Gruev, Jason Zhang, Jason Paximadas, Zhongmin Zhu, and Brianna Hayek for their technical support, resources, and insight for the duration of our project.

References

[1] “Multispectral Imaging: Artificial intelligence in medical imaging,” SpectralAI, https://www.spectralmd.com/wound-healing-technologies/multispectral-imaging/#:~:text=Multispectral%20Imaging%20in%20Diagnostic%20Medicine,visible%20to%20the%20human%20eye. (accessed Sep. 28, 2023).

[2] “Multispectral Imaging in Healthcare – a convolution of machine vision and Spectroscopy,” Jai, https://news.jai.com/blog/multispectral-imaging-in-healthcare-a-convolution-of-machine-vision-and-spectroscopy (accessed Sep. 28, 2023).

[3] Hewlett-Packard Company, Intel Corporation, Microsoft Corporation, Renesas Corporation, STEricsson, and Texas Instruments., *Universal Serial Bus 3.1 Specification*. (accessed: Sep. 28, 2023).

[4] IPC, "IPC-2141A: Design Guide for High-Speed Controlled Impedance Circuit Boards," IPC, https://www.ipc.org/TOC/IPC-2141A.pdf. (accessed: Sep. 28, 2023).

[5] CYUSB306X, EZ-USBTM CX3 MIPI CSI-2 to superspeed USB bridge controller, https://www.infineon.com/dgdl/Infineon-CYUSB306X_EZ-USB_TM_CX3_MIPI_CSI-2_to_SuperSpeed_USB_bridge_controller-DataSheet-v17_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ecbbb354559 (accessed Sep. 29, 2023).

[6] AR0330: 3.4 Mp, 1/3-inch format, 2.2 µm pixel CMOS digital image sensor,” Mouser, https://www.mouser.com/pdfdocs/ON_Semi_AR0330_Datasheet.pdf. (accessed: Nov. 18, 2023).

[7] Alvium G5-812 UV | 8.1 MP Sony IMX487 CMOS sensor,” Allied Vision, https://www.alliedvision.com/en/products/alvium-configurator/alvium-g5/812-uv/. (accessed: Sep. 16, 2023).

[8] “The C-mount lens specification...0. package window,” Meiji Techno, https://meijitechno.com/camera-mounts/. (accessed: Nov. 16, 2023).

[9] Grainger Engineering Office of Marketing and Communications, “Salary averages,” Electrical & Computer Engineering | UIUC, https://ece.illinois.edu/admissions/why-ece/salary-averages (accessed Sep. 28, 2023).

[10] “The code affirms an obligation of computing professionals to use their skills for the benefit of society.,” Code of Ethics, https://www.acm.org/code-of-ethics (accessed Sep. 28, 2023).

Appendix A: Requirement and Verification Table

Table 2 below, describes all our requirements and verifications for our project.

Table 2: Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The device must be situated in an enclosure that is at most 2.25 x 2.25 x 7 inches in dimension.	Measure all sides of the enclosure using a tape measure.	Y

Image sensor must support and sustain real-time video at a rate above 20 frames per second for at least 5 seconds.	1. Program the board to read in real time video.
	2. There are two options to calculate framerate.
	    1. Using a performance analysis tool such as FRAPS on Windows or Quartz Debug on MacOS.
	    2. While grabbing frames, instantiate a variable in your program to increment each time a frame is grabbed. Program a timer that will restart every time the number of frames grabbed reaches 20. At this point, divide the number of frames (20 in this case) by the time elapsed to calculate the frame rate.
	3. Using either method, display the frame rate as a numerical value to see if it is within a range.
	4. Start a timer for 5 seconds
		Y

	and see if the sustained frame rate is for 5s. 5. Results will be presented in a table.	
The device must be able to detect no more than 3 cm in any direction from 2 cm away; as a result, the "field of view angle" for the image sensor must be 74 degrees.	1. Measure the distance from the camera to a flat surface, like a wall, using a tape measure, making sure the measure is perpendicular to that surface. 2. Looking at the monitor with the camera output, measure horizontally across the entire field of view up to the edge of visibility on the monitor to obtain the width. 3. The tangent of half the width divided by the distance to the surface will provide the field of view angle of the image sensor.	N (see section 3.1.3)
Image Sensors Power Delivery: The power subsystem must be able to supply at (2.8 $\pm$ 0.1) V for Analog, PLL, Pixel, and MIPI supply voltages. It must also provide (1.8 $\pm$ 0.1) V for core voltage and I/O digital voltage. 1. V_{AA}, $V_{AA\text{-}PIX}$, $V_{DD\text{-}PLL}$, $V_{DD\text{-}MIPI}$ = 2.8 $\pm$ 0.1V 2. V_{DD}, V_{DD_IO} = 1.8 $\pm$ 0.1V The power subsystem must be able to supply the following currents for the pins: • Digital Operating Current: 114 - 136 mA • I/O Digital Operating Current: 0 mA • Analog Operating Current: 41 - 53 mA • Pixel Supply Current: 9.9 - 12	Measure the output voltage of the regulator of interest (1.8V, 2.8V) using an oscilloscope or multimeter across the linear regulators, check to see if the potential difference is within tolerance.	Y

