STRE&M: LED Absorption Spectrometry

By

Adrian Jimenez

Gage Gulley

Yichi Zhang

Final Report for ECE 445, Senior Design, Spring 2023

TA: Abhisheka Mathur Sekar

3 May 2023

Project No. 52

Abstract

This report aims to design a low-cost and sturdy LED spectrometer that can be integrated into a concise urine collection and analysis system (STRE&M device). To make the integration and make the system run autonomously, a few requirements need to be met. First, the spectrometer needs to be concise and able to let the sample pass through it. Second, it needs to have the ability to generate the absorbance spectrum ranging from 440nm to 660nm with fair accuracy. Third, the spectrometer needs to have WiFi functionality for data transmission. The first requirement led us to use LEDs as our light source. The third requirement led us to choose ESP32-S3-WROOM-1 as our microcontroller since we need to have WiFi functionality for data transfer. It turns out that the LED spectrometer can generate a correct absorbance spectrum and can be integrated into the STRE&M device.

Contents

1. Introduction

1.1 Problem

Urine tests are critical tools used in medicine to detect and manage chronic diseases. These tests are often over 24 hours and require a patient to collect their sample and return it to a lab. With this inconvenience in current procedures, many patients do not get tested often, which makes it difficult for care providers to catch illnesses quickly [7]. The tedious process of going to a lab for urinalysis creates a demand for an "all-in-one" automated system capable of performing this urinalysis, and this is where the STRE&M device comes in.

The current prototype is capable of collecting a sample and pushing it to a viewing window. However, once it gets to the viewing window there is currently no automated way to analyze the sample without manually looking through a microscope, which greatly reduces throughput. Our challenge is to find a way to automate the data collection from a sample and provide an interface for a medical professional to view the results.

1.2 Solution

Our solution is to build an absorption spectrometer that is capable of measuring and plotting the absorbance of casts, bacteria, and cells that may be present in the sample. Since each protein that we are trying to detect absorbs light at a particular wavelength, we need to emit this wavelength of light. Our approach is a low-cost, effective spectrometer that can emit these wavelengths of light corresponding to the proteins we desire to detect and measure the absorption at the wavelengths.

1.3 Visual Aid

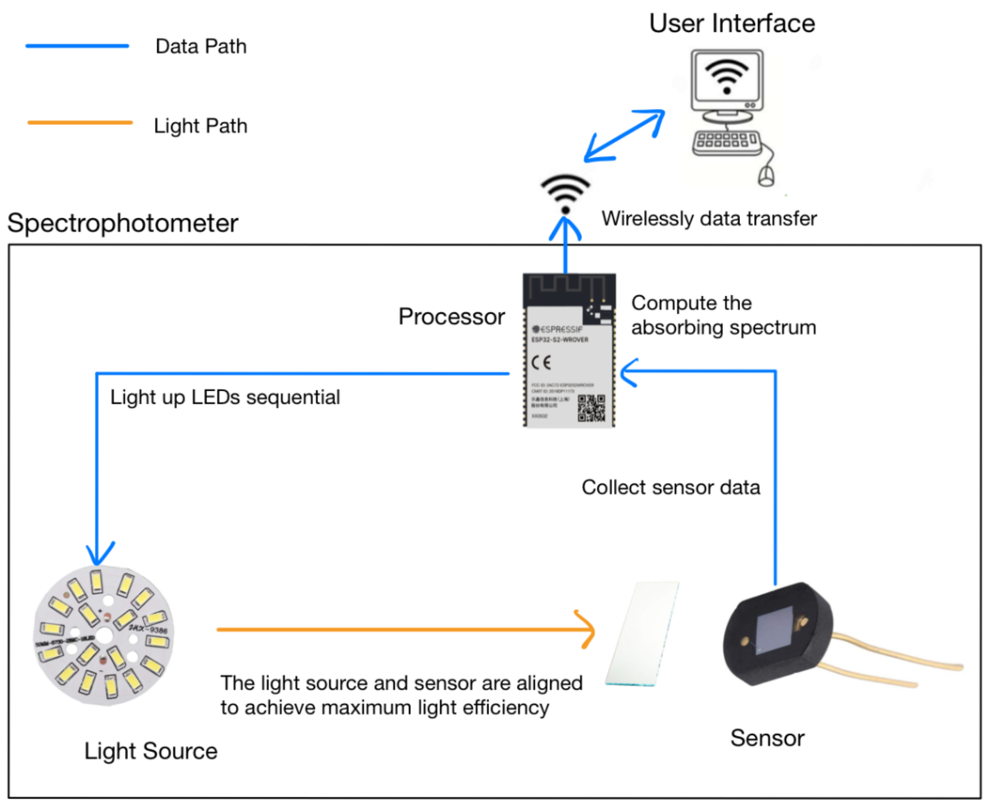


Figure 1: Visual Aid

1.4 High-Level Requirements

To consider our project successful, our LED Spectrometer must fulfill the following requirements:

- The spectrum analysis and data transfer must be completed in less than 30 seconds
- Our system must be able to produce an absorbance spectrum for a sample with known absorbance in our desired range (440-660nm) and generates no response for a sample with known absorbance in the range excluding (440-660).
- The device must be capable of performing absorbance spectroscopy with a resolution rate of ~36 nm for the light source emitting lights ranging from 440 nm to 660 nm.

