

# SENIOR DESIGN II

UV-Vis Spectrophotometer



*Department of Electrical Engineering and Computer Science  
University of Central Florida  
Dr. Lei Wei  
Design Documentation*

## **Group 29**

Fall 2016

Evan Zaldivar – Computer Engineer [evanzaldivar@yahoo.com](mailto:evanzaldivar@yahoo.com)  
Jimmy Vallejo – Electrical Engineer [vallejojimmy69@gmail.com](mailto:vallejojimmy69@gmail.com)  
Sean Pope – Electrical Engineer [pope.sean.e@gmail.com](mailto:pope.sean.e@gmail.com)  
Joshua Beharry – Photonics Engineer [joshbeha@knights.ucf.edu](mailto:joshbeha@knights.ucf.edu)

## Table of Contents

1. Executive Summary.....	3
2. Project Description.....	4
a. Motivation and Goals .....	4
b. Specifications.....	6
c. House of Quality.....	8
3. Research Related to Project Definition .....	9
a. Method of Action and Common Uses .....	9
b. Customer Research .....	13
c. Market Research.....	15
d. Possible Architectures.....	16
4. Part Selection and Research .....	18
a. Sensor Processing.....	18
b. CCD Selection .....	20
c. Light Source .....	21
d. Spectrograph .....	23
i. Entrance slit.....	23
ii. Focusing elements.....	24
iii. Dispersing element .....	26
e. Buzzer Selection .....	29
f. LCD and Driver .....	31
g. MicroControl Units .....	32
h. Power Supply.....	33
i. Project Enclosure .....	34
5. Standards and Constraints .....	35
a. Standards and Impact.....	35
i. Sample Cuvettes .....	35
ii. USB Communication.....	36
iii. Power Supply and Outlet .....	37
b. Economic Considerations .....	38
c. Health and Safety.....	39
d. Manufacturability .....	40
e. Maintenance .....	41
6. Subsystem Design.....	43
a. Sensor Processing.....	43
b. Czerny-Turner Spectrometer Configuration .....	60
c. Buzzer Interface .....	62
d. MicroControl Units .....	64
i. MSP430F5528/Other Options .....	66
ii. Memory.....	69
iii. SPI/I2C.....	70
iv. Temperature .....	72

v. USB .....	73
vi. LCD Connectivity .....	77
e. Software Design.....	78
i. C software .....	78
ii. Assembly .....	79
iii. Display .....	80
f. Power Supply:.....	81
h. Summary of Design .....	87
7. Prototype Design .....	87
a. Full Schematic .....	87
b. PCB Assembly and Vendor Selection .....	90
c. Firmware Coding Plan.....	91
d. Software Coding Plan .....	94
8. Testing and Verification .....	105
a. Hardware Environment .....	105
b. Software Environment.....	107
c. Final Product Environment .....	108
d. Subsystem Testing .....	109
i. Sensor Processor .....	109
ii. Optics.....	111
iii. Light and Driver.....	113
iv. Buzzer.....	113
v. LCD and Driver.....	114
vi. Power Supply.....	115
vii. Final Software .....	117
e. Final Prototype Testing .....	118
9. Administrative Content .....	119
a. Milestone Discussion .....	119
b. Budget and Finance .....	123
10. Senior Design 2 Insight.....	124
a. Sensor Processing.....	124
b. Microcontroller .....	127
c. Final Presentation Description .....	128
References .....	130
Appendices.....	132
A. Permissions.....	132
B. Datasheets .....	138

# 1. Executive Summary

This project was chosen with a handful of goals. It was chosen as a showcase of our ability to collaborate as professional engineers and break down a complicated product into distinct parts. It was chosen to highlight each of our unique backgrounds, spanning the fields of biology, photonics, computer architecture, power systems and electronic design. It was chosen to improve upon an existing technology using modern components and research, targeting a growing audience for a unique yet widespread application. And importantly, it was chosen to demonstrate that each of us are competent engineers capable of realizing a practical solution to a problem.

This product was pitched as a moderate power piece of lab equipment suitable for desktop application with a reasonable improvement in convenience over existing products. This simple convenience – formatting, displaying, and exporting collected spectral data using USB communication – was identified as a sufficient improvement over the mid-cost spectrometer market to form a competitive product. Extensive market research showed that this was not exactly true, and that products with these features already exist at the price point we were able to achieve. Several companies exist that specialize in spectrometer design, and the improvements that we planned were already in use for recent designs. While this is promising in that the chosen improvements did create a successful product, it means that our design is not a relevant new product. This was reflected in our inability to secure a sponsorship.

Regardless, this has been a good exploration into modern systems and product integration. Several key standards and constraints were considered in forming this device, such as mains power specifications, optical cuvette compatibility, customer economics, and user safety. Several customer studies and interviews were conducted to ensure this device would be a good fit for the desired application, adding an interdisciplinary element to the project. Initially, we were quite overwhelmed at the new world of design involved in processing high-speed digital signals and producing computer interfaces from scratch. But as the research continued and we saw more of our engineering fundamentals come into play, the design came to make sense and we can proudly display this document as the fruit of our efforts. While we are under no illusions that it will work immediately without complication, we are better equipped to proceed into realization and understand whatever problems may arise.

We were warned when selecting this project that it would be very difficult. There is a great deal of careful consideration in the circuitry, and small details can cause big problems in the final system. While this quickly turned out to be true, we felt as engineers that the only thing separating us from previously successful designers was our dedication and commitment. This document contains our best efforts at understanding the devices necessary to build this product. While a larger team of more seasoned engineers might be able to produce a document twice the size with similar considerations, we felt this was a good starting point. We are eager to put these works into practice.

## 2. Project Description

### a. Motivation and Goals

Biomedical research is responsible for an increasingly significant portion of innovation in medical technology. Many modern drugs and procedures are made possible with the knowledge and manufacturing methods produced in these laboratories. One of the growing sectors of biomedical research is molecular biology, which studies the structures and behaviors of cellular structures using advanced machinery. Scientific endeavors such as the Human Genome Project rely on molecular biological research to understand the genetic information of lifeforms and how they correspond to traits and behaviors. These studies are made possible by increasingly sophisticated machinery to interact with the microscopic material, and one of the most important tools is the spectrophotometer.

Spectrometry is used in the lab to answer questions related to sample concentration, purity, and reactivity. This information is vital to understanding what is happening on the molecular level, and almost every protocol performed in a molecular lab uses a spectrometer of some kind to identify or verify experimental results. Many labs will contain several types of spectrometers, but the most common is a spectrophotometer capable of single-wavelength measurements with a minimum amount of sample usage. Often several of these devices will be available in a research laboratory due to their popularity in modern scientific protocols.

This project aims to create a spectrophotometer specifically tailored for molecular biology labs that study proteins and genetic information. This device aims to be a low-to-moderate cost model suitable for fast sample readings with a specific focus on ease-of-use and clear data presentation. The end product will have good portability and easy user configuration, resulting in simple benchtop installation with a low profile. The goal is to provide researchers a quick and easy solution to running optical spectroscopic protocols without significantly impacting overall workflow or lab efficiency.

This is not a unique concept in the market, and several products exist that try to satisfy this audience with various resolutions, price-points, and interface options. Some companies exist only to produce these instruments and related supporting material. While many of

these products are adequate for lab use and embraced by the scientific community, there is room for increased specialization and more features thanks to improvements in both biology and electronics. Embedded devices are cheaper and more powerful than ever before, and manufacturing refinements have brought many high-quality parts into an affordable domain. The market is ripe for a new product to appeal to biology researchers at a time when more studies and laboratories are being established every day. With so many studies requesting funding and so many devices required for state-of-the-art research, budgets are tight and any quality low-budget product is a welcome sight.

One of the most valuable things to a researcher is their time, as many tasks need to be completed in short order to produce good data and meet the deadlines of their investors. Many common tests done using a spectrophotometer, such as a kinetic assay for measuring enzyme activity, require collecting data points over 5-10 minute periods, but current budget models on the market only display one data point at a time. These points can be 30 seconds apart, requiring a researcher to manually sit by the device to record measurements for some simpler devices. More advanced devices may include some automation steps, but often do so via disorganized printouts that must be manually recreated for statistical analysis at a later time. These steps add up to time lost in the lab that could be put to better use elsewhere. Even a few minutes saved per assay could translate to an extra procedure performed by that researcher, improving the quality of their research and reducing mental fatigue from repetitive simple tasks.

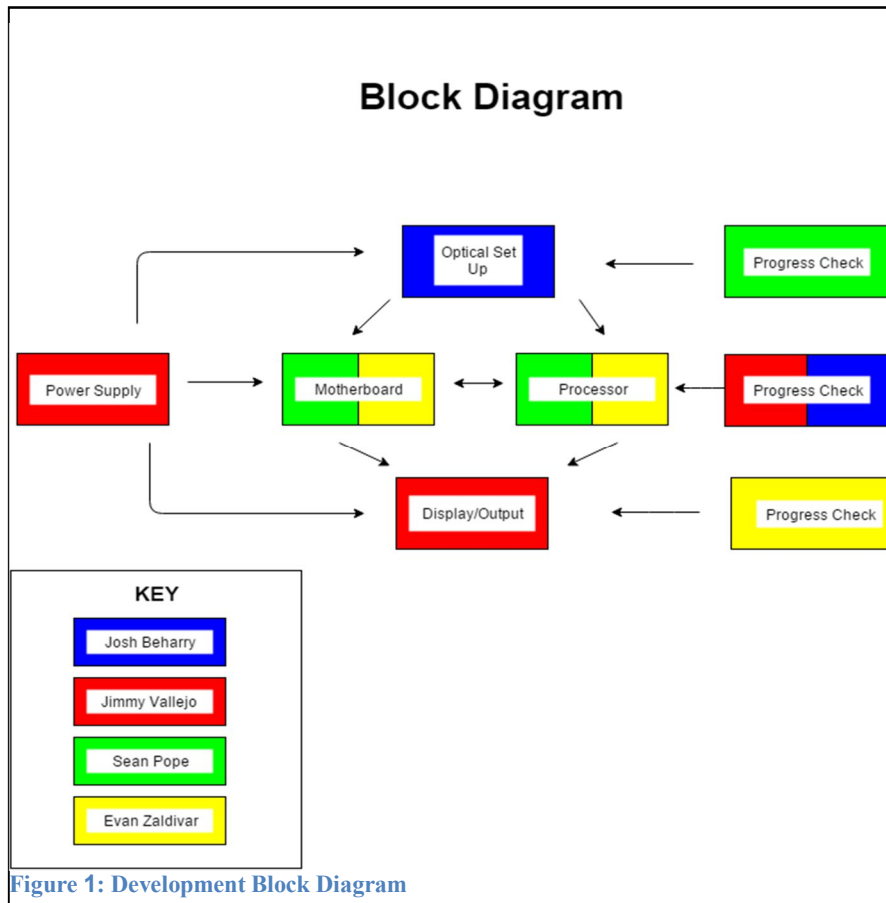
One feature that has been requested by researchers is a combination of real-time and discrete measurements when running assays, which was not possible on older devices. Using a modern display screen and a more powerful processor will enable real-time graph generation for desired applications. This information can provide more detailed understanding of reaction kinetics and reduce the chance of outliers affecting the data. Another significant improvement is compatibility with modern computer interfaces. Several models on the market only supply RS-232 output for data exports, which is considered obsolete and not featured on many computers. Upgrading to standard communication ports will improve the integration of this device into the lab environment.

Overall this project hopes to produce an attractive product for newly-established laboratories aiming to work in molecular biology or older labs seeking to upgrade their equipment. It also serves as an excellent educational tool, providing an easy-to-use measurement device with quality results that reflect good technique in the lab.

## b. Specifications

The following list of specifications was decided for this device. Figure shows a block diagram separating the system into individual sub-elements and Table shows the status of each block at the time of writing.

- USB compatibility
  - table and image output
- Weight: < 30 pounds
- Functional between 10 – 40 Celsius
- Output detection sensitive to within 0.1 AU
- Power conditions: 90-260 VAC, 47-63 Hz
- Form factor: maximum 2 ft. by 2 ft.
- Supported wavelengths: 200 – 800 nm
- Wavelength accuracy: < 10 nm
- Spectral bandwidth: < 10 nm
- Organized Display
  - Graphs of spectra captured
  - Simple user interface



Block Status		
Acquired	Optics	Display
Research	None	
Design	Software	
Prototype	Power	Motherboard

**Table 1: Development Block Status**



### c. House of Quality

The Marketing-Engineering Tradeoff Matrix is used to develop an idea to be able to correlate the requirements needed for the project. By developing the Marketing-Engineering tradeoff chart will help us overcome situations encountered while creating the project. Table shows the expected correlations of the engineering and marketing requirements for this product.

1. Marketing Requirements (▼)
  2. Engineering Requirements (►)
- ↑ = Positive correlation
  - ↑↑ = Strong Positive correlation
  - ↓ = Negative correlation
  - ↓↓ = Strong Negative correlation
  - + = Positive Polarity
  - - = Negative Polarity

(▼) (►)		Bandwidth	Speed	Weight	Compatibility	Resolution	Cost
		+	+	+	+	+	-
Install Ease	+	↑		↑↑	↑↑	↑	↓↓
Speed	+	↑↑			↑	↑↑	↓
Heat	+	↑				↑↑	
Ease of Use	+		↑↑		↑	↑↑	↓
Resolution	+	↑↑	↑↑		↑		↓↓
Weight	+	↑			↑		↓
Cost	-	↓↓	↓	↓		↓↓	↓↓
Goals		≤ 10 nm	< 5s read time	≤ 30lbs	USB	≤ 10 nm	≤ \$6000

Table 2: House of Quality Analysis

### 3. Research Related to Project Definition

#### A. Method of Action and Common Uses

The primary function of this product is fast protein and DNA quantitation by measuring the transmittance of a single wavelength of light through a homogenous liquid sample. The principle of Lambert's Law states that a medium will always absorb the same proportion of light regardless of the intensity<sup>1</sup>, as described by Equation:

$$A = \log_{10}(I_0/I) = \epsilon bc$$

Equation 1: Beer-Lambert Law for Absorbance

Where A is the absorbance of the sample, I is the measured light from the sample, I<sub>0</sub> is the incident light, b is the path length of the sample, c is the concentration, and epsilon is the molar absorptivity constant for the desired substance, sometimes called the molar extinction or attenuation coefficient. This coefficient relates the concentration for a particular substance to the absorbance, and has the unit of [volume]/ ([length][amount]).

During testing, the path length is always known and the molar absorptivity constant is generally supplied by the user before the test. It is possible that the molar absorptivity constant may be found by testing known quantities of a pure substance, which can be an option in the software for quick calculation. The majority of the intended use of this product will be finding the concentration using the absorbance at one wavelengths. Several deliverables related to this and related test protocols will be discussed below.

#### Single Scan

The simplest tests of this type take the form of presenting a sample diluted to measurable levels and finding the instantaneous absorbance at a desired wavelength. This absorbance corresponds to the concentration of unfixed biochemical compounds known to absorb light at the chosen frequency. These tests are quick and easy to perform, but are often too general and measure unwanted background particles along with the intended contents. Nevertheless, the absence of outside reagents makes these tests common in a laboratory setting.

Biologists often use colorimetric methods to improve the resolution of their concentration beyond normal spectrometric constraints<sup>2</sup>. These methods add chemicals that bind to certain characteristics of the target protein or nucleic acid and absorb a different wavelength than other elements in the sample. These wavelengths are usually in the visible light spectrum where fewer contaminants are likely to absorb the light. Table lists a few of the most popular single-scan tests<sup>3</sup>. These tests will be offered as quick presets from the

main menu of the device to improve the ergonomics of lab function, as well as customizable presets for specific applications. There is a nearly endless amount of test protocols available in scientific papers published in the last few decades, and it would be impossible to provide coverage of all of them. Rather than provide built-in presets for these protocols, a customizable preset option will be offered to allow the customer to use their desired protocols.

Single-Scan Absorbance Tests				
Name	Substance Measured	Reagents	Wavelength(nm)	Common Uses
Direct 205nm	peptide bonds	None	205	fast estimates w/ contaminants
Direct 230nm	Organic contaminants	None	230	purity assessment
Direct 260nm	DNA bases	None	260	fast estimate
Direct 280nm	aromatic amino acids	None	280	chromatography, fast estimate
Benedict	Reducing sugars	Benedict	475	general purpose
Biuret	Alkaline copper salts	Biuret, buffer	550	batch quantitation
BCA (Smith)	Copper peptide complex	BCA	562	general purpose, plate reading
Bradford	Coomassie dye	Dye, buffer	595	general purpose, cell fractions
Dische DPA	2' sugars	DPA	600	DNA vs RNA purity
Lowry	Copper peptide complex	Folin, copper, buffer	650	general purpose
Bial Orcinol	Pentose sugars	Bial	660	pentose quantiation
Modified Lowry	Copper peptide complex	Folin, copper, buffer	750	general purpose, more accurate

**Table 3: A selection of commonly performed single-scan test protocols used in protein and DNA research. The transmittance of light measured at the selected wavelength is used to find the concentration of the substance measured.**

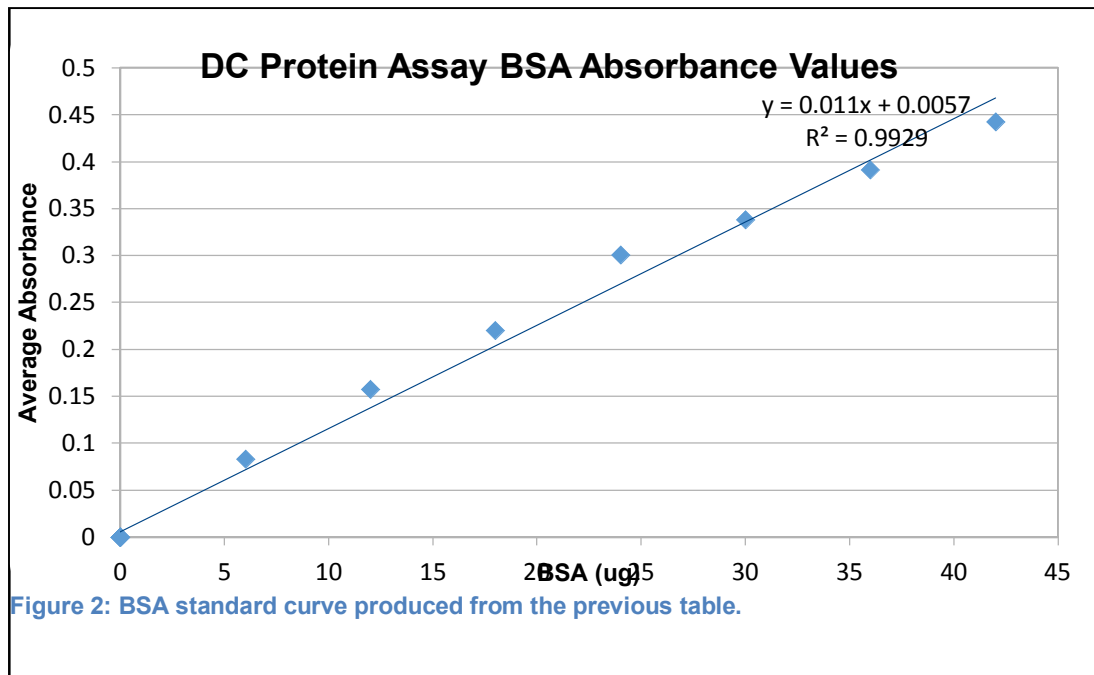
Fluorescence-based measurements have been omitted for this application as they would require additional complicated optical components and sample presentation. These tests are usually performed in a specialized gel imaging device and not a tabletop quantitation spectrophotometer. The accepted specifications for this device did not include fluorescent scanning, so it can be comfortably left out for this project.