mA • PLL Supply Current: 15 - 27 mA • MIPI Digital Operating Current: 35 – 49 mA.		
Microcontroller Power Delivery: The power subsystem must be able to supply at (5 ± 1) V for USB operation, (1.2 ± 0.05) V for USB 3.0 supply, digital power, and MIPI operations, and 1.8 to 3.3 V for I/O and Clock operations. 1. USB Over Voltage Protector a. 4 V < VUSB < 6 V with at most 60 mA. 2. 1.15V < V_{DD_MIPI}, V_{DD}, $U3TX_{VDD}$, $U3RX_{VDD}$, A_{DD} < 1.25 V with at most 190 mA. 1.7 V < V_{DDIO} and C_{VDDQ} < 3.6 V with at least 4.58 uA.	1. Using a signal generator or voltage source, apply 15V to the IN pin on the OVP. Using an Oscilloscope, read the potential difference across pin 2 and pin 5, which corresponds to GND and OUT, respectively. There should be no difference since 10V exceeds overvoltage protection. 2. The pin V_{USB} must be 5 +/- 1V to function, connect an oscilloscope across the linear regulator feeding into the pin to measure the correct potential difference. 3. Measure the output voltage of the regulator of interest (1.8V, 2.8V) using an oscilloscope and multimeter across the linear regulators, check to see if the potential difference is within tolerance.	Y
USB-C 2:1 Mux: 1. Mux requires @ 0.7 mA. a. V_{CC33}= 3.3 ± 0.3V V_{DD5} = 5V $\pm$ 0.5 V	1. Measure the output voltage of the 3.3V and 5V using an oscilloscope and multimeter across the linear regulators of interest, check to see if the potential difference is within tolerance.	Y

Appendix B: Intended Schedule

Table 3: Preliminary, Intended Schedule

Month	Week	Tasks/Goals	Person Assigned
September			
	Week of 9/18		
		Start Design Document	All
		Order and Finalize Components: Get Clarification on how PCB assembly works and payment process	All
		Expenses Chart	Amartya Bhattacharya
		Visual Aids	Isha Akella
		Physical Diagrams	Jason Jung
		Start PCB Design Once Finalized Components. PCB Design: KiCAD	Jason Jung
	Week of 9/25		
		Get TA to look over design document	All
		Order/Finalize components: FINAL Notice	All
		Find Code that Works with Dev Board	All
		PCB Design: MCU	Isha Akella
		PCB Design: Image Sensor	Amartya Bhattacharya
		PCB Design: USB Connection	Jason Jung
		Design Document is Due	
		Get Professor to Review PCB Designs	All
		Cross verify each other's PCB designs	All
October		Prepare for Design Review (Power Point)	All
	Week of 10/2	**Design Reviews**	All
		Cross verify each other's PCB designs	All

		Dev Board Programming (If arrived)	Amartya Bhattacharya
		Start looking into 3D Enclosures to print out	Isha Akella
	Week of 10/9	**Order PCB way (First round of PCB Design Finished)**	All
		PCB Design Feedback and Assemble PCB Board (ASAP)	All
		3D Printing First Round	Jason Jung
		Teamwork Evaluation I	All
	Week of 10/13 to November		
		Individual Progress Reports Due	All
		Assemble PCB Board (Final Round of Design)	All
		Test PCB Design and functionality,	All
		Integrate PCB into Enclosure, 3D print Enclosure (2nd Round)	All
November		Prepare for Mock Demo and Presentation	All
	Week of 11/13	**Mock Demo**	
		Get Feedback from Demo and Implement Changes	All
	Week of 11/27	**Final Demo**	
December			
	Week of 12/4	**Final Presentation**	All
	Week of 12/7 (Reading Day)	**Lab Notebook**	All