1.5 System Overview

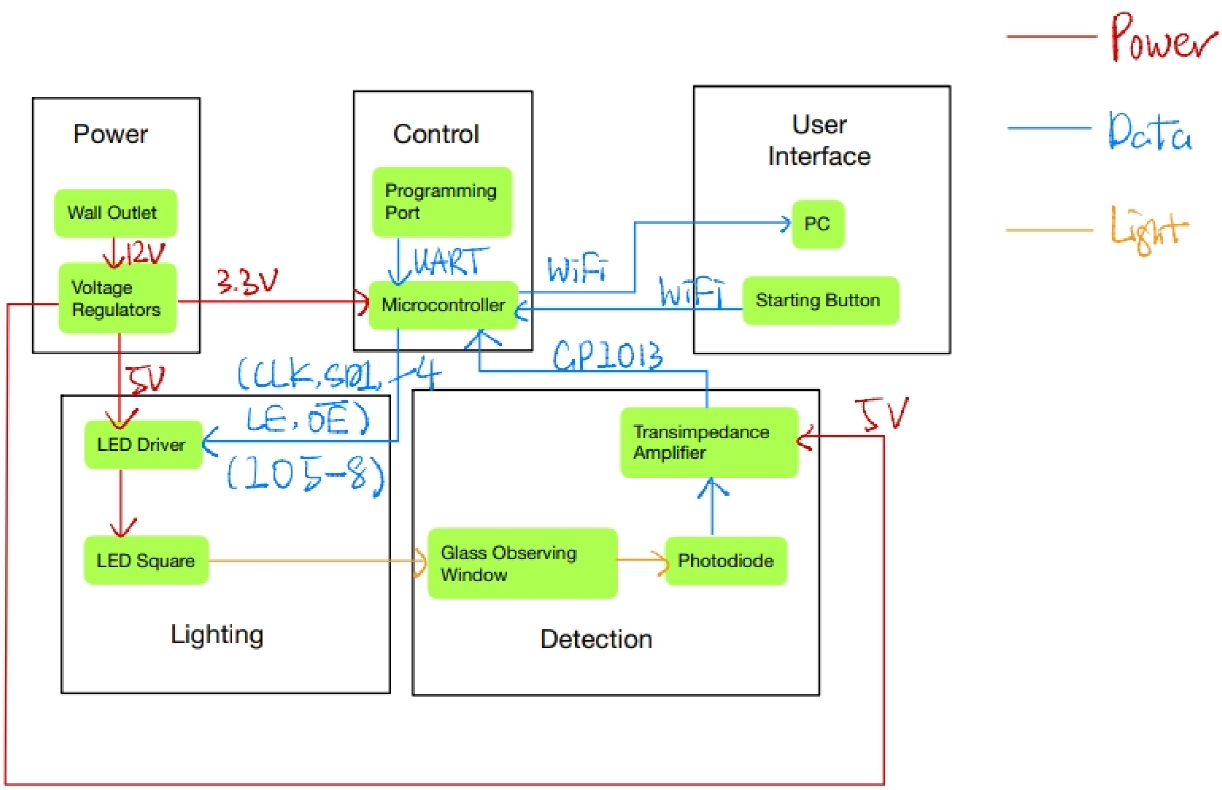


Figure 2: Block Diagram of System

The spectrometer consists of four unique subsystems that are interconnected as well as a webpage that acts as a collection point for spectrum analysis. The power system uses voltage regulators to produce stable 5V and 3.3V outputs to the other subsystems. The lighting subsystem uses an LED driver to sequentially light the LEDs to illuminate the sample. The detection subsystem uses a photodiode to measure the current an LED generates after passing through the sample and converts it to voltage to send back to the microcontroller. The Control system uses an ESP32-S3 microcontroller to interface with the other subsystems based on control signals from the user interface. It will also calculate and transfer the absorbance spectrum of a sample to the webpage. The user interface is an HTML webpage that can display the calculated absorbance spectrum in a graphical format. Through the webpage, you can also launch the calibration and sample collection of the device.

2. Design

2.1 Design Procedures

2.1.1 Subsystem 1 (Lighting)

The lighting subsystem serves to produce light at wavelengths that are of interest in the field of urinalysis. The lighting subsystem is connected to the ESP32 microcontroller as well as to an LED array. The lighting subsystem required great thought when designing schematics and selecting the components to use. One of the biggest considerations for the lighting design was the necessity of an extremely stable and consistent current source to drive the LEDs. Upon searching for components, we selected the STP08DP05MTR, as it allows for 8 outputs to be controlled independently, and features an adjustable current value. Upon selecting this we chose LEDs with wavelengths corresponding to samples that we could generate with known absorption spectrums. This is desirable in the case of proving the concept since we do not need to use biological samples or any other potentially dangerous samples. Additionally, by choosing LEDs in the range of 440-660 nm, we get a larger spectrum, though it is more spread out and loses resolution [2]. An alternative approach would be to use LEDs with wavelengths much closer in value but evenly spread out, then you would increase the resolution and the graph would be a better representation of the full spectrum.

2.1.2 Subsystem 2 (Detection)

To detect the absorbance, we use a photodiode to produce a current that corresponds to a given light intensity captured after passing through a sample [3]. We then collect the light after passing through the sample. The design has light from the LED focused on the sample, which then absorbs and emits the remaining light that the photodetector will be nearby to detect. Our first task was to find a photodiode capable of detecting in the range that we needed to be detecting in and make sure it was efficient in this range as well. After selecting the photodiode, we determined that we would need some amplifier circuit to convert the current produced by the photodiode to a voltage that could be read by the ESP32.

The amplifier circuitry was initially a complex circuit consisting of two op-amps and many peripheral resistors and capacitors. However, we determined that this design was simply too complex for our needs and that we valued the ease of adjusting gain quickly. As a result, we decided to use a trans-impedance amplifier so that we only needed to use a single through-hole resistor to adjust the gain depending on the environment.

2.1.3 Subsystem 3 (Control Unit)

The control subsystem is programmable. The control subsystem is connected to the lighting subsystem, the detection subsystem, and the power subsystem. It gathers and transmits data with the peripheral subsystems (the lighting system and detection subsystem) to obtain the measurement data and then calculate the final spectrum. After it gets the final spectrum, it wirelessly sends the result to a server using its built-in wifi capability. Included on the webpage is a button to initiate calibration as well as a sample analysis. Figure 3 gives a better understanding of how the control unit works as a whole in the design, as the microcontroller handles all tasks outlined.