The copper-based protein tests often require the generation of a standard curve using several known dilutions of a stock protein. Constructing this curve could be done in the software to improve workflow and immediacy of the results. One discussed feature was automating sample presentation using complex mechanical systems, but this was decided not to be feasible for this prototype and was left out of the design process.

Table shows an example of generating a standard curve using several absorbance readings on samples of BSA, a cheap and well-characterized protein commonly used for standards. It is accompanied by Figure to display the final curve that will be plotted by the final device. Both examples are from an undergraduate teaching experiment completed by a member of our group, and utilizes a protocol developed by Bio-Rad.

DC Protein Assay BSA Absorbance Values			
Sample #	BSA (ug)	Absorbance	Average Absorbance
Blank	0	0.000	0.000
1	6	0.079	0.083
2		0.087	
3	12	0.152	0.158
4		0.163	
5	18	0.221	0.221
6		0.220	
7	24	0.290	0.301
8		0.311	
9	30	0.336	0.339
10		0.341	
11	36	0.389	0.392
12		0.394	
13	42	0.450	0.443
14		0.436	

**Table 4: BSA standard curve measurements. All absorbance's were measured at 750nm using pairs of single-scan tests.**



## Enzyme Assay

One popular experimental technique for quantitation of specific proteins is the initial-rate kinetic assay. This test introduces enzymes that react with the target protein, or in the case that the target is an enzyme it uses molecules that react with that enzyme. The appropriate reagents are added to induce an enzyme-catalyzed reaction and the product that is produced by that reaction is measured over several time periods. This information is used to find the rate at which the reaction is occurring, which is described by the kinetics of the reaction equation for the target protein. With known values for the enzyme or substrate added and the rate of product formation, the unknown concentration can be found.

From a design perspective, this test can be performed as a single-scan test repeated periodically over a desired time interval. The results are plotted and the rate between two chosen points is reported and used to calculate the desired concentration. Producing these values within the device is extremely desirable by customers as the bulk of the time associated with a kinetic assay is calculating and presenting the rate and resulting concentration. The computational power of modern processors makes this feature feasible for a low-cost device.

## Threshold Alert

One popular specialization of the kinetic assay is sample incubation. A sample might be presented for measurement simply to see if the absorbance exceeds a certain threshold, usually indicating a desired optical density due to bacterial colony growth. Monitoring this incubation to assess if maturation is complete is time-consuming and can be automated by performing periodic absorbance measurements and alerting the user when the threshold has been met.

The task of alerting a user in a busy environment is most commonly met through auditory alerts. For this task, a small buzzer will be included in the device to signal test completion. This buzzer will be accompanied by a status LED to identify the specific device that has completed testing, which may be important if more than one of these devices is used in the same laboratory environment. These elements provide a level of practicality for the user in a realistic setting. Sound alerts can be undesirable in some scenarios, so volume adjustment will be necessary within the software.

The samples used in a threshold test are unlikely to be presented in a cuvette, so the optical bench will require modification to hold these samples. One method is to utilize a large compartment that does not require a closed lid to function. The additional stray light would reduce the SNR for the test, but a steep angle into the opening slit of the optical bench will reduce the impact of this complication. Reduced SNR is less of an issue for a threshold test, as incubation is not often extremely precise and coarser measurements are acceptable.

## Full Spectrum Scan

Reporting the absorbance for the entire provided spectrum is often done in qualitative studies to begin analyzing what substances are present in a sample. While this may be an infrequent application or entirely absent from the majority of protocols, it is a desired feature from users and can be easily provided in our architecture. In non-mechanical monochromatic devices, the entire spectrum is scanned simultaneously and read sequentially using a shift register, so spectrum scans are done by default. Reporting the entire spectrum is also desirable as a demo feature or debugging tool to assess the quality of the system.

Collecting and presenting spectral scans can be done with the same infrastructure as time-based single-scan testing, but any periodic spectral scans would be difficult to present. As this is not a desired feature in any known literature, and the memory requirements to provide such results would be very limiting, periodic spectral data will not be innately supported for this application. The user is free to repeat spectral scans as desired, but complex data presentation will not be included.

## Range Requirements

Two approaches were used to characterize the spectral range of the device to fulfill the desired applications. The first was to compile a list of common test protocols and provide enough range to perform each of them in turn. This was accomplished through inspection of Table and equaled to the expected 200-800nm requested in the customer interview. The second method was to analyze existing products in the market and attempt to match the average delivered spectral range. This revealed a similar range with occasionally larger wavelengths, spanning into IR at 1100. As this was not requested by any customer, the first range will be used.

## b. Customer Research

To better understand the desired characteristics of the device, a few molecular biology researchers with experience using similar devices were interviewed. The results of these interviews have been tabulated and presented below. The consequences of these results are also discussed.

## Interview Results

Personnel interviewed:

Dr. Sean D. Moore (UCF) and Dr. Robert Borgon (UCF)

- What are the primary tasks that this device will be used for?  
Protein and DNA quantitation, cloudiness assessment, turbidity measurements

- What spectral requirements do you have for an adequate device?  
At least 200-800nm range. Minimize bandwidth, maximize accuracy and precision
- What sources of damage or failure are most frequently encountered?  
Broken cuvettes in compartment, external wiring torn or entangled, bulb failure and prohibitive replacement.
- What manufacturers do you find make adequate products for your needs?  
Biotek, Agilent, Bio-Rad
- What workflow and useability improvements can you recommend?  
Reduced post-analysis formatting, full standalone functionality, simple but flexible user interface, no overabundance of settings, accurate threshold indication/alarm, fast full spectrum scan, in-vitro incubation monitor.

## Interpretation

The answers recorded for the uses of the device can all be completed with a spectrum generated over the desired range. Turbidity is usually assessed using a kinetic assay during incubation, which is provided in this application through periodic measurement plotting. Quantitation methods are discussed in the method of action section of this documentation.

The final device will only require one power cable to function independently, which is a standard heavy-gauge insulated cable resistant to tearing. USB connection will be intermittent and usually expected to be with a portable USB flash drive, which reduces points of failure and entanglement. Broken cuvettes will require a watertight compartment with a minimum of divots to catch solution and broken quartz. Bulb failure can be reduced with minimal light pulses to reduce stress, and a door will be provided to replace the bulb if necessary.

The desire for reduced post-analysis formatting was the initial inspiration for this device, and will be addressed with improved communication and figure generation over existing models. The threshold feature is discussed in the method of action research. Designing a GUI capable of customization without overwhelming the user requires good isolation of the settings from normal operation, which will be discussed during the UI layout section. The mentioned manufactures will be researched for additional design suggestions

### c. Market Research

The initially suggested retail price for this device was decided as somewhere between \$1000 and \$6000, which is a popular price range for devices of this quality. Table and Table (split for good print resolution) show a comparison containing several other devices in this price range that form potential competing products.

Name	Manufacturer	Price (\$)	Range Low (nm)	Range High (nm)
SmartSpec Plus	Bio-Rad	5900	200	800
NanoDrop One	Thermo	15000	190	850
Nanodrop 2000	Thermo	10000	190	840
Nanodrop Lite	Thermo	5000	260	280
Spectronic 20	Thermo	2100	340	950
S2150-UV	Unico	4000	200	1000
SQ-4802	Unico	8000	190	1100
Genesys 10S	Thermo	6000	190	1100
UV/Vis	Cole-Parmer	5300	190	1100
6305	Jenway	3000	198	1000

**Table 5: A selection of low-to-mid budget spectrometers from several vendors. Prices are listed as quotes for individual purchase.**

Name	Bandwidth (nm)	Accuracy (+/- AU)	Display	Interface	Other
SmartSpec Plus	5	0.01	2x24 LED	RS-232	Printer
NanoDrop One	1.8	0.03	Tablet	Ethernet, USB, WiFi	Small sample
Nanodrop 2000	1.8	0.02	None	USB	Small sample
Nanodrop Lite	8	0.03	Display	None	Small sample
Spectronic 20	20	??	Meter, 3 LED	RS-232	Analog
S2150-UV	4	0.04	LCD	USB, Parallel	
SQ-4802	1.8	0.003	LCD	USB, RS-232	Dual Beam
Genesys 10S	1.8	0.05	LCD	USB, RS-232	Dual beam
UV/Vis	4	??	LCD	USB, RS-232, Parallel	
6305	8	0.01	LCD	RS-232	

**Table 6: Continuation from above**

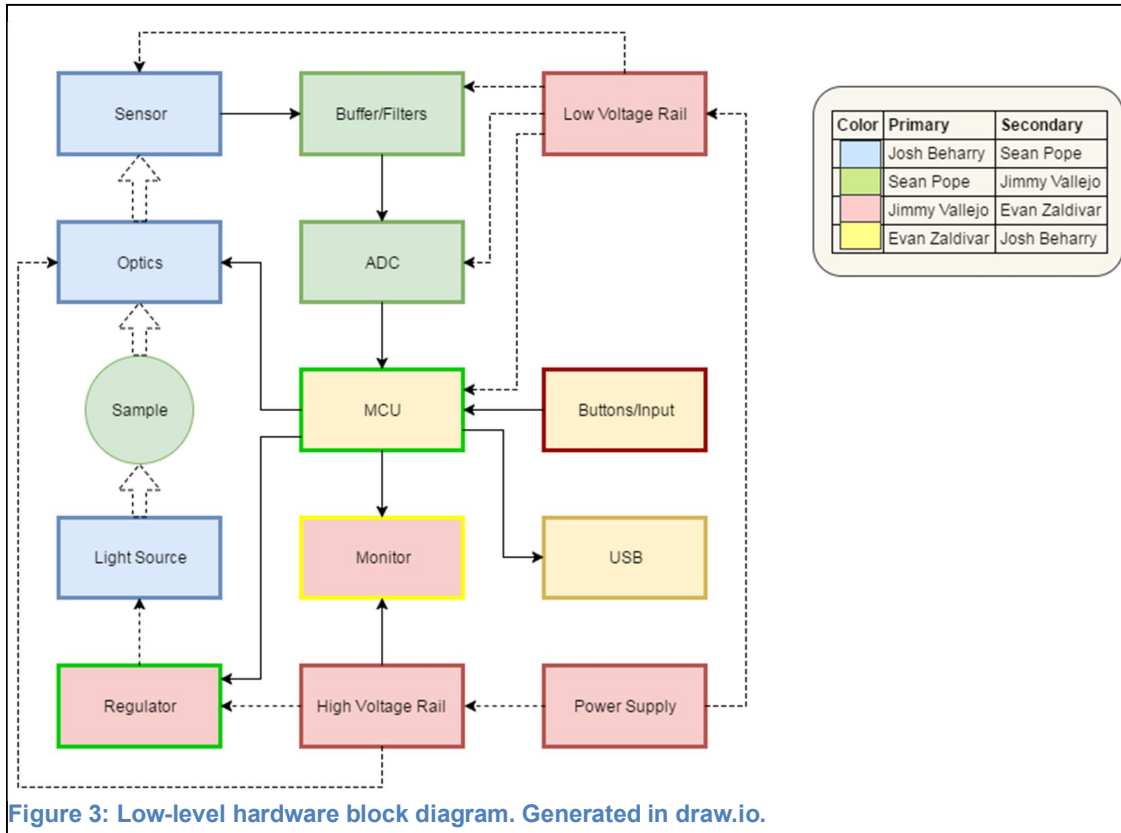
A few of the higher-end Nano drop models were included to show how user interface improvements with no noticeable spectral improvements can cause a great increase in price. A few common traits are a spectral range that reaches 200 nm (except the Spectronic 20, which is obsolete at the time of writing but still for sale occasionally) and interestingly a high range of 1100 for several models. The Nano drop Lite is unique in only offering 260-280 nm range, which is the peak absorbance range of untreated DNA and proteins. This is a convenient budget device for high-accuracy measurements in a popular range, but not a flexible instrument.

A wide range of interface and display options were noted, ranging from no stand-alone interface to high-resolution LCD with USB interface. The ever-dropping price range of microcontrollers and improved capabilities makes these user interfaces cheaper to include, and it is becoming a popular trend among spectrometer manufacturers to include these in budget devices.



## d. Possible Architectures

Several possible methods for designing this device are discussed in this section. The initial block diagram breaks the total device into a number of subsections, each of which has historically seen several methods of realization in commercial products. A few of those architectural topologies stand out as more appropriate for the application but each identified method will be considered as appropriate. Figure shows an early hardware flow model that formed the basis of the selected architecture.



## Sample Presentation

A few of the recent products in the molecular biology spectroscopy market feature extremely short path-length optical setups, such as the Nano Drop or any type of microplate reader. These setups are convenient because they require less of a particular sample to be rendered useless for the test procedure, and no dilution process is necessary to achieve the desired optical density. Other advantages of this approach include no dependence on external optical vessels like quartz cuvettes that must be maintained and cleaned between

uses. The sample presentation stage must be cleaned, but the smaller form factor makes this easier in practice than cleaning a 10mm cuvette.

While this ease-of-use is attractive for our product, short path-length devices suffer from inconsistent and generally inaccurate measurements caused by poor homogenization and excessive light scattering of the incident beam. These issues can sometimes be reduced by taking multiple readings and averaging them, but the time disparity and lack of confidence is generally not acceptable, even considering the benefits. Additionally, any colorimetric methods further reduce the accuracy due to the small volume, so these short path-length devices are undesirable for this application. Instead, support will be offered for the standard 10mm volume cuvette, and any compatible third-party adapters that the customer chooses to use.

## Wavelength Selection

The monochromator setup itself has seen great improvements in recent years. Early monochromators for spectrophotometers generally used the Czerny-Turner setup with the grating mounted on an extremely precise stepper motor to adjust the wavelength that led to the detector. In practice these motors had to be extremely precise, as individual wavelengths were often only a few arcminutes apart. The mechanical systems required to guarantee accurate wavelength selection generally included complicated set point control systems and a speed-reducing gearbox to achieve small adjustments. Better selection values were achieved by reducing gearbox backlash and controller noise in the feedback system.

Modern optical setups generally feature array detectors, such as the charge-coupled device (CCD) or photodiode array (PDA). These detectors feature several elements in a series that are read out individually at a very high speed. This technology led to monochromator gratings that were fixed in place and the whole spectrum spread over the detector at once. Now the accuracy of the system is decoupled from mechanical accuracy, instead relying heavily on detector quality and processing. This setup was preferred for our application as our team has little experience with mechanical systems but good coverage of electrical signal processing. Additionally, detector quality is now high enough to produce competitive selectivity with mechanical systems, especially at the desired price point for this device.

Unfortunately our design initially attempted to create a scanning monochromator but the final design was deemed unacceptably complicated for this project. Getting a precise wavelength selection from a motor requires a very accurate mechanical system with multiple control elements and careful construction to produce good selectivity. Many affordable servomotors cannot reach this level of accuracy, and designing a mechanical compensator was not ideal for our group. The CCD detector was chosen to remedy this issue, resulting in a new batch of designs and considerations for this topology. Wavelength

selection in budget spectrometer devices has been a waning trend in market devices, as array detectors have become cheaper and reduce the mechanical requirements of a product.

## Signal Processing Methods

Coordinating the optical setup and processing the signal from the detector is most commonly implemented with a microcontroller, but other solutions were considered before going forward with that choice. The entire product could feasibly be realized with a powerful FPGA. The benefits of such an arrangement include excellent parallization of the control and processing blocks and great response time on the digital signal processing arrangement. The biggest argument against this realization is the requirement for USB communication and LCD output, which would be difficult and inefficient to implement in the FPGA itself. Even with supporting drivers and electronics the higher power consumption and lower calculation efficiency of the general hardware make it undesirable for this system.

Using dedicated hardware like a microcontroller with on-board support for USB communication and GUI production is a better choice, especially considering the relatively higher cost of FPGAs compared to MCUs. Dedicated hardware solutions generally feature better performance over emulated hardware in terms of cost, power efficiency, calculation efficiency, and development complexity, and the built-in features of the MCU make it a clear choice for this project.

## 4. Part Selection and Research

### a. Sensor Processing

The most important component in this processing scheme, outside of the MCU itself, is the amplifier used to match the dynamic range to that of the ADC. This amplifier is bandwidth-limiting and must introduce no additional noise to the signal. Table shows the requirements derived in the design section.

Parameter	Value
Slew Rate	6 V/us
Bandwidth	3 MHz
Rail Voltage	6-10 V
Noise Voltage	700Vn/rtHz

**Table 7: Sensor Processing Amplifier Specifications**

Several products that were adequate for these specifications are presented in Table below.