Appendix C: Schematics

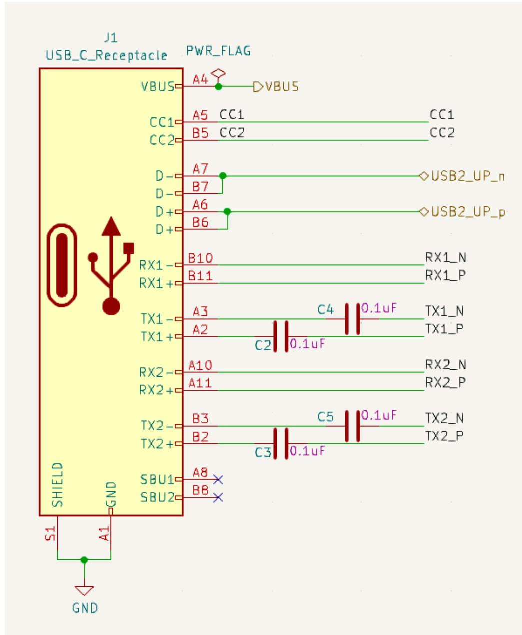

Figure 14: USB-C Receptacle Schematic

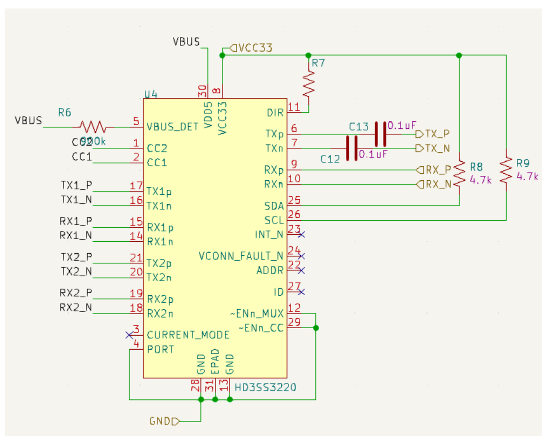

Figure 15: USB Mux Schematic

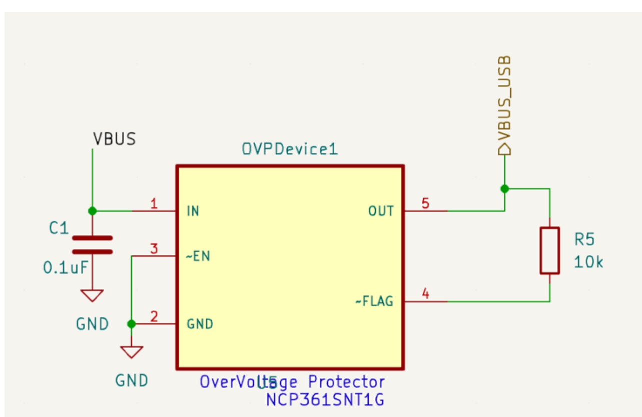