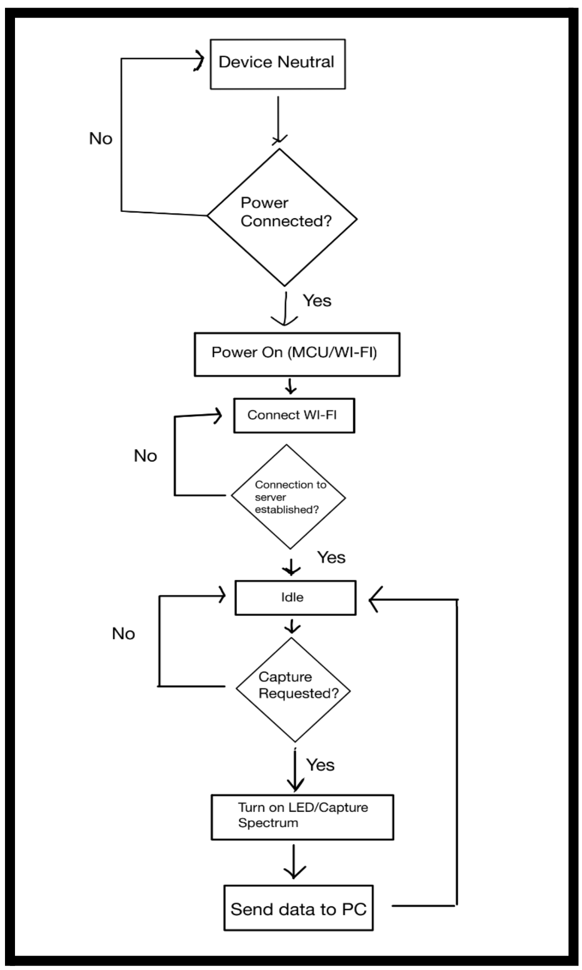


Figure 3: Software flow chart

2.1.4 Subsystem 4 (Power)

The power subsystem required the least amount of components and design considerations. We wanted a design that was simple and not overcomplicated. For this design, we are not overly concerned with extensive power management, as our system as a whole is fairly low power. We decided to choose components that were well-documented and have been used commonly. The subsystem consists of two linear voltage regulators to turn our 12V input from a wall power supply into 5V and 3.3V to be used for components in our design. There are only 2 linear regulators and a barrel jack, accompanied by some peripherals to aid.

2.2 Design Details

2.2.1 Subsystem 1 (Lighting)

The lighting subsystem consists of an LED driver and an LED array. The LED driver is the STP08DP05MTR, and this LED driver is an 8-output linear shift register. The LED driver is capable of supporting up to 100mA per channel, but for our application we only need 20mA. The datasheet of the LED driver features a chart correlating an external resistor value to the current value. Upon evaluating the chart, we decided to use a 1kOhm resistor to achieve the desired 20mA. To control the LED driver we utilize the serial data in (SDI) pin to select the led that we want to control. All LED states are initialized as off and latched in the "0" state for each output. When we want to sweep the LEDs sequentially, the LED we desire to turn on at that time is updated in the shift register, and we do this one at a time until all of the LEDs are lit.

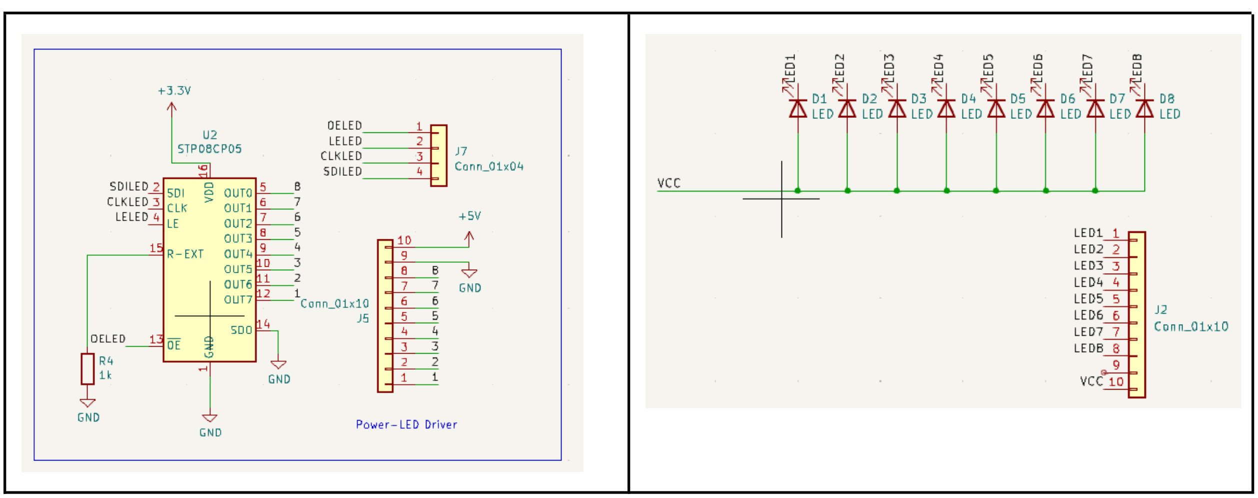


Figure 4: Schematic for LED driver and LED array 10-pin header

The LED driver also has an SDO pin, but since we do not need to feed this back into the microcontroller, we decided to ground this pin to avoid any potential issues. Besides the external resistor selected to control the current value, the lighting subsystem is a very simple physical design. There are only 4 connections to the microcontroller, and these were done via a 4 pin header. We decided to connect these with a ribbon cable to the microcontroller, which also has multiple headers surrounding it. We did this to allow for quick and easy debugging if we were facing issues with the LED driver, though luckily this issue never arose. These headers could now be eliminated and replaced with PCB traces now that the design is confirmed to work.

2.1.2 Subsystem 2 (Detection)

The detection subsystem consists of a trans-impedance amplifier, a photodiode, and a biasing resistor. The light from the LED passes through the sample in the viewing window. The light that is not absorbed by the sample passes through to the photodiode. The photodiode then outputs a current that is then sent to the trans-impedance amplifier. The trans-impedance amplifier we used is the OPA380AIDGKR made by Texas Instruments. The trans-impedance amplifier converts the input current to an output voltage. The gain of the current to voltage converter can be set using a single resistor. We made the biasing resistor a through-hole component to allow for easier ability to change the gain value. We utilized a 100k Ohm resistor to give us a large gain value while staying under the 3.3V threshold for the ESP32 pins. The trans-impedance amplifier is also able to maintain a constant biasing voltage using only a positive 5V input. This eliminated the need for a -5V bias that we could not generate for our previous detection circuit. The output voltage is then sent back to the microcontroller.