Name	Manufacturer	Slew rate (V/us)	Unity Bandwidth (MHz)	Noise Voltage (nV/rtHz)	Max Rail (V)	Price (\$)
TL084	TI	13	3	18	36	0.17
TL074	TI	13	3	18	36	0.17
AD8390	Analog	260	45	5	24	1.08
LTC1254	LT	250	90	3	28	4.40

**Table 8: Amplifier options**

Several low-noise and high-speed op-amps were acquired and tested for this purpose, but many of them are only available in TSSOP packages, which proved very difficult to work with. In the interest of producing a good prototype, a standard TL084 quad op-amp from Texas Instruments is used for this processor. This amplifier has adequate specs for the sensor processor and is available in a convenient DIP format. The suggested retail price of this chip is \$0.17 for a run of 1000 units. The real product is likely to use either this or the TL074, which had very similar characteristics but a slightly lower input current noise. For the characteristics of this design the lower noise would be excellent, but the TL084 was available for testing and prototyping so it will be used for our schematic.

The FETs used to control the signal flow can be realized using 4 common 2N7000 family n-channel FETs. These are extremely available, and will be quoted as \$0.01 each, though the likely production cost for 1000 or more is potentially even lower. No special resistors or capacitors were identified for this application.

In the interest of considering all options, Table and Table show a few options for complete integrated systems that could be used in place of the designed processor.

Product	Vendor	Bandwidth	Max Gain	Approx. Cost (\$)	Voltage Max (V)	Power @ max (mW)
AD9945	Analog	40	4.5	4.62	3.6	140
TLV990-13	TI	13	36	4.04	3.3	150

**Table 9: IC options for signal processor**

Product	Package	Pins	ADC Resolution (bits)
AD9945	LFCSP	32	12
TLV990-13	TQFP	48	10

**Table 10: Continued table from above for formatting**

The existence of a cheap complete integrated analog front end is a newer consideration on the market. Having a robust ADC built in to the front end reduces the requirements for the MCU that has to be selected, and the parallel digital readout is very easy for IC

communication. In a real iteration of this product design, it is likely that one of these ICs would be selected to produce a simple, error-free device. Things like black level calibration and noise trimming are done in software, making it a user-friendly device to program. For this project, these ICs will only be used in the case that the discrete design does not work.

For the initially posted design, the sensor processing will be replaced with a simple two-stage dynamic range matching system and AC coupling. This may change in future designs, but this design will still be considered for final layout. Both the TL072 and LF351 amplifiers were considered for this design, but those chips retail for \$0.31 and \$0.40 (for a pair two) respectively, which does not beat the price of the quad op-amp package of the TL074. While quad packages suffer from increased crosstalk and poorer thermal characteristics, these traits were not deemed to be problematic for this design and the entire TL074 package can be easily fitted into the final board space. While it seems inefficient, the cost-benefit analysis reveals this to be the best choice.

## b. CCD Selection

An essential piece of equipment in a spectrophotometer is the photodetector of the system. The light may not be quantifiable to the human eye, so the detector enables a human to understand and analyze light. Today, linear array detectors are mainly used in spectrometer-based systems today. A linear array refers to a fixed amount of pixels in a line. Each pixel can convert the light energy into an electrical signal. As a result, many different light levels can be measured and analyzed. Before linear arrays, spectroscopy was done with a photodetector with a single photo detecting element. In other words, the detector is only able to measure one light level at a certain rate. With this detector, the spectrometer would have to rotate the dispersing element. Light of different wavelengths would be incident on the detector at a certain rate. Therefore, the light intensity can be captured as a function of wavelength, and the data can be processed into a spectrum understandable to the consumer. This method of spectroscopy is difficult to fabricate and obsolete by today's standards. The linear array detectors allow for spectrophotometers to be fabricated without moving parts, such as a rotation stage. Consequently, the size and cost of a spectrophotometer are reduced.

The two main linear array detector technologies are the photodiode array (PDA) and the charge coupled device (CCD). The PDA can be considered the traditional linear detector. A PDA is a linear array of photodiodes. A *pin* photodiode structure will be considered in this discussion. The photodiode structure consists of a *p* region and an *n* region, as mentioned in the semiconductor field. However, these two regions are separated by a large region that is very lightly doped. This region is called the intrinsic (*i*) region. Light is incident on the *p* region of the device, but it is very thin so that there is little absorption. Photons are supposed to be absorbed in the *i* region. If a photon has energy greater than or equal to the bandgap energy of the material, an electron may be excited from the valence band to the conduction band. These photo carriers, charge carriers that are generated via

light, can be used to produce current. Photo carriers are generated mainly in the depletion region, which is the *i* region in this device, because most of the light is absorbed here. The photo carriers can separate and flow when an external voltage is applied, giving rise to a photocurrent.

The charge coupled device (CCD) is the more popular choice of detector array when it comes to spectrometers today. CCDs are usually created on a silicon substrate. An integrated circuit made with photosensitive elements (pixels) is etched on the substrate. Each individual pixel has a corresponding potential well, which can also be considered as a capacitor in this case. Photons are incident on the pixels for a certain amount of time, called the integration time. An electric charge proportional to the intensity of the light is accumulated. At the end of the integration time, the charges must be read out to be analyzed. A circuit causes the charge of each capacitor to shift to its neighboring capacitor. The last capacitor's charge is fed into a charge amplifier to be converted into a voltage. This process is repeated for the entire array, and a sequence of voltages is created. This voltage signal can then be processed and analyzed by someone.

There are advantages and disadvantages between PDAs and CCDs, and these drastically affected our choice of detector. Both types have a similar responsivity curve with respect to the wavelength of light. However, CCDs are a little more responsive in the UV range. PDAs have a larger dynamic range and better linear response than CCDs. In addition, the readout speed is faster than CCDs. The average CCD array contains many more pixels than that of a PDA, which can help in the resolution of the output. In addition, a CCD has an overall better sensitivity than a PDA. A disadvantage of CCD is that the SNR is poorer than that of a PDA.

The linear array detector we are using for our spectrophotometer is the Toshiba TCD1304AP CCD linear image sensor. One reason we chose this is because this detector is used in a spectrometer on the market today. Other reasons for the choice are related to CCD technology. CCD detectors are the state-of-the-art, so we would like to create a device that is relevant and advanced.

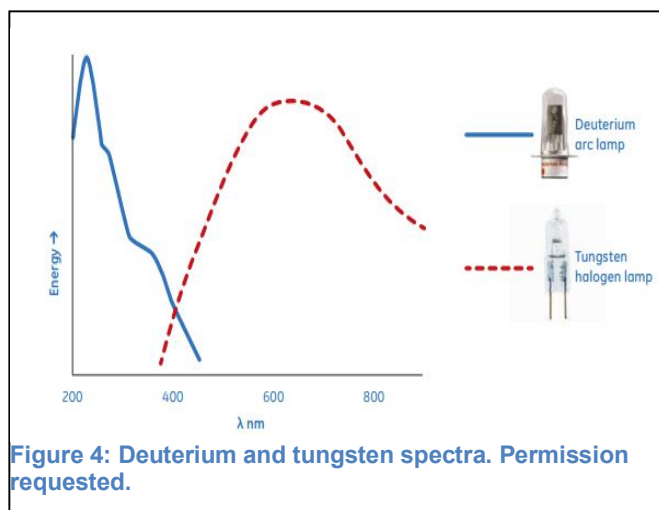
It is known that cooling an array detector reduces dark noise, thus increasing signal-to-noise ratio. Cooling also enhances dynamic range, and enhances the detection limit. It has been demonstrated that cooling a CCD to 10° C can reduce the dark current by about four times and the dark noise by about half the amount, compared to a CCD at room temperature. We will consider this as an additional feature to our spectrophotometer if time and money allow for it. We may also consider if the SNR of our device is an unacceptable level.

### c. Light Source

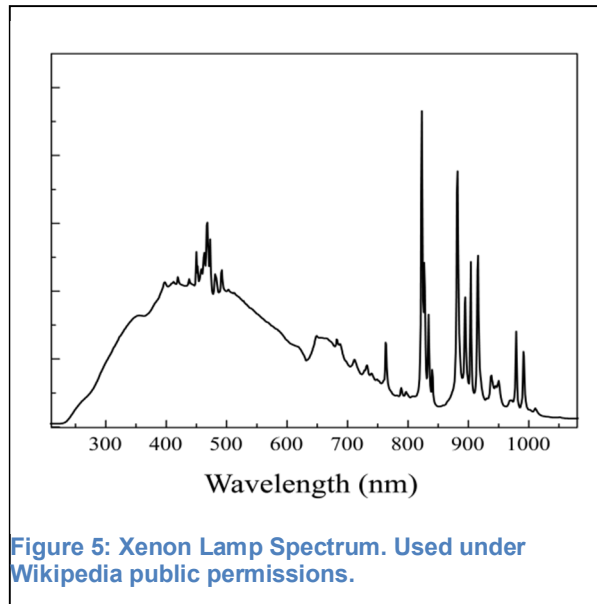
A suitable light source that covers the UV and visible part of the electromagnetic spectrum is of interest for this application. Thus, monochromatic sources such as most light emitting diodes or lasers will not be suitable for this application. We turn to lamps as the light source

for a spectrophotometer. In general, the intensity of the light should be stable throughout the measurement period. There should also be an adequate intensity over the needed wavelength range.

We will go over all the lamps used in UV-VIS spectrophotometry. To begin, there are deuterium lamps. These types of lamps provide powerful and stable light between 190 nm to 370 nm. There is a special type of deuterium lamp that can last 3 times longer than conventional deuterium lamps. The lamp is kept warm when not in use, but it is not actually illuminating. Deuterium lamps paired with tungsten lamps allow full coverage of the UV-VIS electromagnetic spectrum. Tungsten lamps illuminate powerful and stable light between 320 nm to 1100 nm. When using the two lamp setup, the lamps are setup on separate mounts. Mechanical equipment is used to manipulate the position of the lamps and eliminate cross-talk between sources. The spectra for both lamps are shown in Figure.



There is a better solution to provide the light needed for UV-VIS spectrophotometry. The release of xenon lamp sources allows the use of the wavelength range from UV to near infrared with one lamp. The electromagnetic spectrum is shown in Figure. Xenon lamps have a long lifetime compared to a conventional deuterium or tungsten lamp. However, there are many disadvantages with using the xenon lamp. It does not have great intensity stability like the other two lamps, so measurements can be suspect. It is up to the electronics to stabilize the lamp as much as possible. In addition, the stray light is higher, but it is not that big of a problem if the measurement cycle is short. We will be using a xenon lamp in our spectrophotometer. At this time, they are commercially available. Despite the disadvantages, it is still a better option than manipulating two separate light sources.



The light source we will be using is the Bulb America H7 bulb. It is a halogen xenon light bulb. It is rated at 12 V and 55 W, so it is safe to handle and install. The luminous flux is rated at 500 lumens, which can be considered bright enough for our application. The price is also definitely in our budget. High-end spectrophotometers use lamps that cost over \$100! It may be some sort of advanced bulb, but the function is all the same.

## d. Spectrograph

The basic purpose of a spectrograph is to take in some light and separate the light into components that will be recorded on a plate or detector. We have seen that the light source supplies a broad range of wavelengths, from UV to the visible range. The spectrograph will separate the white light into different wavelength components to be collected and analyzed by the array detector. The spectrograph is made of several essential components that allow for this process. These components include an entrance slit, focusing elements, and a dispersing element.

### i. Entrance slit

The first element in the spectrometer is an entrance slit. The entrance slit determines the amount of light that will enter the spectrograph. It represents a clear, known object that will be imaged by the spectrograph. The slit width can vary from 5  $\mu\text{m}$  to 800  $\mu\text{m}$ , usually. The image width of the slit can affect the resolution of the spectrograph. We would like to match the image width to the pixel width of the detector. Therefore, we should pick a slit that will allow for the best resolution we can possibly achieve. We can model the image width by Equation:



$$W_i = (M^2 W_s^2 + W_o^2)^2$$

Equation 2: Image Width

$W_i$  represents the image width of the slit,  $M$  represents the magnification of spectrograph,  $W_s$  is the width of the entrance slit, and  $W_o$  is the image broadening caused by the spectrograph. If we set  $W_i$ , the image slit width, equal to the pixel width of the detector, a good estimate of the width of the entrance slit can be achieved. The magnification of the spectrograph,  $M$ , is determined by the focal length of the focusing elements.

## ii. Focusing elements

Focusing elements include lenses and mirrors. For the purposes of this project, we will consider mirrors as our focusing element. Mirrors are both budget-friendly and easier to handle. Other than that, there is no real advantage of using lenses or mirrors. However, popular configurations of the spectrograph use mirrors. The configuration will be explained in the subsystem design section.

For this project, we will be using focusing, or concave, mirrors. This is also synonymous with spherical mirrors. Concave mirrors are usually used in imaging systems because the chromatic aberrations produced by the mirrors are little to none. The first mirror of the system, called the collimating mirror, comes after the entrance slit. As the name entails, it collimates the divergent light from the slit onto the grating. The mirror that proceeds the grating is called focusing mirror. This mirror will focus the dispersed light onto either the exit slit or the detector itself.

All spherical mirrors, either concave or convex, can be described by Equation, called the mirror equation:

$$\frac{1}{o} + \frac{1}{i} = \frac{-2}{R}$$

Equation 3: Mirror Equation

Where ‘ $o$ ’ represents the distance the object is from the vertex of the mirror, ‘ $i$ ’ represents the distance from the image to the vertex, and ‘ $R$ ’ represents the radius of curvature of the mirror. The radius of curvature of the mirror defines the focal length of the mirror with the relationship in Equation:

$$\frac{-R}{2} = f \begin{cases} f < 0, \text{concavemirror} \\ f > 0, \text{convexmirror} \end{cases}$$

Equation 4: Focal Length

Therefore, we can rewrite the mirror equation in terms of the focal length, as in Equation:

$$\frac{1}{o} + \frac{1}{i} = \frac{1}{f}$$

Equation 5: Mirror Equation with Focal Length

With a given object distance and focal length of a mirror, we can calculate where the image will be formed. Let us consider the focusing mirror of the spectrograph. The focusing mirror will focus the light coming from the grating onto a detector. We can consider the light from the grating as light of parallel rays. This is because the collimating mirror produces the parallel rays of light that hit the grating. The grating will disperse the different wavelengths of light in parallel rays, also. In geometrical optics, parallel rays of light correspond to an object distance of infinity. Let us plug this into the mirror equation to produce Equation:

$$\frac{1}{\infty} + \frac{1}{i} = 0 + \frac{1}{f} \text{ yields } i = f$$

Equation 6: Mirror Equation for Infinite Distance

If the object seems to be from infinity, the image of the object will form at the focal plane of the mirror. As noted in a previous equation, the focal length of a concave mirror is a positive value. Thus, the image distance is also a positive value. By convention, this means that the image formed by the mirror is a real image. A real image means an image is formed by the actual light rays, not the projection of light rays (which are called virtual images). Therefore, we can place our detector at the focal length of the mirror to be able to capture the image of the dispersed light.

Other details of mirrors to consider are the surface specifications and the reflectance at certain wavelengths. Surface flatness is one type of surface specification, and it specifies the deviation from a flat surface. Most optics used for precision have a flatness grade of  $\lambda/4$  or  $\lambda/8$ , where  $\lambda$  is the wavelength of the source used to test flatness (usually 633 nm). A mirror at this flatness grade or better will be suitable for the spectrophotometer. Surface quality is another surface specification. Scratches and digs do not affect the performance of a system in a significant fashion, but they should be minimized if possible. Finally, the reflectance of a mirror is important since many applications are limited to a certain wavelength range. The reflectance is determined by the material of the mirror coating.

The usual materials that make a coating are aluminum and gold. Gold-based coatings are usually used for infrared applications and can provide reflectance's over 95%. Gold-coated mirrors are more expensive than aluminum-coated mirrors. Aluminum-coated mirrors are used from the visible range to the infrared range. It is cheaper than gold, but the reflectance is not of the same quality. The usual reflectance for aluminum-coated mirrors is over 85% in the visible range and over 70% in the near to mid UV range. A development called UV Enhanced Aluminum has increased the reflectance in the UV wavelengths to over 85%. Shown in Figure below is a graph by Edmund Optics, which show reflectance's for different coating materials in the wavelength range from UV to near IR. In conclusion, a

spherical, concave mirror with UV Enhanced Aluminum coating and standard surface specifications will be both ideal and cost-effective for our project.

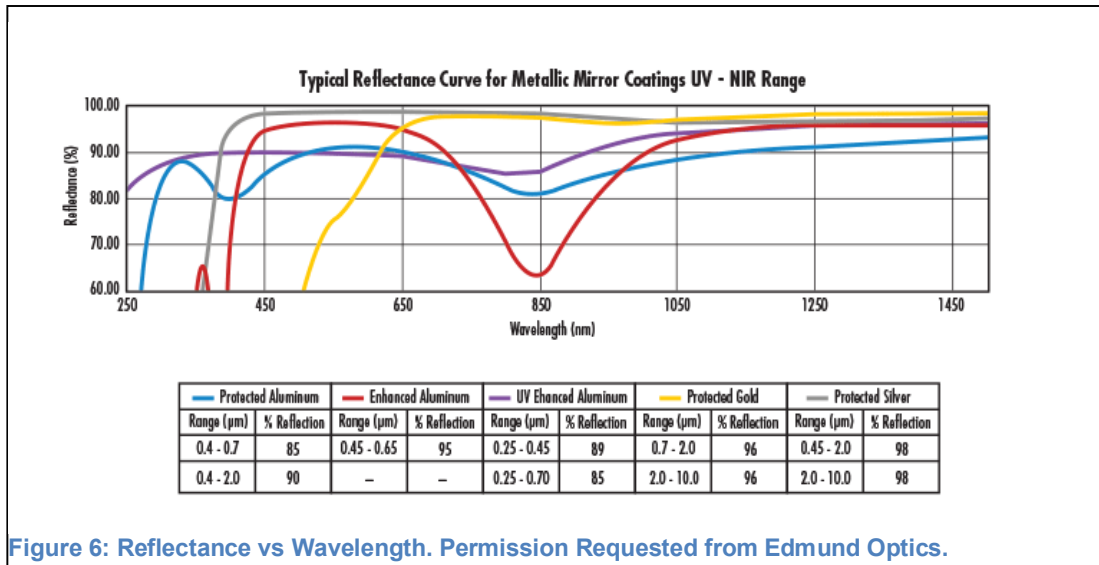


Figure 6: Reflectance vs Wavelength. Permission Requested from Edmund Optics.