Figure 16: Over Voltage Protection Schematic

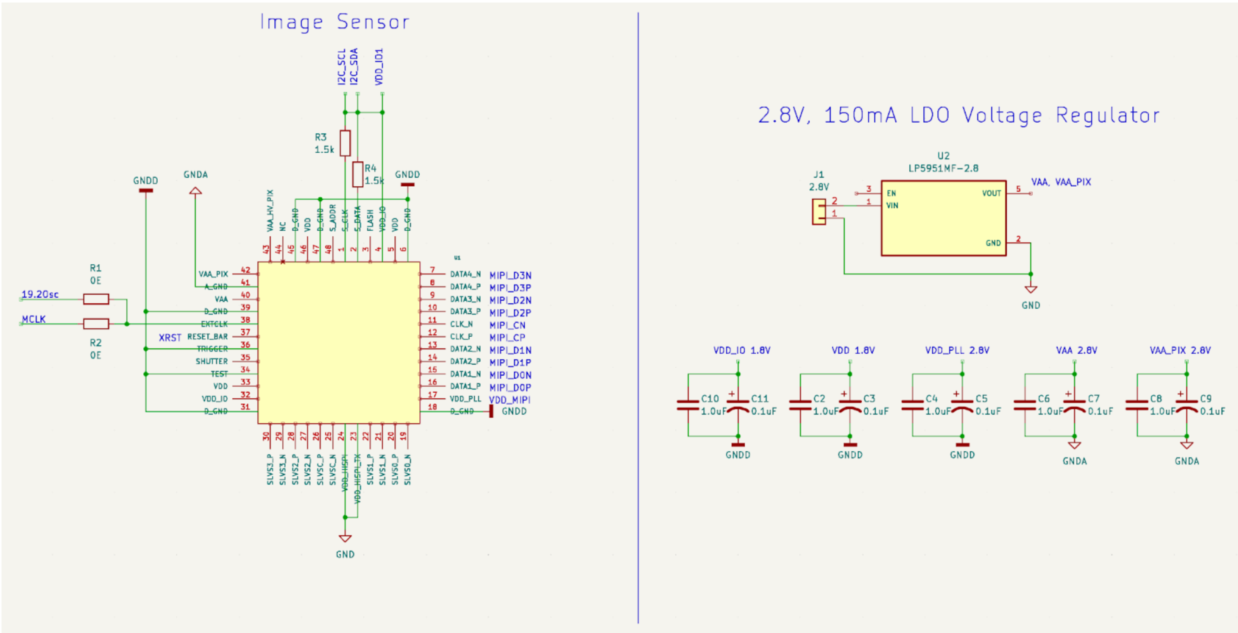

Figure 17: Image Sensor and Voltage Regulator Schematic

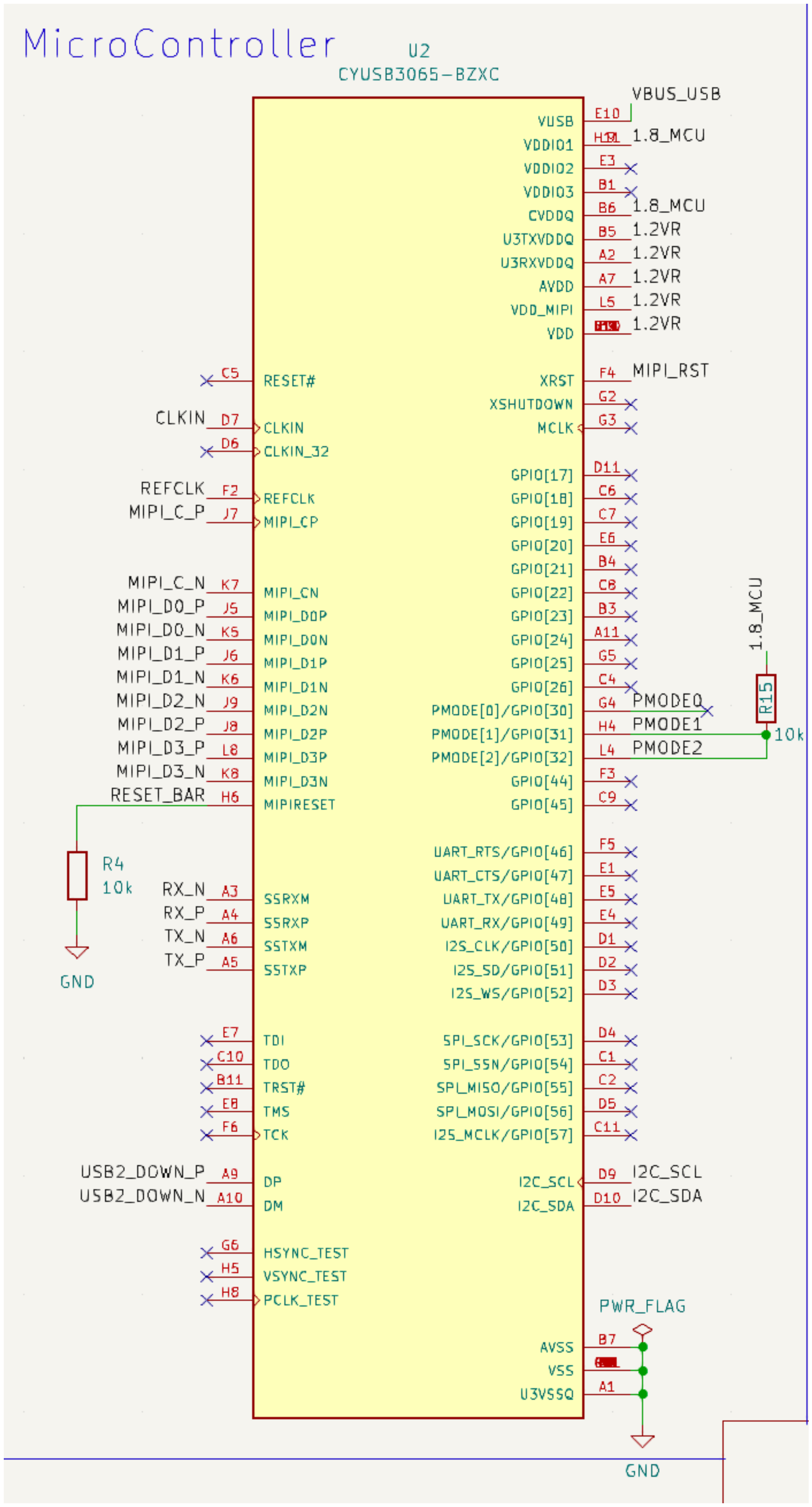

Figure 18: Microcontroller Schematic