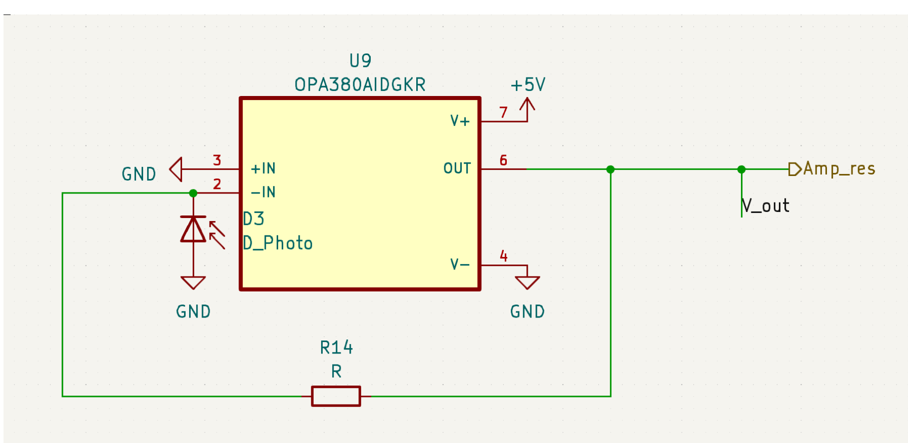


Figure 5: Schematic for the detection subsystem

2.1.3 Subsystem 3 (Control Unit)

The ESP32-S3-WROOM-1 was chosen as our microcontroller due to its Wi-Fi and Bluetooth capabilities, as well as its internal ADC. The microcontroller was placed on the perimeter of our PCB to allow for the antenna to have the most headroom possible for Wi-Fi communication.

Additionally, for the control unit, we decided to include reset and boot buttons on our initial PCB design, though we later discovered that we only needed to have the reset. The reset button is in place so that we could reset the PCB quickly and without having to use a programmer.

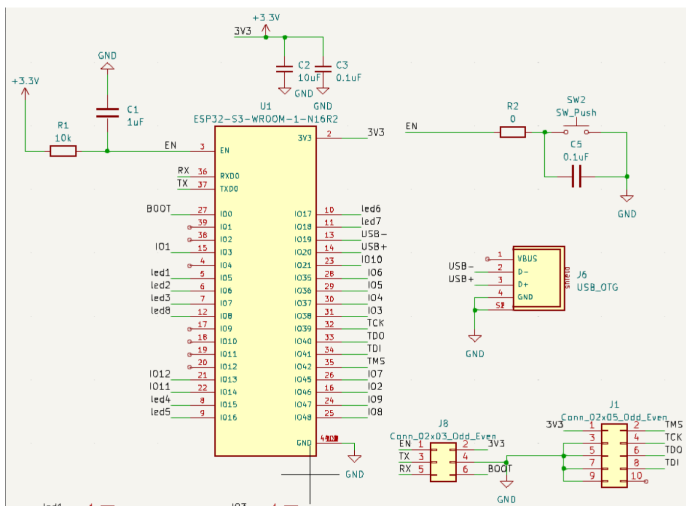


Figure 6: Control Subsystem Schematic

Another vital component of our control unit is the UART and JTAG programming headers. For our project, we only used the UART pins on our layout to program and send/receive data from the microcontroller.

Overall the system is condensed and simplified to avoid using any unnecessary components. However, an alternative approach would be to delete the UART and JTAG headers and instead program the microcontroller through the USB port.

2.1.4 Subsystem 4 (Power)

The power subsystem was designed to avoid adding too many unnecessary peripheral components or unnecessarily selecting components that were excessive in ability. The power system begins with the AC wall outlet being converted to 12VDC via a standard wall wart with a 5.5mm barrel connector. Additionally, our barrel jack connection features a diode to prevent a situation in which the polarity is reversed from affecting the device. This barrel connector supplies 12VDC to the PCB to be distributed amongst the device, however, there is not a single component on the device designed to work with a 12V supply voltage.

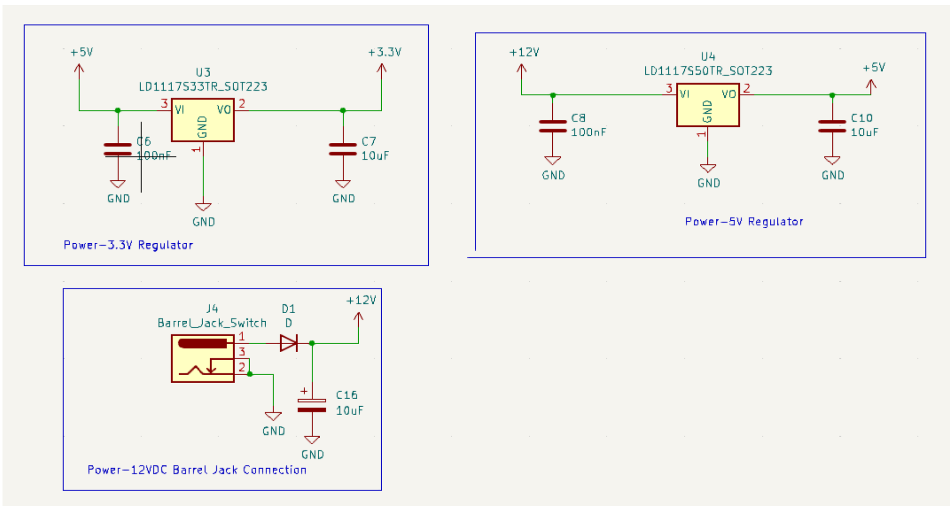


Figure 7: Barrel Jack and LDO schematics

Upon inspecting the datasheets of our components, we concluded that a 3.3V and 5V voltage regulator would be necessary on our PCB. To complete this task, we utilize two commonly found linear voltage regulators, for the 3.3V regulator we use the LD1117S33TR and the LD1117S50TR for the 5V regulator. We first step the voltage down to 5V with the 12V as the input for the 5V voltage regulator. Then the 5V regulated supply serves as the input voltage for the 3.3V regulator. This was done in this fashion to prevent excessive heat from becoming an issue in the future.

3. Verification

3.1 Subsystem 1 (Lighting)

Our original requirement for the LED Driver was to have it sequentially light LEDs starting with the lowest wavelength (1) and ending with the highest wavelength (8). Due to the change in the design from 8 LEDs to 6, we also updated this requirement to be sweeping from one to six, in order of increasing wavelength. To verify this functionality, we initially wrote code in Arduino IDE that functioned to light each LED sequentially. This was achieved by utilizing a for-loop to change each LED state from “off” to “on” and update them in the register. Our LED driver functioned correctly and was able to sequentially light up each LED in a non-overlapping manner in order of increasing wavelength.