### iii. Dispersing element

The dispersing element is the key component involved in any sort of wavelength selective or spectra-based instrumentation. A dispersing element of a spectrograph or spectrometer determines the wavelength range of the system and partially determines the optical resolution. The ability to select a certain wavelength or range of wavelengths from a broadband source is important to different fields in optics and photonics, such as spectroscopy. There are three different ways in which wavelengths can be selected. These three ways are dichroic filters, prisms, and diffraction gratings. Prisms and diffraction gratings are the two devices that actually use the phenomenon of dispersion. Filters are included in this discussion because it is still a method of wavelength selection.

A filter used by wavelength selectivity is called a dichroic filter. It is, basically, a colored piece of glass or gelatin. It transmits light only from a small range of wavelengths and reflects the rest of the light. The problems associated with dichroic filters include a restriction to the visible range (because it is colored glass) and the spectral bandwidth is too large for accurate and precise measurements by spectroscopic instrumentation. The spectral bandwidth of a dichroic filter can exceed 40 nm.

A prism is a suitable device that can provide a continuous spectrum and well-separated wavelength components. A prism uses the concept of material dispersion to separate wavelengths. Material dispersion is the phenomenon where different wavelengths of light travel at different speeds through a certain material, such as a prism made of glass. Due to the differences in speed and the law of refraction, the electromagnetic waves of different

wavelengths move at different angles within the prism. When the light exits the prism, the law of refraction describes that the light will be even more spread out. This basic explanation of dispersion also describes how rainbows are created. It is practical to use the prism to just have the ability to separate white light into a large spectrum to be incident on the detector. The addition of a rotation mechanism and exit slit allow for single wavelength selectivity. Prisms can achieve spectral bandwidths of 1 nm or better, and greatly improve on a dichroic filter for selection. The downsides to a prism spectrometer are the difficulty of controlling the wavelength, the non-linear effects of the prism, and temperature related issues of the prism materials. For these reasons, we turn to diffraction gratings as another option to produce dispersed light.

A diffraction grating is a periodic series of grooves that act as many miniature mirrors. The two types of diffraction gratings are ruled gratings and holographic gratings. Ruled gratings are the classic example of a diffraction grating. The ruled grating is created by physically cutting grooves into the reflective layer of the grating substrate. They are simpler and cheaper to fabricate than holographic gratings, but they introduce more stray light into the system. Stray light is any unwanted light that reaches the detector. This is not much a problem if the grating is not used in a monochromator configuration. Ruled gratings are also used when the groove frequency needed is low. Groove frequency will be explained later. Diffraction efficiencies of 85% can be achieved. Holographic gratings are produced by interfering ultraviolet beams in the glass to produce a modulated index of refraction that acts like a large number of grooves. Holographic gratings are preferred for applications that need a large groove frequency, low stray light, or a concave grating. A concave grating is useful in the sense that it can work as a focusing element, as well as the dispersing element. The stark downside of a holographic grating is that the maximum diffraction efficiency is only around 35%. Some of the reasons diffraction gratings are preferred over prisms are the better resolution they supply, the lower level of stray light, and a linear dispersion that leads to a constant bandwidth.

The groove frequency is one of the main properties of a diffraction grating. The groove frequency is usually represented in the unit of grooves per mm. It determines the wavelength range and the spectral resolution of the system. The groove frequency is directly proportional to the dispersion, directly proportional to optical resolution, and inversely proportional to the wavelength range. The linear dispersion of the grating is important to consider when the light is supposed to be incident on a photographic plate or an array detector. The linear dispersion of the system, in terms of nm per mm, can be expressed as in Equation:

$$\frac{d\lambda}{dL} = \frac{10^6 d \cos\beta}{mF}$$

Equation 7: Linear Dispersion

Where 'd' is the groove period (inverse of the groove frequency), 'β' is the diffraction angle, 'm' is the diffraction order, and 'F' is the focal length of the focusing mirror. By

multiplying the length of the detector by the linear dispersion as in Equation, we can get an idea of the spectral range that is able to be detected.

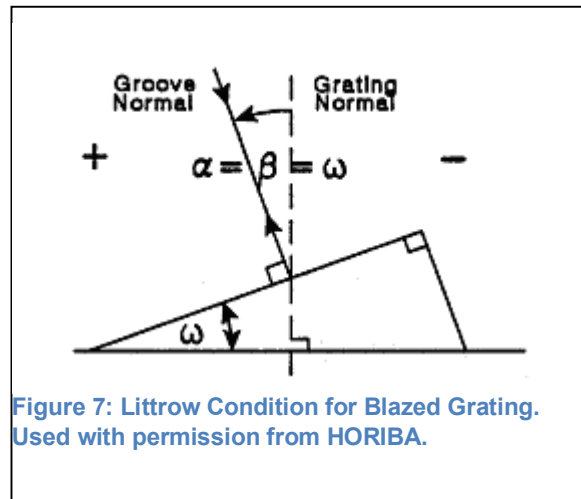
Given that this project is based in the UV and visible spectra, we already have an idea of the spectral range we want to achieve. We are looking at a spectral range of about 450 nm. With a given detector length, we can calculate the linear dispersion needed for this spectral range, and choose a diffraction grating accordingly.

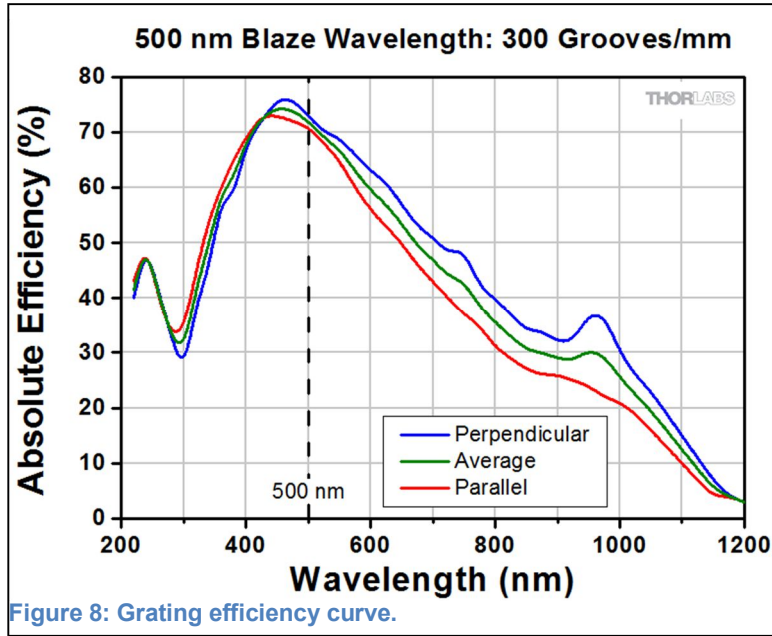
$$\lambda_{max} - \lambda_{min} = L_D \frac{d\lambda}{dL}$$

Equation 8: Spectral Range Derivation

Other important properties of a diffraction grating are the blaze angle and blaze wavelength. The blaze angle is defined as the angle between the normal of the grating substrate and the normal of the grooves, shown in Figure. From the blaze angle, a blaze wavelength is established. At this wavelength, the diffraction efficiency of the grating is at its highest. The strength of the diffraction decreases at wavelengths away from the blaze wavelength. In general, the maximum diffraction efficiency is cut in half at 2/3 of the blaze wavelength and at 1.8 times the blaze wavelength. It is best to not use wavelengths outside of this range. A blaze wavelength of 400 nm or 500 nm will allow for a wavelength range to stretch encompass the near UV and the entire visible the range of wavelengths we are interested in using for our project.

The diffraction grating we chose for our spectrophotometer is the ThorLabs GR25-0305 Ruled Reflective Diffraction Grating. This grating has a blaze wavelength of 500 nm and a groove density of 300 lines/mm. These two parameters allow for the grating to be able to efficiently diffract light at the extreme wavelengths of our projected wavelength range. The typical efficiency curve for this grating is shown in Figure.





### e. Buzzer Selection

A summary of the specifications derived in the speaker section is presented in Table. All other values are not subject to rigid specification, though cost will be minimized where possible. Table shows several external drive piezo buzzers available from Jameco.

Parameter	Value
Frequency	4000 Hz
Sound Pressure	70 dB
Nominal Voltage	5 V

Table 11: Buzzer Specifications

Jameco Part#	Sound (dB)	Voltage (V)	Current (mA)	Diameter (mm)	Price
2098523	80	5	5	13.8	1.19
1956776	85	5	1.5	30	2.35
1874543	70	3	3	12.6	0.69
336276	80	5	2	13.5	0.75
2207081	85	3	5	30	1.09
138713	80	3	7	13.7	1.99

Table 12: Possible Buzzer Parts

From this list, the part labeled 336276 was selected for its low cost and low current consumption while having a high peak sound pressure.

The selected transistor is a 2N2222 family NPN BJT commonly used for switching and linear applications. This transistor was selected because it was readily available during

prototyping. During part research, nearly every identified transistor on the market was capable of fulfilling the requirements in Table. In this instance, cost and packaging are the only relevant characteristics. Table shows a few vendor specs for these transistors.

Parameter	Value
Operating Frequency	5 kHz
Power	10 mW
CE Voltage	5 V
CE current	10 mA

**Table 14: Q1 Specifications**

Product ID	Vendor	Length (mm)	Width (mm)	Height (mm)	I <sub>C</sub> max (mA)	CE Voltage	Power total (mW)	Frequency (MHz)
MMBT2222A	NXP	2.9	1.3	1	600	40	250	300
PMBT2222	NXP	2.9	1.3	1	600	30	250	250
PMST2222	NXP	2	1.25	0.95	600	30	200	250
NTE22	NTE	7.11	2.6	4.7	2	80	900	100

**Table 13: Transistor options**

The selected product most closely matches the NTE22, which will be used for estimate. The prices for this product cannot be affirmed without a better estimated production scope, but a small sales estimate of 0.13 cents will be used for the package in this project.

The fly back diode has similar minimal constraints to function correctly. Here a classic 1N4148 switching transistor is used. The requirements are posted in table Table. Table shows the selection from Jameco, though the prototype will likely use an already acquired part.

Parameter	Value
Reverse Voltage	5V
Power	1 mW

**Table 15: Speaker Diode Requirements**

Jameco Part#	Manufacturer	Max current (mA)	Reverse Voltage (V)	Forward Voltage (V)	Price (\$)
151247	Major Brands	10	100	1	0.08
36038	Major Brands	150	100	1	0.05
179215	Vishay	300	100	1	0.05

**Table 16: Speaker Diode Parts**

The selected diode most closely matches part 36038.

## f. LCD and Driver

The Kentec QVGA Display Booster Pack, one option for our LCD and seen in Figure below, is fully compatible with the development board we have chosen. This pack is originally designed to connect to the development board, but since the microcontroller will be placed on a PCB board, we will have to manually connect the booster pack. This process will not be complicated since there are data sheets that will tell us where to plug in each pin. TI offers many resources, such as the data sheets for both the microcontroller and booster pack. These resources will be able to tell us where each connection will be made, why they are made, and what their function to the LCD is. The LCD display given by this booster pack will be discussed further in the LCD section below. Other options are to find an LCD screen that meets our needs, and connect it through the SPI/I2C, similar to the booster pack. The main difference here is that this second option will give us a larger variety of LCD screens to choose from, and won't limit us to a certain size due to compatibility. The cost difference between the booster pack and other LCDs differs by resolution and size. The booster back costs \$24.99, while the other optional LCDs cost anywhere from \$20.00-\$50.00. This factor won't be the sole reason we choose any specific screen, but is something to keep note of since we don't want to spend a large amount of money on a screen.

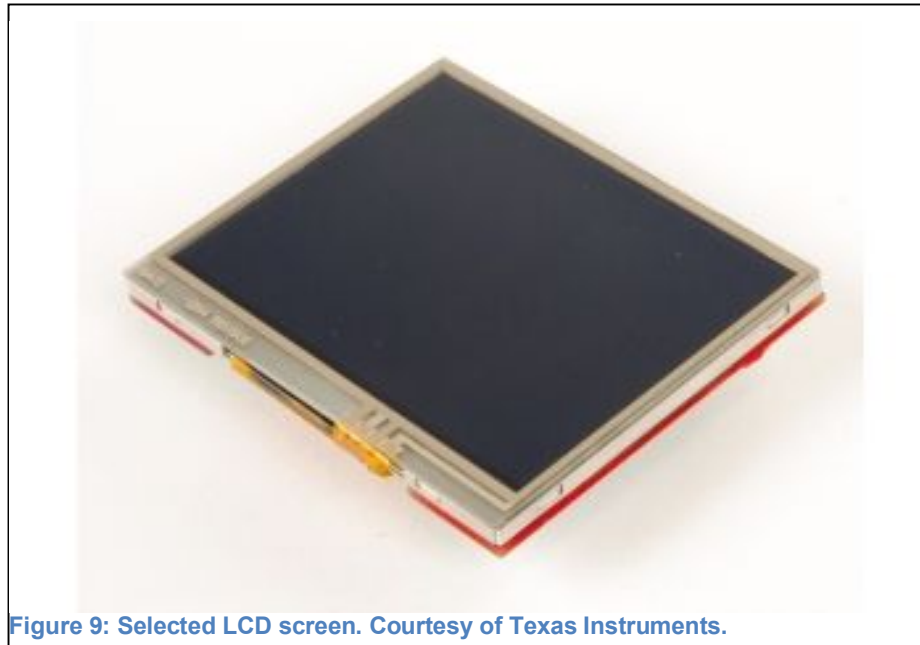


Figure 9: Selected LCD screen. Courtesy of Texas Instruments.



## g. Micro Control Units

The microcontroller unit is a key part of any electrical system. It allows the system to complete multiple tasks, and has many components to it. This device will be the center of the system, and will control the majority of other components in the system. It will do this by connecting to external hardware, and reading code loaded in either the device it's connecting to, or from code loaded onto it. For our particular system, we chose to use a MSP430F5528. The reason for choosing this controller is because all of us have experience with MSP430s from previous classes. This ensures that we can get right to work on it, and we don't need extra time to learn how it works. We are also provided resources for this component on campus, and through teachers who have helped us use this device. There are many choices of microcontrollers to choose from. Some of the more popular devices are Arduinos, and while there seems to be a good amount of support for these devices, it would be similar to using the MSP430. One of the main differences between these two devices is price. The Arduinos can cost anywhere from \$20.00-\$40.00, while the MSP430s cost from \$1.00-\$6.00. The catch here is that the MSP430s require a development board which can cost anywhere from \$10.00 - \$100.00. In order for us to save money here, we will be using the development boards from the TI lab on campus. Most microcontrollers have the same ports, but are usually structured differently or have a different number of ports. This means that most of the external devices needed for this project will be compatible with the MSP430, just as it is compatible with the Arduinos and other microcontrollers.

The ports needed for our microcontroller are the SPI, I2C, general pins, ADC, UART, and USB. These ports will be supporting the power to the light, the LCD, the buttons for the interface, and will receive information from the optics portion. One variable to consider for our microcontroller is the memory. There will be two types of memory used, non-volatile, and the RAM. The non-volatile memory is saved on the controller, while the RAM is lost once the system is turned off. This means we need a microcontroller that has enough memory to support our external components and to run the code stored on it. This is why we chose a controller with a large amount of memory, this way we will have enough in case we need it. One factor in choosing the microcontroller was the ADC channels. For our system we need at least one ADC 12 bit channel. ADC stands for analog to digital converter, and it represents data given to it and converts it to bits. A 12 bit ADC will be able to convert data will be able to represent 2 to the 12<sup>th</sup> power numbers. This will be useful for connecting to our optics portion of the system and CCD in order to read in the data. The majority of MSP430s viewed had a slope ADC. This just means they have 2 to the 1 power of numbers to be represented. This limited our choices, but thankfully, since there are so many MSP430 microcontrollers, we were able to find one that fit with all the other specifications that we needed.

Another variable to consider is the operating temperature and power. This has to be considered with the operating temperatures of the other external devices in order to keep the heat of the whole system at a minimum to avoid overheating. And another threat to the

system could be if the heat creates a problem with the box holding everything. The MSP430F5 series are all low power microcontrollers. This means that they use a low amount of power compared to other microcontrollers, and will produce less heat, and need less power from the power supply to run efficiently. All these factors are reasons why we chose MSP430F5 series. This series of MSP430s has a low operating temperature, has a USB port so we can easily connect devices with USB, and has enough SPI and I2C channels to connect our external devices. The biggest differences between the specific MSP430s within the 5 series are the ADC channels and the amounts of memory given to each microcontroller. This is why we chose the MSP430F5528, since it has enough memory and the correct ADC channels that we need.

## h. Power Supply

The final BoM for the selected PSU components is shown in Table.

Device	Description	Quantity	Cost (\$)
Diode	1N4007 rectifier	4	0.60
Heavy Duty 3 wire replacement	Male electric plug for power supply	1	5.35
Voltage Regulator	LD1117	1	0.16
Voltage Regulator	LM7905	1	0.64
Voltage Regulator	LM7812	1	0.64
Voltage Regulator	LM7912	1	0.64
Capacitor	10uF	6	0.01
Capacitor	100uF	4	0.01
Capacitor	470uF	1	0.01
Transformer	120V/12V, 1A Initial voltage conversion	1	17.90
Resistor	100k ohms	1	0.01
Resistor	180k ohms	1	0.01
<b>Total</b>			<b>25.98</b>

Table 17: Power Supply Part Selection

## i. Project Enclosure

The electrical enclosure in Figure below will be used to hold all the components of the spectrophotometer that is to be designed. The enclosure will provide enough room for the project. The dimensions of the enclosure are 11”x 8”x 3”. The project enclosure is made of fiber glass and other components which helps with it being able to be modified, in order to fit our needs.



The other enclosure shown in below is also needed for the photodiode array, CCD, and light source components. This enclosure is useful because of the reduction of light and protection from the rest of the components. The mirrors, light bulb, and cuvette must be securely mounted and enclosed to obtain best results. Everything must be in perfect alignment to pass light through the cuvette to provide good readings. The cuvette must have a stable slot in order to repeat testing.