An additional requirement for the lighting system states that the LED driver must be able to supply the LEDs with a constant current supply of 20mA. To verify this we followed a very simple procedure utilizing Ohm's law to determine how much current the LED driver was outputting. Figure 9 displays the simple circuit consisting of one output of the LED driver being connected to a breadboard containing nothing more than a LED and a 10 ohm resistor. This simple circuit when run under a conventional sample analysis yielded a voltage value of 0.194V (shown in Figure 8) across the 10 ohm resistor. By Ohm's Law:

$$V = I * R \text{ and } I = V/R$$

$$V = 0.194\,,\ R = 10$$

$$I = 0.0194\ [A] = 19.4mA$$

Figure 8: LED driver circuit result

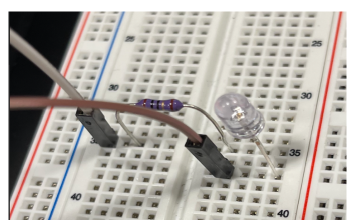


Figure 9: Simple LED driver test circuit

As proved above, the LED driver is providing a consistent and steady current supply that is allowing for consistent results on an overall system level. From this data, we conclude that the lighting subsystem meets both of the specific requirements that we set forth at the beginning of the design process.

3.2 Subsystem 2 (Detection)

Since the MCU cannot accept input larger than 3.3V, the output of the detection subsystem needs to be controlled strictly under 3.3V. To verify that, we recorded the maximum voltage output from the detection subsystem under a sample collection process. The resulting output ranged from 0.27V to 3.2V (Figure 10 shows the max reading of the detection subsystem). From these results, we were able to determine that the microcontroller will not be damaged.

Figure 10: Max Voltage Output of the Detection Subsystem

Another requirement is the consistency of the detection subsystem. As it should be able to produce the same results as long as the same sample is provided to it. To test this response, we calibrated the system with plain water to get our calibration data. Then, we analyzed multiple samples all with different known spectrums. Finally, after running these tests, we reinserted the plain water sample back into the device and observed the behavior. As shown in Table 1, the measured absorption for the plain water after multiple samples was very close to the expected flatline that we would expect for retesting a sample used as the calibration sample. With differences never exceeding more than 5% of the calibrated data, we were able to determine that our detection system was providing reliable and consistent results.

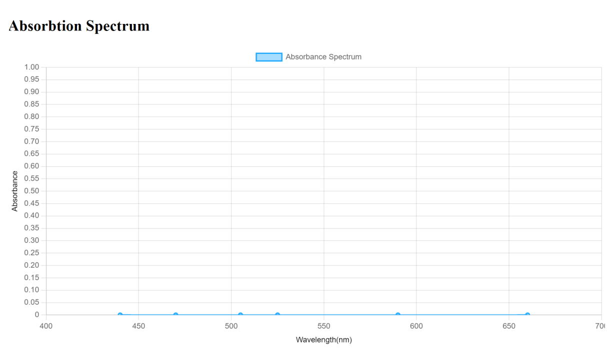
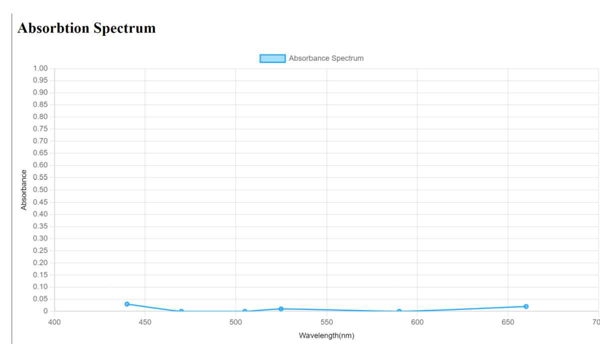
Expected Spectrum for Water	Measured Spectrum for Water (after samples)
	

Table 1: Absorbance spectrum for plain water

3.3 Subsystem 3 (Control Unit)

The requirement for the control unit is it needs to be able to wirelessly transfer data from the MCU to the user interface within 2 seconds. To test the data transfer speed, we record the time before the MCU sends data out and the time after the MCU sends data out (refer to Figure 11 for the timing code). Taking the difference between the two times of the given function gives us the data transfer speed [6]. It turns out that the data transfer speed is very fast. It only takes 0.000021 seconds for the MCU to send the data out (refer to Figure 12 for the serial monitor output).

```
clock_t t;
t= clock();

updateGraph();

t= clock()-t;
double time_taken = ((double)t)/CLOCKS_PER_SEC; // in seconds
Serial.println();
Serial.printf("Data Transfer took %f seconds to execute \n", time_taken);
```

Figure 11: Timing Code

```
Finished with data collection, update Graph
Calibrated Values array:
3851 1502 3917 3546 2619 480 42 42
Sample LEDvals array:
3818 1501 3914 3544 2612 480 42 42 Difference array
0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00

Data Transfer took 0.000021 seconds to execute
```

Figure 12: Serial Monitor Output

There is an additional requirement regarding the accuracy of the spectrometer. The spectrometer should be able to find the wavelength of peak absorbance within ±15nm. To verify that requirement, we used blue dye as the testing sample[1]. Table 2 shows the normal absorbance spectrum and the absorbance spectrum produced by the LED spectrometer of blue dye.

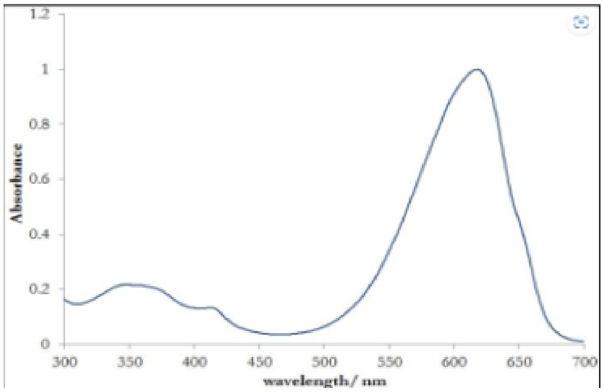
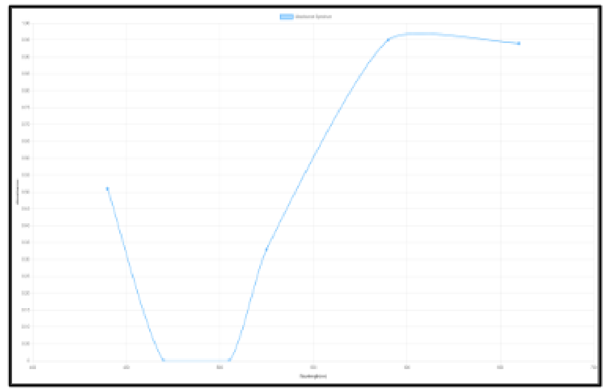
Normal Absorbance Spectrum	Absorbance Spectrum of LED spectrometer
	

Table 2: Absorbance Spectrum for Blue Dye

In the normal absorbance spectrum, the peak is located at around 600nm, and, in the absorbance spectrum produced by the LED spectrometer, the peak is located at 590nm. The wavelength difference is 10nm, which is less than 15nm.

3.4 Subsystem 4 (Power)

For the power subsystem, the requirement is about how accurate the outputs of the LDOs are. The voltages provided by the LDOs cannot vary more than ±0.2V. To verify this requirement, we measured the voltages across the 3.3V LDO and 5V LDO when the LED spectrometer is analyzing a sample, which means the spectrometer is run at full load and will drain maximum current out of the LDO. The measurement voltages are 3.30105V and 4.98708V (refer to Table 3 for the measurement results) respectively. As shown in Table 3 below, the LDO regulators are able to function as extremely stable voltage sources.

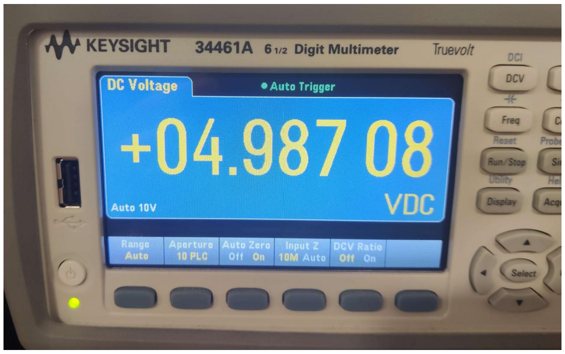
3.3V LDO output	5V LDO output
	

Table 3: LDO Outputs

4. Costs

4.1 Parts

Table 4 Parts Costs

Part	Manufacturer	Description	IndividualCost ($)	QTY	Actual Cost ($)
ESP32-S3-WROOM-1-N16	Espressif Systems	Microcontroller	**3.83**	**2**	**7.66**
L6R06H-120	Tri-Mag Inc	Wall power supply	**6.05**	**1**	**6.05**
PJ-036AH-SMT-TR	CUI Devices	Barrel Connector	**1.45**	**1**	**1.45**
LD1117S33TR	STMicro	3.3V Regulator	**0.66**	**1**	**0.66**
LD1117S50TR	STMicro	5V Regulator	**0.66**	**1**	**0.66**
VTB8441BH	Excelitas Technology	Photodiode	**3.41**	**1**	**3.41**
OPA380AIDGKR	Texas Instruments	Trans-impedance Amplifier	**5.30**	**1**	**5.30**
STP08DP05MTR	STMicro	LED Driver	**1.08**	**2**	**2.16**
3020-10-0100-00	CNC TECH	JTAG Port	**0.61**	**1**	**0.61**
61200621621	WURTH	UART Conn	**0.48**	**1**	**0.48**
61301011121	WURTH	10-Pin Header	**0.95**	**3**	**2.85**
L4-0-T5TH30-1	LEDsupply	505nm LED	**1.20**	**2**	**2.40**
L1-0-B5TH15-1	LEDsupply	470nm LED	**0.49**	**2**	**0.98**
L2-0-R5TH50-1	LEDsupply	660nm LED	**0.49**	**2**	**0.98**
L1-0-G5TH15-1	LEDsupply	525nm LED	**0.49**	**2**	**0.98**

L4-0-P5TH15-1	LEDsupply	440nm LED	**0.49**	**2**	**0.98**
L4-0-Y5TH30-1	LEDsupply	590nm LED	**0.49**	**2**	**0.98**
			Total Cost		**$39.57**

4.2 Labor Cost

The average salary of an Electrical Engineer graduate from Illinois is $80,000, which is approximately equivalent to $38.00/hr [5]. We estimate that we will meet weekly, averaging 4 times a week spending on average 4 hours per meeting, for a total of 16 hours per week. Over 12 weeks, this is about 16*12 = 192 hours per person. When multiplied by 2.5 we get 480, and this multiplied by 3 (since there are three group members) is 1440. To get the total labor cost, we multiply this adjusted 1440 hours total by $38/hr to get an estimate of $54,720 for labor cost.

4.3 Total Cost

The total of our project including the cost of labor comes to a total of $54,759.57. The cost per unit could also be driven down by ordering in bulk and reducing the quality of components put in place because this was a project only seeking to prove the concept, and cost analysis did not play a significant role. However, this cost is compared to a commercially available spectrometer, which is generally greater than $5000 per unit. Our design is just as effective for detecting absorbance in predetermined wavelengths, so it provides major cost savings and makes the technology available to those with small budgets.

5. Conclusion

5.1 Accomplishments

The designed LED absorption spectrometer functioned as intended and produced spectra consistent with known absorption spectrums. We were able to accomplish all of the high-level requirements we set for ourselves as well as verify that we met all of the goals of our RV table. Our project can be very easily adapted into the existing Stream prototype. The spectrometer we built gives a cheaper alternative to the medical-grade spectrometers that are available on the market and can cost thousands of dollars. Using the known absorption wavelength of proteins found in urine our spectrometer can be used to accurately detect which are present in a sample using the correct LED wavelengths.