In addition, a suggestion that could possibly be carried out if the group has enough time is to be able to construct our own project enclosure. The suggestion is to have the enclosure 3D printed. First, it will have to be designed on a software, making sure all dimensions are correct. After the sketch, the file will have to be converted to standard tessellation language, which is needed for 3D systems. The sketch is then copied to the computer, which operates the 3D printer. Now that the design is ready to print, adding various materials to the printer will be needed. Revise sketch and make sure dimensions and thickness of walls are correct. Now print and remove object very carefully to prevent and damages or deformations. Let the design sit for 30 minutes and the dip in water to remove powder or water-soluble material. The enclosure is now ready to be used for the project.

The table in Table below shows the specifications required for the project enclosures.

Enclosure Selection	Length (inches)	Width (inches)	Height (inches)
AI-EB1212	12	12	4
CU-1874-B	5.3	2.9	2

Table 18: Enclosure part selection table

## Final Enclosure:

The enclosure initially intended to be used was a 3D printed model of a case nearly the size of a shoe box. The dimensions of the design are as follows, length-12in, width-8in, and height-6 in. the reason to why the 3D printing was not achieved was due to the TI Lab was backed up on files that were also needing to be printed. Instead we decided to use a senior design partner's 3D printer to print the design, but unfortunate the file was too large to print for the printer. The price for the material needed to print the device was \$15.36. Instead we decided to purchase the enclosure, but the price ran close to \$60.00 for one nearly the same dimensions required and \$30.00 for one with dimensions that did not meet the needs for the design. Eventually settling for a tool box with dimensions close to that of the 3D enclosure for only \$6.88. The following picture shows the final product of our enclosure/design. The enclosure needed to be cut in order to place the LCD, and keypad on the top of the lid. A quarter of the lid was needed to be cut and attached a latch in order to open and close when used, due to the accessing of the sample that is being tested. On the right side of the inside of the enclosure is where the power supply with a switch attached to the DC side of the power supply, to ensure the power to be on or off with a flip of the switch is placed. As in the middle of the enclosure is where the switching MOSFET, that is used for the light source and PCB will be located. In the left bottom side of the enclosure was used for the optics material. The optics material needed to have an enclosure of its own due to it being sensitive to light. The optics consisted of two mirrors and a grating, which needed procession and accuracy for the alignment of the mirrors and grating. The light source was placed on the top left of the enclosure to correlate with the optics material.



Final Enclosure: The image above is the final design of our project.

## 5. Standards and Constraints

### a. Standards and Impact

#### i. Sample Cuvettes

A cuvette is the name for the standard sample container for any type of spectrophotometry instrument. The material, design, and cleanliness of a cuvette are important to achieve accurate measurements. The material types of a cuvette can be split into disposable and non-disposable. The wavelength range associated with each cuvette shows the range at which the transmittance is over 80%. Disposable cuvettes are made of plastic materials, and are used for fast measurements, at the cost of accuracy. They also are used to avoid washing procedures and any sample carryover. Disposable cuvettes absorb UV light, so they would not be suitable for this application.

Non-disposable cuvettes can be made from materials such as optical glass, quartz, and sapphire. Optical glass is the cheapest option, with a wavelength range from 340 nm to 2500 nm. Many applications of spectroscopy fall into this wavelength range, so this material is incredibly viable. However, it is not the ideal material for applications involving UV light. Quartz is the next option when it comes to cuvettes. This material can be used in the range of 190 nm to 3500 nm. Quartz cuvettes are a little pricier than optical glass cuvettes, but at least UV measurements are viable. Sapphire cuvettes are premium grade

cuvettes. They are damage resistant and difficult to scratch. The wavelength range spans from 250 nm to 5000 nm. For the application of UV-VIS spectrophotometry, a quartz cuvette would be the reasonable solution.

Non-disposable cuvettes will be considered in the following discussion on cleanliness and design. The sides where the light passes through the cuvette are called optical windows. Optical windows must be polished, flat, parallel, and incredibly clean. Any residue or lint on the windows can cause unwanted absorption. Inaccurate absorption measurements are to be avoided at all costs. It is important to clean the cuvette before and after each use. The Z height of the cuvette is the distance from the bottom of the cuvette to the center of the sample chamber. It can be thought of as the height at which the light hits the cuvette. The standard Z heights are 8.5 mm, 15 mm, and 20 mm. At the Z height is where the light path length of the cuvette is specified. The light path length is the distance between the inner surfaces of the optical windows. The common path length is 10 mm, but it can differ based on the application. The cuvette manufacturer will supply the light path length of the cuvette.

## ii. USB Communication

Two sources of this standard were identified.

[1] 1667-2015 - IEEE Standard for Discovery, Authentication, and Authorization in Host Attachments of Storage Devices

[2] IEEE Standard for a High-Performance Serial Bus

To understand more about how the USB actually works, we viewed the IEEE Standard for a High-Performance Serial Bus. This standard showed us the inner workings of serial buses, such as an USB. Although we are not constructing the USB itself, this standard shows us minimum requirements for how they are built, and what is needed from data a data transfer standpoint. From this standard, we can see that the minimum requirement for the data transfer needed from the cable is 98.304 Mbits/s. This means that our USB port should support a greater data bit transfer than that value [1.6.2, Standard Serial Bus]. The durability needed from our USB is also specified.

The minimum amount of connections between the USB and its port, before performance is hindered, is 1500 cycles [4.2.1.8, Standard Serial Bus]. The USB on our system will be more advanced than the minimum, and should be able to perform for more than double this amount of cycles. The testing done on these serial buses shows us the limits and how much stress our USB will be able to perform under. It will be able to withstand a voltage of around 500 Vdc, and can maintain performance under high humidity [Table 4-5, Standard Serial Bus]. We will not be putting our USB under such extreme conditions, but these tests show us the durability of USBs, and allow us to proceed without us having to test the USBs. We can safely assume now that our USBs will have no problem working under the conditions of a lab, and will not be disrupted by the amount of data we will be passing through it.

### iii. Power Supply and Outlet

Power supply standards are very important when designing a project. National or international agencies inspect equipment in order to pass legal regulations. Government laboratories are used to test the equipment in order to confirm passivity. The main objective of safety standards for power supplies in using electrical equipment is to protect against electric shock, injury and fire hazards. While building the project we need to meet the requirements from the associated standards organization or by a mark specifying passivity with local legislation, within a defined economic area.

The organizations responsible for electrical safety standards are The International Electro-Technical Commission (IEC) and the associated International Organization for Standardization (ISO). In order to identify a standard, it will obtain a prefix following with the standards number such as IEC60950. IEC60950 is associated with the 2005-2013 standards with the first and second amendments. The standard applies to battery-powered, office/lab equipment information technology (IT) or other equipment with a rated voltage not exceeding 600 V. to help protect people or property from hazards as high temperatures, fire, electric shock, and mechanical unsteadiness.

The power supply standards are divided into three sections each with different regulations. The first standard, Class I protects for electric shock through a basic insulation and protective grounding. In case of a hazards voltage in basic insulation failure all conductive parts should be connected to a protective conductor. For the Class I equipment protective conductor must have a green or yellow covering, with no switch or fuse. Grounded parts and the grounded end must not exceed  $0.1\Omega$ . This is obtained by testing current 1.5 times the current capacity of the hazards voltage circuit, at the point the basic insulation fails and the grounded section is live. The test for voltage must not exceed 12 V and the testing for current (ac or dc) must not exceed 25 A. Grounded Leakage Current should not surpass 0.75 mA for hands on operations for, and other Class I devices should not pass 3.5 mA.

The Class II does not need any grounding because; it provides protection by using double or reinforced insulation. Grounded Leakage Current for Class II equipment should not exceed 0.25 mA. Testing for Class II devices entails conductive metal foil to be attached to an area not surpassing 10 cm x 20 cm on nearby non-conductive parts. Tests are made between non-conductive and conductive parts; also all tests should be measured at the highest voltage supply. While, the Class III equipment functions from a SELV (Safety Extra Low Voltage) supply circuit. This means that it naturally protects against electric shock. Not needing to worry about insulation or high hazardous voltages.

A hazardous voltage is considered dangerous if the voltage exceeds 42.2 V ac peak or 60 V dc without a limited current circuit. For an extra-low voltage (ELV), circuit will need separation and basic insulation to prevent hazardous voltage if the voltage is not exceeding 42.4 V ac peak or 60 V dc. For primary Safety Extra-Low Voltage (SELV) circuits must be separated from hazardous voltages, by two levels of protection, which may be protected with basic insulation combined with a conductive barrier or double insulated. In addition,

secondary SELV are considered safe for operator access. In regards of being safe, these circuits do not require extensive safety testing. The flammability standard necessitates that the design avoids high temperatures and divides flammable materials from high temperature parts. Passivity can be accomplished by using V-2 rated insulating and printed board materials. Also, be sure there is enough spacing between high temperature components and plastic parts.

There are many different kinds of insulation to shield live components from hazardous voltages. A few of these insulations are functional insulation, which does not protect against shock but it is still useful for the equipment to operate. Another method of insulating is basic insulation, which provides functionality and protection against electric shock. While for supplementary insulation is a separate insulation applied to the basic insulation, which aids protection in case if the basic insulation fails. However, double and reinforced insulation are similar. They both have the same degree of protection against electric shock, but double insulation contains basic supplementary insulation, though reinforced is a single insulation system.

For basic electrical safety and avoiding shock hazard, we will need the third edition of power supply standards. This edition further divides resources of protection into patient and operator protection, as shown in the Table

Classification	Insulation	Isolation	Creepage	Solution
One MOOP	Basic	1500 Vac	2.5 mm	IEC60950-1 Rated Power Supply
Two MOOP	Double	3000 Vac	5.0 mm	IEC60601-1 Rated Power Supply
One MOPP	Basic	1500 Vac	4.0 mm	IEC60601-1 Rated Power Supply
Two MOPP	Double	4000 Vac	8.0 mm	IEC60601-1 Rated Power Supply plus an IEC60601-1 Rated Dc-Dc Converter

*3<sup>rd</sup> Edition Requirements by Classification*

**Table 19: Third Edition Power Supply Standards. Used with permission from [rhornberger@cui.com](mailto:rhornberger@cui.com).**

## b. Economic Considerations

As mentioned in the introduction to this report, many research labs are popping up and requesting funding from the same centralized government sources. The major repercussion of this is that most of these labs receive less funding than they would like. Every device they can acquire for less capital than expected is more research time and higher data quality that can be acquired for no additional grant consideration.



This device was designed to provide adequate spectral resolution for a number of common midpoint tests in the research environment. If a report requires extremely precise results to maintain credibility, usually a much higher price point device will be considered to produce that quality. Even when these devices are purchased, they are often unfit for common benchtop application and not preferred for casual data integrity verification, such as confirming the success of a purification scheme or assessing a colony maturity. For these purposes, many labs are likely to purchase a cheap desktop device to produce this data, and the conveniences of this product while not exceeding existing price ranges puts it in a good place for consumption.

During the market analysis, a number of common specification trends were established and reflected in our pitched product specs. One of the most common is relatively large spectral bandwidth, sometimes ranging into 5nm for a single wavelength scan. Another very common trend was the complete lack of user interface on the device, existing entirely as a peripheral for a larger computer system. While computers are extremely common in a lab environment, tying up the available peripheral slots with a simple device and hogging the computers to run very quick scans is undesirable. A standalone device was requested from both individuals during customer interviews, and that makes this a preferable device.

The prototype is expected to be constructed with an inexpensive car headlight Xenon bulb to facilitate multiple prototyping cycles and part replacement, if the produced spectral range is adequate it turns out to be an excellent economic opportunity. Light sources are the only major mechanical component to the system and the most likely to fail, and one professor mentioned that a failed bulb cost around \$500 to replace in one similar device. This cost is prohibitive and renders the original investment useless. Having a product with cheap, easily replaceable components makes for a strong investment that, if found to be adequate in specification, will likely be a beloved product by many struggling lab startups.

Overall the main driving force behind this application design is to fit into the economic profile of the current molecular biology community. The generated bill-of-materials appears to reflect a good price point, and the user-interface improvements such as pre-generated graphs and figures for reports make this an attractive fit for labs struggling to collect decent quality equipment.

### c. Health and Safety

Without a properly grounded electrical system, all components in the design are at a very high risk from being destroyed. In the foulest situation when an overload of power occurs it can cause a fire, jeopardizing not only extensive property and data loss but physical injury also. If you detect a wire insulation, deteriorating and rubbing against a metal frame do not touch the object. If the wire were to be touched contact between the metal current conductor and the metal frame could make it live with electricity. If grabbed the current will surge through one's body and potentially being very harmful or even death.

Another way to prevent from deteriorate occurring is to inspect, maintain, and organize repairs of wires wear there is a lot of scraping or entering a pipe. Safety is first priority, ensuring electrical system is well grounded, and the ground circuit is electrically continuous. This could be very harmful to humans, not only could they lose their life but the can lose a finger or a limb. Everything must be really well ground and protected from unauthorized use.

Many customers will use quartz cuvettes in this device, which has a tendency to break while being handled. This leads to large amounts of broken glass and sample material in the compartment, which can be a large risk if improperly handled. The sample compartment must be watertight and have a minimum of corners for the material to catch on and create unsafe areas.

Our light source does smit light in the ultraviolet (UV) part of the electromagnetic spectrum. UV radiation has been documented to cause problems to the human body. Even though the xenon lamp is not a dedicated UV source or incredibly high in intensity, it is still important to note the hazardous effects of UV light. UV light is an invisible form of radiation. Since UV light cannot be detected by the human eye, precautions need to be taken to reduce hazardous effects. One of the most important precautions to take is to have protective equipment, such as safety glasses and lab coats. The human eye is sensitive to UV radiation. A short exposure of UV radiation can cause temporary effects to your eyes, such as blurry vision and eye watering. Long exposure to UV radiation can lead to permanent vision loss, pain, and distaste for bright light. Long exposure of UV radiation to the skin can cause skin aging, freckling, wrinkling, permanent sun spots, and even cancer. Another precaution to take is to inform and educate individuals working near the equipment about the effects of UV radiation and how the equipment can create UV rays. People must be aware of the devices that create UV rays in order to prevent exposure.

#### d. Manufacturability

Over many decades most all manufacturers of electronic devices say smaller is better. Design engineers from mobile phones, personal computers, to larger residential and appliances continuously struggle with the market driven requirement for smaller packages. Solid-state relays are commonly used in many commercial and industrial applications, medical equipment, renewable energy systems, spectrophotometer, and many more. Therefore, solid-state devices are main goals for engineers that are trying to reduce the size of their system. An example of a solid-state relay in use is medical equipment such as a portable dialysis machine. However, engineers must be subtle to the mechanical and electrical limitations of the solid-state relay when restructuring their systems. If the total power requirements of the new design is not reduced in amount to the reduction in overall system size. Then the solid-state relays may be endangered to more strenuous operating conditions.

The enclosure printing for the project will every simple but yet efficient. It will consist of a sketch on eagle cad and then printing on a 3D printer. Laser trimming resistors are intended to be used in all applications requiring an initial calibration, such as, sensors, voltage dividers, and spectrophotometers. This technique allows parametric values of the

circuitry to be realized by edging the resistance value. Which can be used as replacements for trimming potentiometers or manually selected precision fixed resistors to calibrate an electronic device. Some advantages are trimming of the output frequency of an oscillator instead of the resistance value, while other benefits include cost reductions, and PCB space savings. I believe the spectrophotometer will be a top selling item of its kind. The spectrophotometer has all these cheap and easy build constructing components to help with calibration, efficiency, and even providing a graphical representation of sample tests.

This device is ideal for current manufacturing methods because the precision of the electronics is not the limiting factor in the accuracy of the device. Large tolerance margins mean no laser trimming will even be necessary to produce a working device. The ICs used in this product can purchased in surface-mount packages to improve machine soldering and productivity while ensuring a small footprint.

There may be a point in the time of operation of the device in which the optics may need to be cleaned. The usual culprit is dust and other artifacts. There may be instances where the optics are contaminated by other means, such as human skin. Optical devices must be maintained in sensitive instrumentation, or the results from tests will be inaccurate or even indecipherable. It is also important to know the correct way to handle optics with a human hand. Gloves must be worn when handling important optical devices, like the mirrors and diffraction grating in the spectrophotometer. If dust or similar particles are present on the optic, a small blast of compressed air should be sufficient to eliminate the particles.

In order to clean harsher contaminants, like fingerprints, more detailed procedures are required. A spectrophotometric-grade solvent and lens tissue are the usual materials needed to eliminate most contaminants. One popular solvent to use is a mix of 60% acetone and 40% methanol. The edges of the optic should be cleaned first. Cleaning the edges first prevents more particles from adhering to the face of the optic. Wiping slowly will sufficiently clean the edges of the optic. For metallic-coated mirrors and diffraction gratings, the brush technique for cleaning. Fold a piece of lens tissue until it is as wide as the optic to be cleaned. Grip the tissue with a pair of tweezers and wet it with the solvent you are using. Shake off excess liquid. Place the tissue on the surface and softly wipe across the surface. Repeat the wipe two or three times to fully remove any impurities. If one is too harsh in the cleaning process, the surface of the optic can be damaged beyond repair. Diffraction gratings are a pain to clean. This is because the groove structure can be damaged by any touch to it. So, a decrease in the quality of the grating is expected after cleaning and handling.

## e. Maintenance

One of the most disappointing design characteristics is a system with poor access to removable parts or even no replacement parts supported at all. The audience that is expected to use this device contains scholarly individuals and technical personnel, so a few

conveniences can be included to improve the product life of this system in the hands of qualified technical individuals.

In any electrical system, the components that are most likely to fail first usually contain the mechanical elements. The most obvious point of failure for this system is the light source, which is subject to constant transient stress in flashing for every scan. A simple consideration for the maintainability of this device is a standard light fixture and an easily accessed compartment to change the bulb. While these bulbs can be expensive, it is often less expensive than replacing the entire device, and customers are more likely to think highly of the product line if they don't have to throw it out every time a bulb dies.

Another common failure point is the USB socket, which is expected to experience a high level of strain and stress during normal operation due to users literally jamming USB flash drives into the port. This is a tougher element to support, as the connections of the socket must be board level, but leaving large traces can allow for after-market solutions as required. This is a secondary level of maintainability, but small efforts can pay off in the long run.