5.2 Uncertainties

Our spectrometer can generate an absorbance spectrum that has the correct peak position, but at some other wavelengths, the absorbance spectrum is a bit off. For example, from Table 2, at 660nm, the absorbance spectrum should rapidly go down, but the spectrum generated by the LED spectrometer is not going down fast enough. There are two possible reasons. First, the blue dye that we used doesn't follow the absorption spectrum we found on the internet. It is common for manufacturers to have different recipes for blue food dye. Another reason could be the quantum efficiency of the photodiode. At certain wavelengths, the output current of the photodiode might not have a linear relationship with the light intensity that shines onto the photodiode. For example, when the light intensity is 4000mcd, the photodiode produces a 40uA current. However, when the light intensity is 8000mcd, the produced current only goes to 45uA. Therefore, using one simple equation to calculate the absorbance spectrum is not enough. That could explain why the absorbance spectrum is still high at 660nm. At 660nm, the light that passes through the sample may be decreased a lot, but it's not reflected by the current produced by the photodiode. But to get the real reason behind this problem, we will need an accurate spectrometer.

5.3 Future work

The future work on this project will involve integrating our design into the collection/pump system already built by previous STRE&M groups. This should be an easy implementation as our enclosure was designed with the previous system in mind to allow for an easy transition. Also, the resolution of the spectrometer can be improved by adding more LEDs to the design. Another aspect the Stream team wanted to be added to the project that was outside the scope of this class is the implementation of a digital microscope to magnify the sample and capture an image to be uploaded along with the absorption spectrum. We added a USB header to our PCB with this in mind so that a digital microscope can be implemented simply by plugging it in and writing the code necessary for the image capture. One last improvement that could be made to the device is to size down the PCB to get rid of any unused space which would lead to a more compact enclosure.

5.4 Ethical considerations

In considering the safety of our project we initially were planning on using a wavelength range of 380-480 nm. This would have resulted in us using LEDs in the UV range for testing purposes. We did not want to work in the lab constantly testing our UV LEDs in front of the other students exposing them to UV light. It would also have been very expensive for us to purchase the required PPE for uv light that is required to work with it safely. For these reasons, we decided to shift up our LED wavelength range. We also chose to use food coloring dye for our samples instead of biological samples as they can give us accurate absorption spectrums without the need for actual waste samples that could be hazardous. We assured that we upheld the IEEE Code of Ethics to make sure we were working safely in the lab and listening to feedback from our professors, TAs, and project sponsors [4].

6. References

[1] Garcin, C., Nicholls, F., Randall, B., Fraser, M., Griffiths, M., & Harrison, S. (2018, October). *Development of a low cost LED-photodiode based spectrophotometer ... - WRC*. Water Research Commision. Retrieved February 24, 2023, from https://www.wrc.org.za/wp-content/uploads/mdocs/KV249-10.pdf

[2] McClain, R. (2002, June). *The LED spectrometer 2010 - department of chemistry*. Department of Chemistry College of Letters & Science. Retrieved February 24, 2023, from https://www2.chem.wisc.edu/deptfiles/mcclain/Lab%204%20LED%20Spectrometer%20 2010.pdf

[3] Teja, R. (2021, September 8). *What is a photodiode? working, V-I characteristics, applications*. Electronics Hub. Retrieved February 23, 2023, from https://www.electronicshub.org/photodiode-working-characteristics-applications/

[4] IEEE Board of Directors. (2020, June). *IEEE code of Ethics*. IEEE. Retrieved February 23, 2023, from https://www.ieee.org/about/corporate/governance/p7-8.html

[5] Grainger Engineering Office of Marketing and Communications. (n.d.). *Salary averages*. Electrical & Computer Engineering | UIUC. Retrieved February 23, 2023, from https://ece.illinois.edu/admissions/why-ece/salary-averages

[6] "Esp32 Useful Wi-Fi Library Functions (Arduino IDE)." *Random Nerd Tutorials*, Random Nerd Tutorials, 23 Feb. 2023, https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/.

[7] Mayo Foundation for Medical Education and Research. (2021, October 14). *Urinalysis*. Mayo Clinic. Retrieved May 2, 2023, from https://www.mayoclinic.org/tests-procedures/urinalysis/about/pac-20384907

Appendix A Requirement and Verification Table

Table 5 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The LED Driver will sequentially light LEDs starting with the lowest wavelength (1) and ending with the highest wavelength (8).	Remove LEDs. Connect DMM across LED pads. Record current with LED activated.	Y
The circuit will supply 20mA of current for the LEDs to be emitted.	Code slow sequential lighting of the LEDs. Validate that LEDs light in order from LED 1-8.	Y
The photodiode buffer circuit will convert any producible current into a voltage value not to exceed the MCU pin rating of 3.3V.	Turn on each LED individually. For each LED, measure the voltage output of the Photodiode Buffer Circuit. If any value is above 3.3V, add a current limiting resistor.	Y
Voltage connecting to MCU can be measured continuously with less than 5% variance for 10 seconds.	Set up a photodiode buffer circuit on a benchtop. Connect DMM to the output of the buffer circuit and set it to measure voltage. Fix a light source in front of the photodiode. Collect data points for 10 seconds. Calculate the variance in measured voltage values.	Y
There is no response when we flow water into our calibrated system.	We flow water through the device repeatedly and record the response of the photodiode circuit to calibrate. We then flow a sample through the device with	Y

	absorption within our range of detection and record response. Finally, we reflow water again and confirm there is less than a 10% response from our previous water baseline.	
The data must be transferred to the computer within 2 seconds of sending data.	Initiate time capture in code at a point right before data send. Then capture the second point when the data transfer is complete to the PC. Find the time difference and confirm the Δt is less than 2 seconds.	Y
The data must be displayed in a graph that resembles the output of a standard spectrometer.	Analyze the graph for at least 2 different samples with known wavelength absorbance spectrum. Compare the wavelength value of max response for our design and known. Must be within +-15nm of the known spectrum peak.	Y
The output voltage of the 3.3V voltage regulators should be (3.3±0.2)V	When we use the voltage regulator to drive the LEDs, we could use a voltmeter to measure the voltage across the voltage regulator	Y
The output voltage of the 5V voltage regulators should be (5±0.2)V	When we use the voltage regulator to bias the op-amp, we could use a voltmeter to measure the voltage across the voltage regulator	Y