The device enclosure should be sturdy enough to prevent accidental mishandling from ruining the case integrity and possibly damaging the electronics and optics. In a production model, the case would likely be an ergonomic and minimal size package with good trussing to absorb the impact of general wear-and-tear. This project uses a simple enclosure to protect and deliver the system components, but care will be taken to ensure it is strong enough to withstand construction as well.

A standard mains outlet fuse is included on both the isolating input and the transmission line to the motherboard to protect the components in the case of overdraw. Fuses are generally inferior to circuit breakers in most situations, but these fuses are not expected to fail unless the system is damaged in some way, usually irreparably. Regardless, using good standardized values for these fuses does make the system more user-friendly and extends the product life in a real environment.

Many production models use internal trimming of some kind on the sensor inputs to prevent long-term voltage drift at the terminals. This was not deemed necessary for this prototype to meet the requirements, but internal trimming or even system calibration is a beloved feature in technician work. A maintenance mode is usually included in sophisticated devices that can calibrate these values automatically, especially after repairing a hardware failure of any kind. Maintenance modes also generally contain robust error reporting and diagnostics to assist repairs.

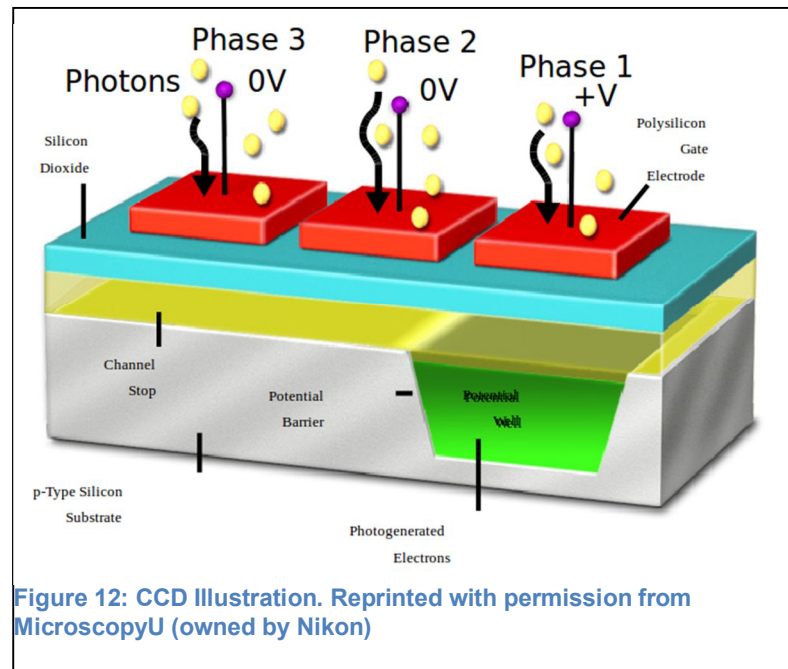
## 6. Subsystem Design

### a. Sensor Processing

The signal captured by the CCD must be accurately conveyed to the microcontroller for analysis and presentation of the spectral data. This section addresses the method of action and electrical considerations for processing the signal from the sensor.

### Sensor Method of Action

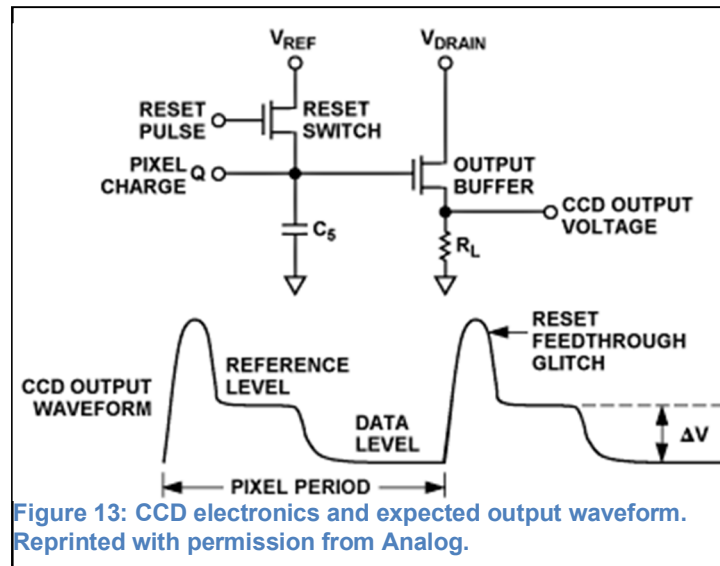
The charge-coupled device (CCD) is a dense matrix of MOS capacitors formed in a silicon substrate and surrounded by a two or three clock plate structure. Incident photons strike the semiconductor material and form an electron-hole pair when the bandgap energy of the substrate is exceeded<sup>6</sup>. Under normal conditions this pair simply recombines to return to neutral, but when a positive charge is applied to the set of plates opposite the MOS insulator, the electron is attracted away from the depletion region and stays fixed to the insulator, forming a potential well that stores a charge. This charge is exactly proportional to the incident light that strikes the pixel, reflecting the intensity of light at that point. Figure Presented in Nikon's Microscopy U's introduction to CCDs shows an illustration representing this structure.



To read this image, the charge of the pixel attached to the output pin is sent to the output buffer. The voltage applied to the first set of plates is then transferred to the second set, causing the potential wells to shift and follow the positively charged plate. This is repeated on a clock cycle to move the charge of each pixel to the output buffer. The result is a series of voltage outputs correlating to each pixel of the CCD, which can be reconstructed in software to form the image that was presented during integration time. For this application, that image represents the spectrum of light transmittance through the sample.

## Characterizing the Input

The first step to any processing block is understanding the expected signal characteristics both before and after the circuit. Once the CCD has been properly driven, it starts delivering a very distinct time-domain waveform such as in Figure. Typical on-board electronics are also shown in this figure.



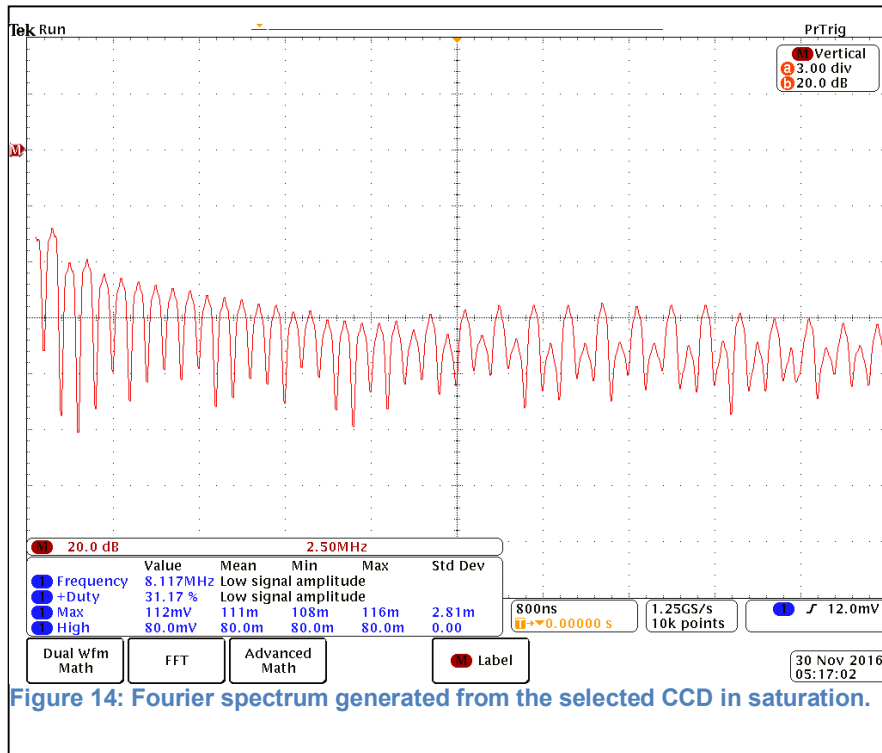
The waveform contains 3 distinct elements per data cycle, as marked on the figure. The first large peak is known as the “reset feedthrough glitch”. This is caused by the brief exposure to the voltage rail when the reset switch is opened to clear the integration capacitor. No interesting data is contained in this peak, but it must be considered when selecting the settling time of the amplifiers. The relatively high peak is usually too brief for

the slew time of the op-amps, especially when following a dark pixel. Measurements must not occur too closely to the peak or the level may not have settled correctly.

The first flat level is called the “reference level” and is the voltage of the integration capacitor before collecting any pixel charge. Ideally this level would always be the same, but in reality it is subject to several types of unwanted noise. This level is useful in noise reduction techniques, since the time between this level and the data is often less than 1 microsecond. A technique called “correlated double sampling” is often used to compare this level to the data and reduce any high-speed noise.

The last level, also relatively flat, is the “data level”. This final level is a steady output anywhere in the reported dynamic range of the sensor that gives the charge collected for the pixel being read. The goal of this sensor processor is to retrieve this data level with a minimum amount of noise. A full signal noise analysis will be included in a later section.

Unfortunately inspecting the signal in the frequency domain does not reveal much of interest to the design. The lengthy variable data level causes the spectrum to contain many overlapping harmonics that saturate the upper part of the spectrum. The natural response to this is to include a high-pass filter with a cutoff below the data rate, which acts as the fundamental frequency. Unfortunately most sources of EMI in the expected operation environment are above this data rate, but the effects of double sampling will have to suffice for removing high frequency products. The spectrum is displayed in Figure.



The data rate for the sensor is controlled by the clock rate delivered to it. For the selected sensor, the readout can be anywhere between 0.2 and 1 MHz. Any of these speeds will complete a full sensor scan in less than 20 ms, which is well below our desired scan speed before processing. Slower speeds allow for less expensive electronics and even new circuit topologies, but also causes excessive buildup of dark current which reduces the sensitivity of the system. For this project, an initial data rate of 0.5MHz was selected for initial testing. If this rate introduces errors due to settling time or bandwidth, we will consider lowering this data rate.

Table shows a summary of the key characteristics of this signal as a general input, which assists in selecting components to process it.

Parameter	Min	Max
Data Rate	0.2 MHz	0.8 MHz
Signal Range	0.4 V	0.6 V
DC offset	1.5V	3.5V

**Table 20: CCD Output Characteristics**

## Noise Sources

Several forms of noise manifest between the sensor and the processor. Some are intrinsic to the elements and cannot be avoided, while some are part of the front-end electronics and can be considered during the design. The ratio of the dynamic range of the system to the sum of all noise sources gives the system signal-to-noise ratio, an important figure of merit for any signal processing circuit. Signal-to-Noise (SNR) is often expressed as shown in Equation. In this equation, S is the signal and N shows the noise voltages that are added in quadrature.

$$SNR = \frac{S}{\sqrt{\sum(N^2)}}$$

**Equation 9: SNR Calculation**

Since noise sources are generated as the sum of squares, one simplifying assumption is ignore sources that are more than 3 times less than another source. For example, the op-amp input voltage noise only has to be lower than the read noise by a factor of 3 to have



negligible overall effect on the system SNR. This rule-of-thumb will serve as the governing factor in the design process.

## Photon Noise

This form of noise is a type of shot noise and originates from the statistical distribution of photons in the incident light flux on the CCD pixels. The variation in photon arrival within the incident beam is governed by physics, where a poisson distribution relates the actual arrival time of an individual photon with its real arrival time. This is a natural phenomenon, and cannot be reduced via electronic design, but maximizing the number of photons collected during a reading reduces its impact. The integration time of the sensor can be selected to always generate the full dynamic range for the selected wavelength and keep this noise from increasing, but it is likely to be the dominant noise source in the system. Equation shows the expected relationship between the signal and this noise. S is the signal magnitude, implying a photon noise magnitude of 100 electrons for a 10,000 electron signal.

$$N_{\text{photon}} = \sqrt{S}$$

Equation 10: Photon shot noise equation

## Dark Noise

Inherent to every light-sensitive semiconductor device is a phenomenon known as dark current, where some small current flows though the semiconductor without any incident light present. This current is a figure of merit in selecting a light sensor. The current causes some electron buildup in each pixel during the integration period, which will eventually saturate every pixel and prevent data from being collected. The key methods to control this noise source include clearing the integration gate between each reading and keeping integration time at a minimum for the desired reading. The noise is expected to follow the model in Equation.  $I_D$  is the dark current flowing through the CCD, and  $t_i$  is the integration time during the reading. Shorter integration times reduce this noise figure.

$$N_{\text{dark}} = \sqrt{I_D * t_i}$$

Equation 11: Dark Noise Equation

## CCD Read Noise

This is another figure of merit in selecting a light detector, and refers to the noise introduced by the on-chip electronics of the sensor itself. This can include the integration capacitor, the output buffer (usually a source follower), and the integration clear gate. These components are generally reported as one figure. For an ideal CCD processing system, this figure sets the SNR for the system. This noise is often Gaussian in nature, and can be reduced by averaging several exposures together. For this application, several pixels relating to a single wavelength of light can be averaged together to reduce the effects of this noise. This average will still correspond to the transmittance of light at the desired

wavelength, though the spacial resolution will be reduced. This process is known as binning. The selected sensor reports a dark signal voltage of 2 mV, but this figure most closely matches this read noise. For this project, any source contributing less than 1 mV of noise can be largely ignored due to this figure.

## Op-Amp Input Voltage Noise

After the signal is output from the CCD and clamped to appropriate processing levels, the output reaches an op-amp preamplifier that hosts several important noise sources. Ideally, this pre-amp will produce the only relevant electronic noise for the off-chip processing, locking in the SNR as soon as it reaches this amplifier. The first relevant op-amp noise source is caused by the voltage offset at the input and is bandwidth dependent. This figure is reported on the datasheet for any purchased op-amp and is measured in volt per root hertz. Modern op-amps tend to have such low input voltage noise that it does not approach the existing noise sources, but it can still be considered.

## Input Current Noise

Some op-amps have a relevant current flow into the input terminals through the input impedance. If this current flows through a large resistance, such as the input resistor of most op-amp configurations, it can cause a voltage noise. For the popular JFET op-amps available from most manufacturers, this input current is almost non-existent in the range of Pico amps per root hertz and can largely be ignored.

## Johnson-Nyquist Noise

Intrinsic to any passive resistive elements in a circuit is a phenomenon known as thermal Johnson-Nyquist noise, which takes the form shown in Equation. Here,  $k$  is the Boltzmann constant,  $T$  is the temperature of the element in Kelvins,  $B$  is the bandwidth of the reading, and  $R$  is the resistance value.

This noise is often modeled as a voltage source in series with every resistor in a circuit. There is no method to reducing the Johnson noise associated with a resistance value, but using smaller valued resistors can reduce the impact of this noise. Resistor choice in an op-amp circuit is limited on the high end by this noise phenomenon, while the low end is limited by power consumption.

$$N_{johnson} = \sqrt{4kTBR}$$

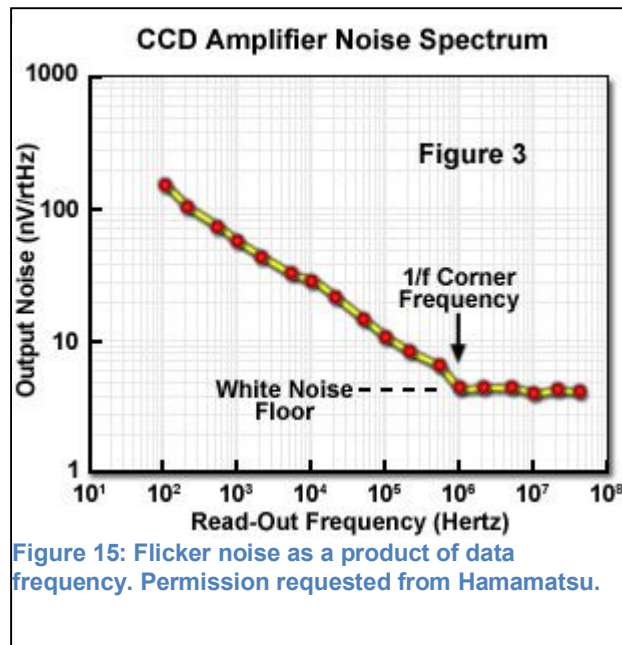
Equation 12: Johnson-Nyquist Resistor Noise

## Electromagnetic Interference Noise

A source that is difficult to quantify without several situational studies, this noise is caused by atmospheric high-frequency waves produced by other instruments in the same environment. These sources can penetrate the device enclosure and cause offsets in the signal that are not expected. Filtering out this noise directly is very tricky, and requires precise understanding of expected interference frequencies and carefully crafted notch filters. The only major source considered in this device will be mains voltage noise, which manifests as a common 60 Hz hum. Many methods are used to reduce other sources in commercial products such as heavy insulation, short trace lengths, and insulated enclosures. These factors will be kept in mind for the prototype construction.

## Flicker Noise

Sometimes referred to as  $1/f$  noise, this is a source characterized by semiconductor device states. The injections of state changes at certain frequencies cause this noise profile with a magnitude inversely proportional to the frequency of injection. This is also known as a “pink noise” in contrast to the white noise that models other sources. For lower readout speeds like those used in this project, this noise tends to be dominant. It is generally represented as a form of Johnson noise with reference to the output resistance of the sensor. Figure shows a typical flicker noise corner frequency analysis.



## Quantization Noise

This noise is introduced by the ADC and causes any voltage levels less than the range expressed by the LSB to be floored into a digital value. The ADC on the chosen microprocessor has 12 bits of accuracy over a 3.3 V range, which corresponds to 0.8mV of granularity in readings. This means that any circuit improvements that provide less than 0.8mV of resolution do not affect the final samples. Quantization dynamic range for this device is shown in Equation.

$$SNQR = 20\log(2^N) = 6.02N + 1.763 = 74dB$$

Equation 13: SNR due to ADC quantization noise for 12 bit ADC.

## Expected SNR Limits

Comparing the noise sources in the previous sections, the limiting factors in SNR include the ADC quantization noise limiting the real SNR to 74 dB and the CCD read noise itself. Due to these conditions, the Johnson noise and op-amp voltage noise are unlikely to limit the noise floor of the design with the selected components. Dark current noise is expected to be handled by creating a dark spectrum internally and removing it from the sample spectrum. Elements like flicker noise, shot noise (for large integration times and binned data), and input current noise are not significant enough to affect this application and can largely be ignored.