Appendix B Schedule

Week	Weekly Schedule
2/27	Designed Schematics for Each Subsystem, Finalized PCB Design Gage: Photodiode Circuit, MCU Schematic Adrian: Power System Schematic, Physical Design Model Ethan: LED Driving Schematic, PCB Design (together)
3/6	Submitted PCB Order, Began to write microcontroller code Gage: Continued spice simulations for the photodiode circuit Adrian: Begin writing MCU code, continue simulations for power/ LED circuits Ethan: Assisted in writing MCU code
3/13	Spring Break (no meetings), followed up with TA on component order No Assignments
3/20	Continued writing code that would allow for Wi-Fi connection, anticipated problems and made revisions to PCB. Design LED PCB. Gage: Aided in writing microcontroller code Adrian: Implemented Wi-Fi code for dev-board Ethan: Conducted additional research on how to display a webpage using HTML
3/27	Solder Components, Begin testing of design, Submit second PCB order, Setup LED control with MCU Gage: Soldered LED PCB, along with peripheral components Adrian: Assist in testing and writing code for LED Control, Assisted with soldering Ethan: Soldered MCU, along with LED driver
4/3	Tested function of the photodiode, redesigned the detection circuit. Continued to test and refine the other subsystems in the meantime. Changed LED spectrum. Gage: Ran more LTSpice simulations to determine the best redesign for the detection circuit. Adrian: Aided in verifying subsystems were functioning as expected. Ethan: LED Requirements verification, researched new LEDs, and placed the order.
4/10	Confirmed new photodiode circuit functions correctly, combining all subsystems together. Write code for the microcontroller to measure the output of the detection subsystem Gage: Confirmed photodiode function, assisted in system integration Adrian: Wrote code for detection measurement, aided in designed synthesis. Ethan: Solder new LEDs on PCB, assist in writing code for detection measurements
4/17	Mock Demo, Enclosure modified and fully integrated this week, Continued to work on data transfer

	to PC, Began testing system with samples with known expected spectrums. Gage: Procured materials to make our own samples with known spectrums. Adrian: Modify enclosure, write code for webpage graph, and calibration data storage. Ethan: Aided in testing, along with debugging graph display issues.
4/24	Final Testing, Adjusted graph to fix X and Y axis, Gave final demo, Began preparing for final presentation and paper due Gage: Aided in requirement verification, worked on presentation Adrian: Removed unnecessary code, confirmed we met all requirements Ethan: Researched fixing graph axes, worked on presentation
5/1	Completed Final Presentation, Submitted Final Paper Gage: Worked on presentation slides, Finalyze Paper Adrian: Worked on presentation slides, Finalyze Paper Ethan: Worked on presentation slides, Finalyze Paper

Appendix C PCB Layout

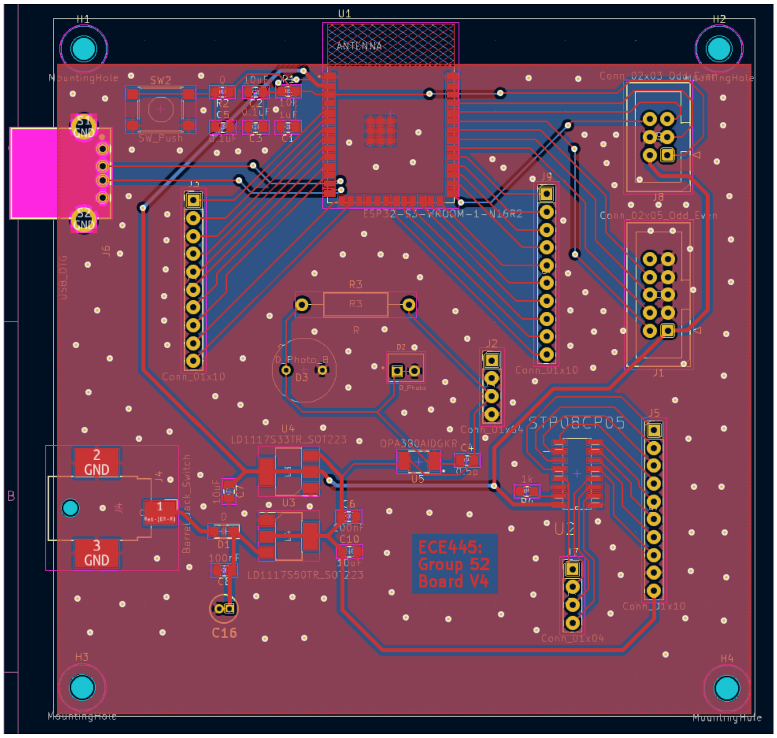


Figure 13: Main board PCB layout

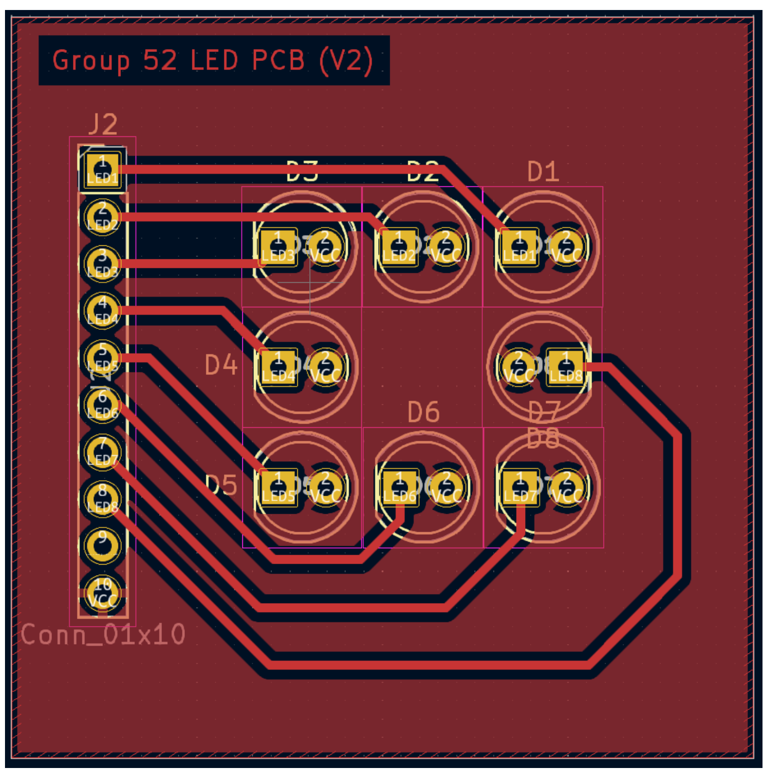


Figure 14: LED board PCB layout