Johnson noise and op-amp noise will still be considered in the event that a better sensor is selected for this project. The low price difference between lower speed op-amps makes this a simple consideration to justify, and low resistor values in this application do not dissipate a large amount of power due to the low voltage.

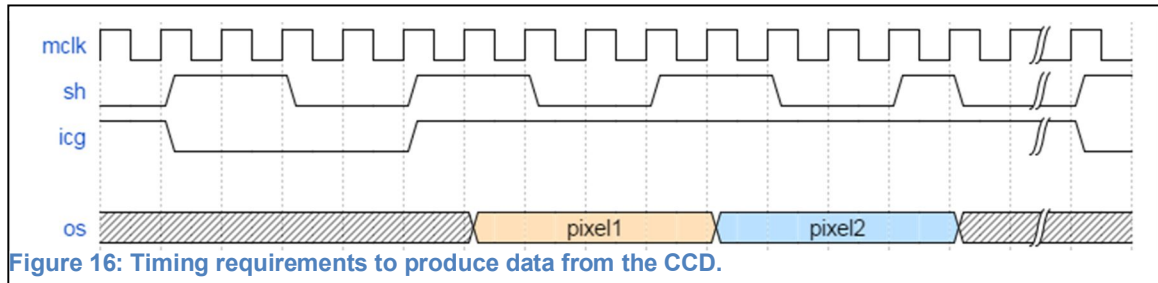
## Driving the Sensor

A typical modern CCD takes the form of a fully integrated semiconductor circuit and requires a minimum of external control to produce the image output. The selected sensor requires a master clock frequency, which can be easily produced by the clock generator of the MCU. Typical clock frequency values suggest a 2 MHz master clock for typical data rates, which corresponds to a data rate of 0.5 MHz due to the four cycle pixel output. The sensor datasheet recommends a minimum master clock rate of 0.8 MHz, which will be considered if the MCU ADC is not fast enough to capture the pixel data clearly.

An integration clear gate is used to clear any stored charge before collecting an image for readout, and can be connected directly to the MCU at TTL levels. This input will be set normally high to prevent charge buildup in the sensor and reduce power consumption when not required. The shift input controls the integration time of the CCD and can be calibrated to match the pulse from the light source without saturating any pixels.

The digital and analog voltage supplies are equal in magnitude and require similar rail filtering for best results, so these input will simply be connected to the 5V rail from the power supply. A small filter capacitor will be shunted to the voltage inputs to smooth out rail fluctuations caused by noise or large immediate charge consumption from the sensor.

Figure shows the timing diagram that will be implemented in the firmware to drive this sensor.



One interesting characteristic of this timing diagram is the non-typical shifting of the shutter input twice per pixel instead of once as listed in the data sheet. Projects that have been completed in the past that used this sensor decided that this timing scheme produced better video as the reset of the integrating capacitor actually occurred on the leading edge, not every edge.

The most important timing factor is the shifting of the integration clear input and the shutter gate nearly simultaneously. The data sheet lists a value of no more than 1 us difference in these two inputs changing. This is easily accomplished by tying all inputs to the clock frequency of the CCD itself. Pixels are always four clock cycles long, and the gates can easily be shifted in multiples of this frequency on the MCU.

The integration clear must be low for at least 1 us to properly clear the pixels on the CCD. Larger values do not cause erroneous measurements according to the data sheet, so an entire pixel cycle will be used here for between 2 and 5 us, which is considered typical for this application.

## Possible Hardware Solutions

A nearly endless quantity of sensor processing systems for CCD devices have been developed and studied, especially as the cost and accuracy of semiconductor manufacturing has improved. Many of the major designs are focused on video processing or astronomical imaging using high-speed and multi-port CCDs. Fortunately, this is not a high-speed or high-voltage application, and some noise correction can be done by averaging consecutive frames and adjacent pixels without sacrificing scan quality.

All solutions contained a few common elements, but differ in the noise correction method. The common elements include a DC restoring system, which is realized in this circuit using a simple AC coupling network. After that, the maximum and minimum voltages produced from the sensor must be matched to the range of the analog-to-digital converter of the MCU. These two steps are bandwidth-limiting and usually set the noise floor of the signal chain, as any future noise introduced will compete against a much larger signal magnitude after amplification. This hardware pair could be all that is required for lower data rates, along with a final dynamic range matching stage, but a hardware solution can also improve the noise of this signal. Figure shows a designed system using only these components, which is also the network that will be used in the case that digital solutions are desired instead of hardware.

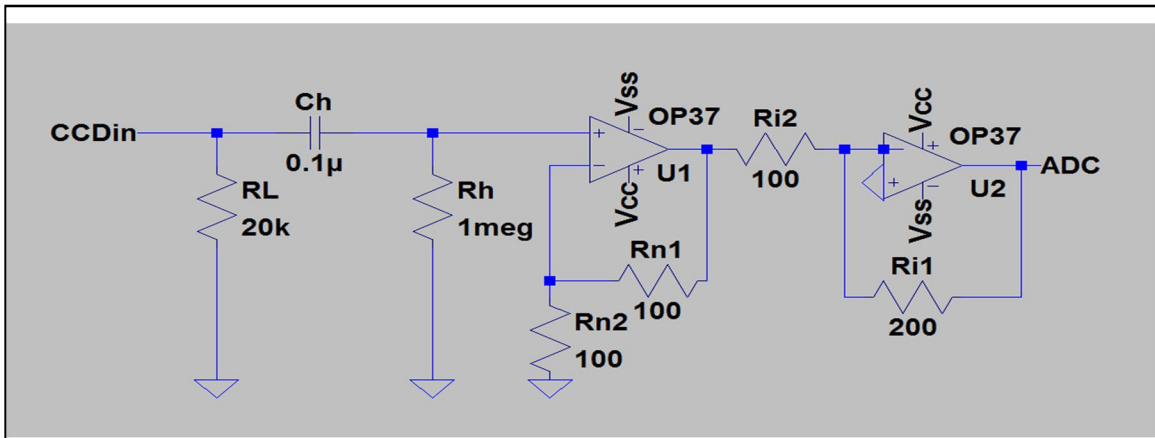
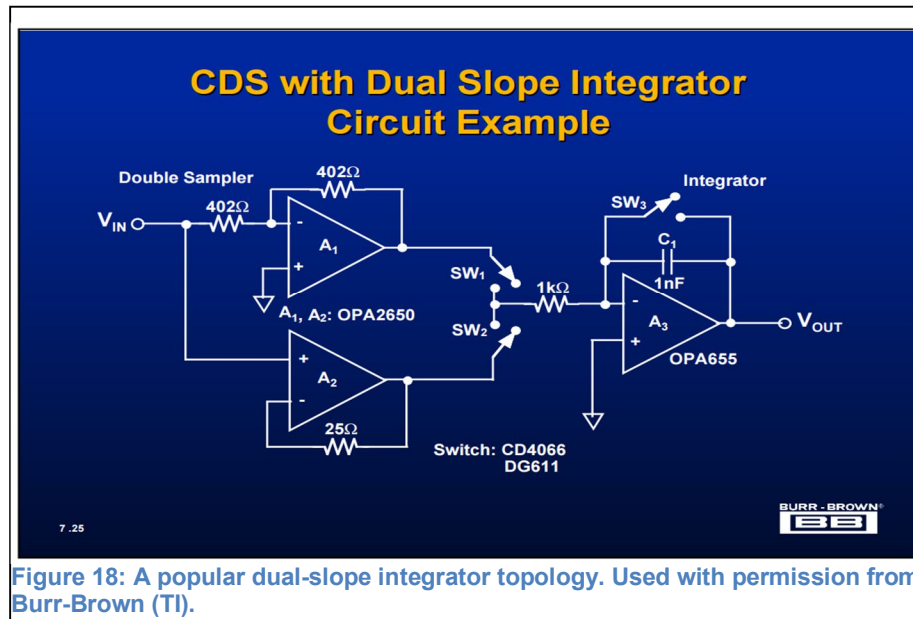


Figure 17: Designed clamp and preamplifier circuit. Suitable for DCDS without extra hardware.

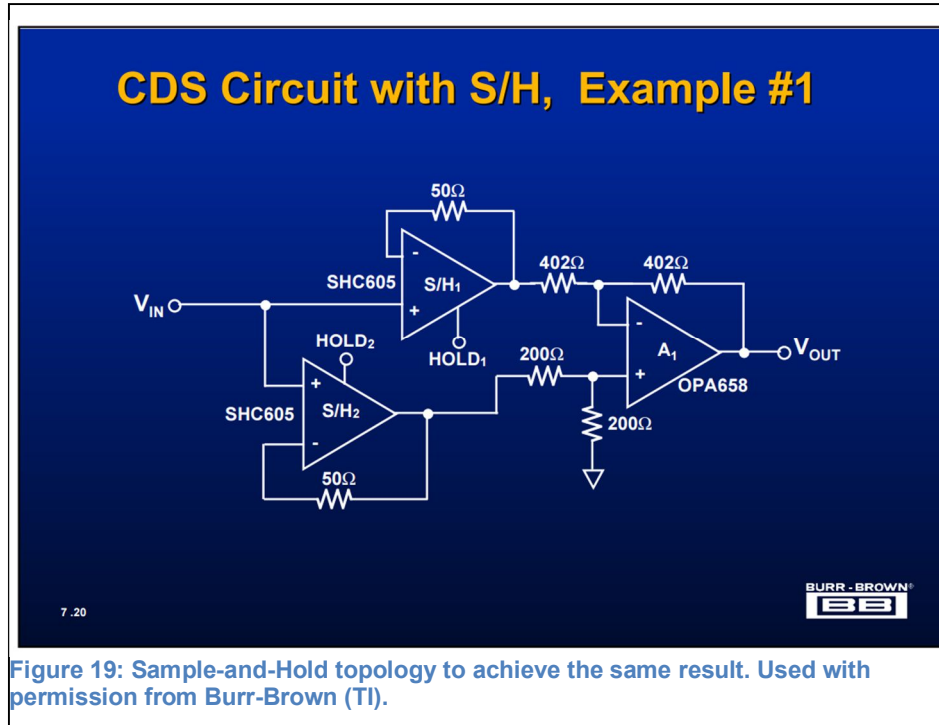
The noise correction method in question is known as correlated double sampling (CDS). Because of the relatively high speed of the pixel rate, most sources of noise are almost identical between the reference level and the data level of each cycle. Extracting the difference between these two levels gives the real data level with much less noise than simply reading the data level alone. Traditionally this task is completed with a hardware solution, though recent efforts have favored software correlation using high-speed sampling. The resulting waveform of an ideal double sampling system is a stream of virtually noiseless steps that can be sampled by an ADC at the data rate, with no need for the Nyquist frequency of any harmonics.

Three major topologies were explored for this system, each with different stress points and benefits. There are also fully integrated solutions that will be addressed, though for this project they will be considered on a secondary basis. At the time of writing, the selected method is the dual slope integrator, though future studies will potentially focus on the DCDS with a lower data rate and a high sampling speed to allow for digital signal processing and improved data integrity with less hardware.

The first, and most popular method in older systems, is the dual slope integrator, as demonstrated in Figure. This system inverts and compares the data level with the reference level using an integrator. The pixel cycle begins with closing the reference switch and charging the integrator capacitor to the reference level from the output of the non-inverting buffer. This switch is opened, allowing the capacitor to settle. As the data level is reached from the CCD, the data switch is closed and the inverting buffer subtracts some of the charge contained on the integrator. The data switch is then opened, and the ADC reads a sample from the charge on the integrator, which now corresponds to the calibrated data level. After the sample is taken, the reset switch is closed, shorting the capacitor and resetting the charge for the next pixel cycle.



The second method is a more traditional sampler with two sample-and-hold amplifier that feed a difference amplifier. One example is shown in Figure. Two sample-and-hold amplifiers store the voltage level at the reference and data levels with no inversion. The sampled values are then compared in the difference amplifier to produce the calibrated value.



The final system is a digital sampler that uses no hardware to compare the reference and data levels. This method requires a very fast ADC to sample the output waveform at the Nyquist rate of twice the highest desired harmonic of the waveform. Since the data waveform approximates a square wave, there is a large number of harmonics that can be very difficult to accommodate. One paper suggests sampling anywhere between 20-50 times the data rate is ideal to produce a consistent noise floor and reconstruct the reference levels. There is no figure for this method, since the hardware is simply the clamp and amplifier used in the other methods.

One major observation is that all of these methods are well-defined enough that many vendors produce a fully integrated signal processing chain, containing all clamping, range matching, and digital conversion steps in one package. Surprisingly this turned out to be the cheapest method, and has several beneficial points that make it the most attractive option. This will be used as a last resort for this project if none of the designed circuits are acceptable.

Table lists a few of identified characteristics of each method. As stated, prototypes will begin with the dual slope method and proceed to the digital or integrated options if required.



Dual-Slope Integrator	Sample-and-Hold	Digital
1 capacitor	2 capacitors	none
Highest theoretical SNR	Good SNR improvement	SNR limited by sample rate
All discrete	Can use monolithic S/H	No additional hardware
Not flexible on PCB	Somewhat flexible on PCB	Flexible after PCB

**Table 21: Comparison of different CDS topologies**

## DC Removal

The reference level of the data period is usually a DC signal near the top of the voltage rail. This high reference imparts constant power into the processor, challenges the rails of the amplifiers, and convolutes the design of the double sampler. The most efficient solution to these problems is a simple AC coupling capacitor, which charges to the DC level of the input and resets the reference level of the circuit to ground.

Research into this system revealed a technique known as “DC Restoration”, which involves a large feedback circuit including several op-amps and analog switches to set the DC level of the input to a specified reference voltage. This system is usually found in high-speed camera applications as a method of controlling the black level of a captured image. For this project, DC restoration was not deemed necessary, as it was not featured in other spectrometers and higher resolution devices.

The principle constraint on the coupling capacitor is the timing constant that it forms with the associated resistor, forming a passive high-pass network in the frequency domain. The resistor is necessary to keep the reference to ground from drifting and causing major errors in the system. The capacitance should ideally be as small as possible, as large capacitors are often expensive and unreliable. The value must be large enough to keep the noise current from the first op-amp stage from creating a relevant noise voltage at the input, as well as properly block the DC signal.

To prevent the corner frequency from drifting due to component tolerances, the cutoff should be set at least one decade below the desired data. This also prevents the filter from changing the phase of the signal, which drifts by 45 degrees at the corner frequency. For this product, a high-pass network at 10 kHz was deemed sufficient. This results in the timing constant as calculated in Equation. Ideally a capacitor less than 1 uF is desired to reduce the cost, though the resistor should also be minimized.

$$f_{DC} = 10\text{kHz} = \frac{1}{2\pi R_{DC} * C_{DC}} - R_{DC} * C_{DC} = 15.92 * 10^{-6}\text{sec}$$

**Equation 14: AC coupling time constant**

This capacitor will be fixed to the common 0.1 uF that is used in most coupling networks, which requires a 1k ohm resistor to create the timing constant with ample breathing room for tolerances. The leakage of the capacitor must also be considered, which could introduce unwanted current into the op-amp input node. Teflon capacitors are valued for their low leakage current, and will be ideal for this product.

## Dynamic Range Matching

The only “mandatory” step in the analog front end for the CCD output is to match the dynamic range of the output voltage to the input voltage of the analog-to-digital converter (ADC) of the microcontroller. The CCD produces voltages limited by the pixel saturation output voltage and the dark signal voltage. The ratio of these two values produces the dynamic range of the sensor, which correlates to the sensitivity of the measured absorbance. The ADC has a dynamic range limited by the voltage that correlates to all bits set high and the voltage for all bits set low. Driving the ADC with the direct output of the CCD would result in never reaching the maximum voltage of the ADC, effectively reducing the sensitivity of the device and introducing unwanted quantization noise.

The solution is simply to use an amplifier to match the two dynamic ranges. Preparing a suitable amplifier requires several considerations to produce good results, such as gain-bandwidth requirements, slew rate, thermal noise. One important consideration is the variable spectral response of the CCD, which causes a change in the relative response of each pixel due to variations in photon energy. Typical performance curves provided for the selected sensor show significant response drop-off, which would limit the sensitivity of the detector without compensation. This will be covered with a new piece of hardware.

The selected MCU has an ADC with a maximum of 3.3V and a minimum of 0V. The selected sensor outputs a saturation voltage of 600mV and an expected dark current of 5mV. This means a gain of 5.5 is required to match the dynamic range. This will be provided with a non-inverting op-amp buffer which will act as a preamplifier. The non-inverting orientation is preferred for excellent input resistance and the extra 1 added to the gain without larger resistors. This first stage acts as a buffer with very low resistor values and good noise characteristics. The second stage sets the overall gain after the initial noise characteristics have been set.

## Quantum Efficiency Matching

The most obvious method of linearizing spectral response is to use a programmable-gain amplifier (PGA) that can alter the dynamic range matching in real time during readout. This is the best solution to reduce quantization noise during conversion, as the dynamic range always matches the ADC as closely as possible. A PGA used in this configuration must be fast enough to receive the digital control signal and adjust the output gain between pixel rows. For a data rate of 500 kHz, this requires an adjustment time of at most 500ns to prevent sampling the reference level incorrectly. The ideal PGA would have excellent granularity on selectable gains, allowing for the closest possible calibration to the spectral response.

One interesting method is to increase the integration time of the system and run several consecutive scans to produce an entire spectrum. This is actually not possible with our selected sensor, but it runs into the issue of increasing the dark noise of the signal for longer scans. This would require several calibration curves and would likely not improve the data quality by much.

The selected CCD shows a quantum efficiency curve with 0.4 relative response at our peak wavelength. A practical approach to quantization noise reduction is to choose a PGA capable of keeping the dynamic range from dropping below the range of the MSB with as many granular stages as possible. This is unfortunately impractical as it would require a network capable of shifting gains in less than 200ns to prevent the settling time of the sampling amplifiers from interfering with the results. For this prototype, a PGA was not used as the only suitable candidates were not available in DIP packages. The PGA is also impractical since the noise level would scale with the gain of the PGA as well, causing no tangible improvement to the SNR.

Figure shows the block diagram flow for the final planned system. Many similar CCD systems use a hardware correlated double-sampler (CDS) to reduce the impact of dark noise on the final system. The plan for initial testing is to use this hardware double sampler, and if this is not found to be adequate it will be replaced with a software sampler and advanced digital signal processing.

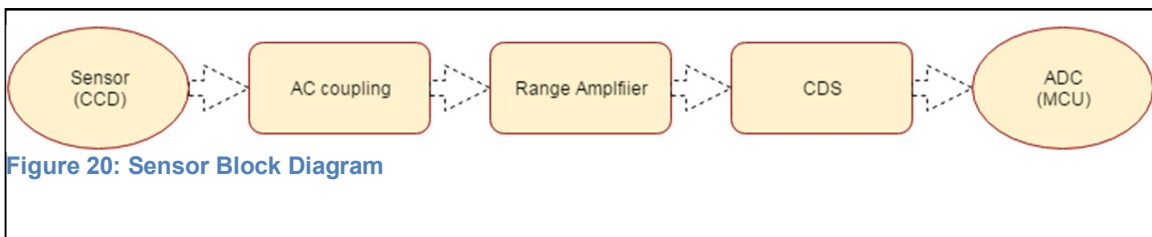
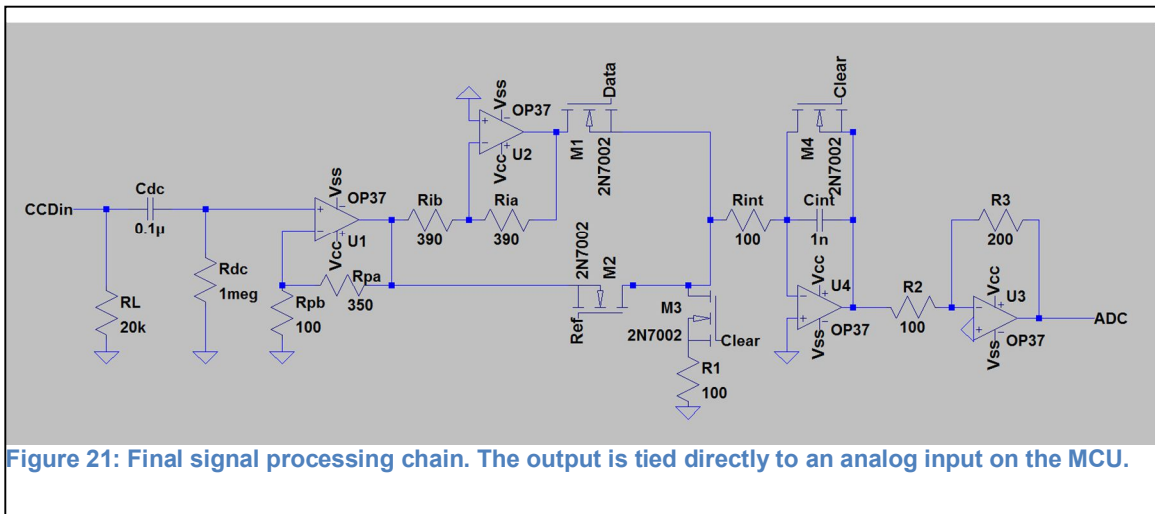


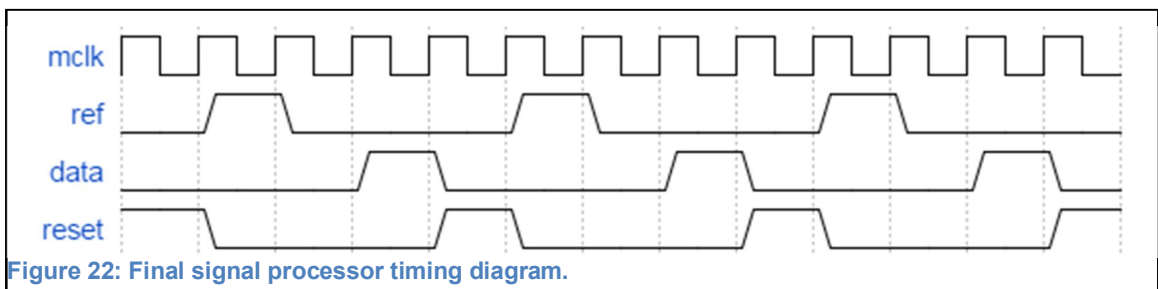
Figure 20: Sensor Block Diagram

## Final Sensor Processor

Figure shows the final schematic that will be used for initial testing and potentially initial PCB layouts. It utilizes a truly resettable integrator as the CDS method with a single inverting buffer. The timing constant for the integrator matches that of the input network. The attachment of an inverting amplifier at the end of the CDS method is potentially redundant, but provides excellent output coupling into the sample-and-hold of the ADC and reduces the dependence on a non-inverting integrator design. The system also uses 3 other amplifiers overall, and a quad op-amp package would likely be the most cost-effective solution even without the final amplifier, so it was deemed satisfactory for this design. The OP37 was used for simulations to prevent having to draft an additional SPICE model for the TL084, but the characteristics are similar enough to make the difference negligible.



The analog switches require additional timing characteristics. Figure shows the initially planned timing diagram. Timings are quantized to the edges of the master clock sent to the CCD for ease of programming and testing, but these can be adjusted in the software during debugging if the values are inadequate.



## Final Considerations

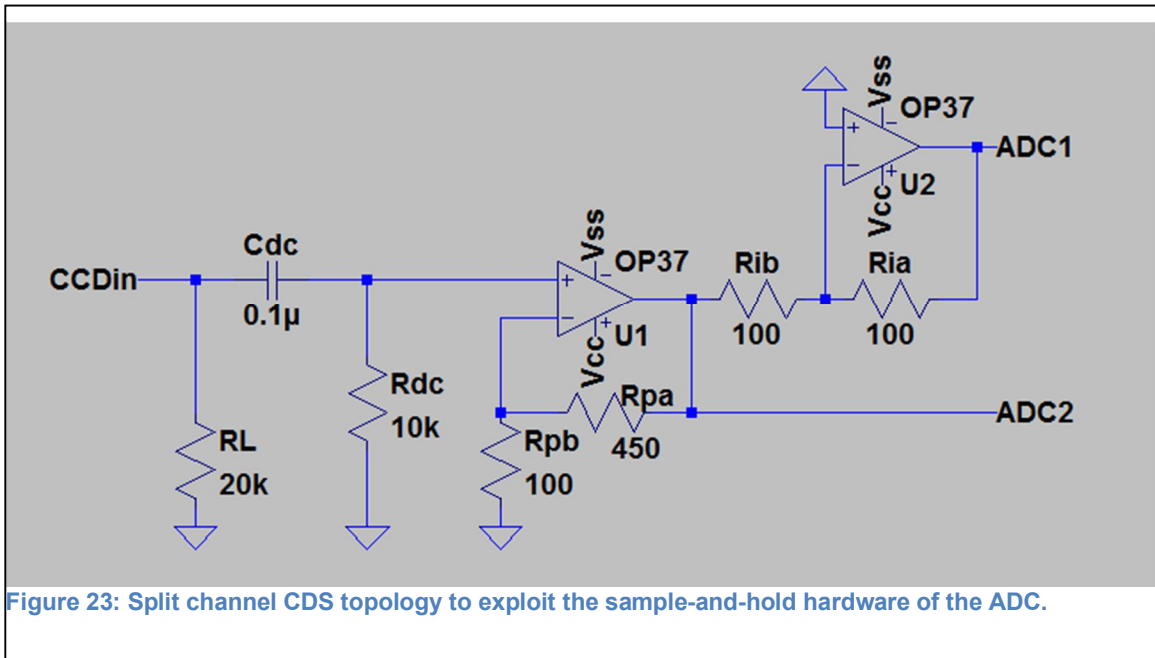
During the research and early testing phase, the desired solution to correlated double sampling was a hardware dual-slope integrator. This solution has been shown to provide good noise reduction when carefully constructed, but during simulation and testing this device did not perform adequately. The identified cause of some issues was the inadequate switching characteristics of the FETs used to control the signal flow in the system. These switches did not provide enough off resistance and injected unwanted current into the signal, causing distortions and rendering the final output noisier than the uncorrelated input. One solution is to include a dedicated FET driver IC to replace these switches, but during price analysis this IC was found to be as expensive as an entire integrated analog front end. Adding the additional hardware cost of the dynamic range amplifiers made this an undesirable solution.

The secondary topology that was designed was the digital correlated double sampler, which omitted the hardware integrator entirely in favor of sampling the data at the Nyquist rate of a desired harmonic, usually 20-50 times the data rate. While this system is extremely robust and allows for theoretical CDS algorithms that are impossible or unrealistic to include in hardware, such a high sampling speed would require an additional monolithic ADC in addition to the selected microcontroller, which introduces redundancy. The ADC12 associated with the MSP430F5529 is only capable of 200 kps with full accuracy; this matches the raw data rate but cannot oversample. While these devices are not prohibitively expensive (as discussed in the part selection), they still rivaled the cost of integrated solutions that bypass any additional hardware cost and calculation time. While this is a good solution for a production device with precisely calibrated optics and sophisticated CDS algorithms, it was not deemed as the best possible method. Some alternative designs that will be tested during the winter break are discussed in this section, though great efforts will continue to be made to improve the originally presented DSI CDS design.

The simplest possible design is simply to use the DCDS circuit and sample once during the steady range of the data pedestal to collect the raw data. This is undesirable as any correlated noise that could have been removed is now lost to the processor, but it is extremely stable as a stand-alone solution and may meet the minimum desired spectral characteristics. Early prototypes will explore this option.

The design presented in Figure is the hardware associated with a DCDS with an additional inverting op-amp on one branch of the output. This design connects to two separate channels on the ADC, providing the output of the dynamic range amplifier in both inverting and non-inverting modes. The idea is to emulate the DSI functionality by forcing the input multiplexer to push both the inverted data value and the following reference value into the sample-and-hold amplifier of the ADC circuit. Since any noise signal that exists would be very small in magnitude - though still above the quantization noise of the ADC - adding the reference noise would only affect the lower bits of the sample, which are still being

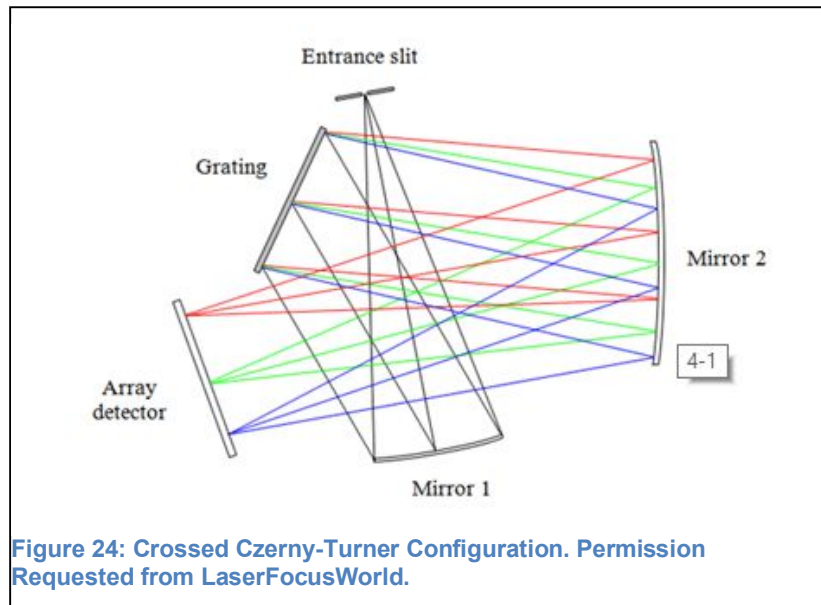
actively converted. This method converts a common error in multi-channel sampling systems into a feature to reduce hardware costs, and forms an intriguing topology.



The last applicable method is simply to use the integrated analog front end that would be used in a production model anyway. This is not a bad solution and in fact is probably the best, but it is not in the spirit of a senior design project. If this design is selected for a prototype model, a discussion will be scheduled with the review board to assess if any additional duties need to be added to guarantee a respectable workload.

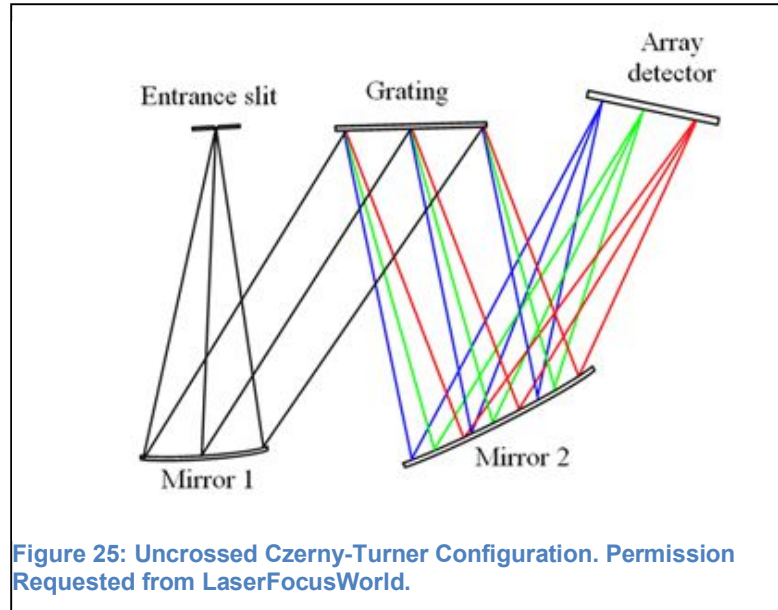
## b. Czerny-Turner Spectrometer Configuration

The components that make up the spectrometer of the spectrophotometer are the spectrograph and the detector array. The spectrograph consists of three key components: the entrance slit, the focusing elements, and the dispersive element. The most common spectrometer configurations are the Czerny-Turner configurations. The crossed Czerny-Turner configuration is shown in Figure.



The Czerny-Turner configurations consist of an entrance slit, two concave mirrors, a diffraction grating, and an array detector. All of these components were described and selected in the Part Selection and Research section of this paper. The Czerny-Turner models can be explained as follows. Light is incident on an entrance slit, which affects the photon flux and spectral resolution of the spectrometer. The light is directed to the first concave mirror, called the collimating mirror. This mirror collimates the white light and directs it toward the diffraction grating. The diffracted light is then incident on the second concave mirror, called the focusing mirror. The focusing mirror focuses the light to an exit slit to a detector array, or directly to a detector array. An exit slit is used if single wavelength measurements are wanted. In our case, we are imaging the diffracted light straight onto the array detector. An exit slit is not needed. The output display will show the full UV-VIS spectrum, along with the absorption peaks.

The Czerny-Turner configuration can be separated into two different types. These two types have their advantages and disadvantages. The first type is the crossed Czerny-Turner. It is a compact design, as shown above. By optimizing the geometry of the system, the spectral resolution of the system can be improved. Also due to its geometry, the image width of the slit is broadened. As a result, the stray light that hits the detector is relatively high. Therefore, this design is used for applications that only require low to medium resolution. The second type of configuration is the uncrossed Czerny-Turner, shown in Figure []. This configuration helps to alleviate stray light problems, which can reduce optical noise in the system. Optical noise and stray light is a bigger problem for low light level UV applications. For that reason, an unfolded Czerny-Turner configuration is a good choice for our spectrophotometer.



### c. Buzzer Interface

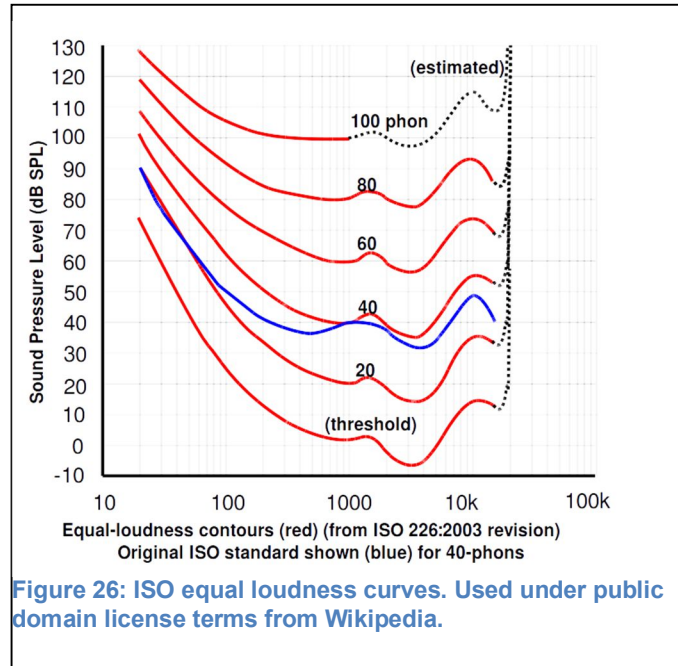
To improve the user experience when completing a threshold test, an audio indication is desired to alert the user immediately. Since no other elements of the device require audio output, this task can easily be completed with a small indicator buzzer. Piezoelectric buzzers are inexpensive and easy to interface with modern microcontrollers without any external circuitry, though voltage buffering will be discussed. These buzzers come in two main varieties, those with attached oscillator circuitry that function at a single frequency, and those requiring external drive but offering more flexibility. Since external drive is possible with a microcontroller, both options will be discussed.

The desired sound pressure associated with an indicator is somewhere between conversational speech and the threshold of human discomfort at atmospheric pressure. Some level of volume control and muting should be provided to allow the user to select a comfortable level for the indicator if they choose to use it. OSHA regulations show a maximum permissible exposure limit of 90 dBA for an 8 hour workday, which reflects a standard work load for the target customer. Supplementary material from OSHA shows normal conversation around 60 dBA, so the buzzer should produce a sound level between those points.

The hearing range of humans is usually cited as 20 Hz to 20 kHz, though the top range falls off with increased age. Figure shows ISO equal loudness curves that show a clear minimum in required sound pressure vs perceived loudness for sounds nearing 4 kHz, so



the ideal buzzer should produce a sound around this frequency. Commercial buzzers are often designed with 4 kHz resonance, so this is easy to achieve.



Some piezo indicator buzzers contain circuitry to generate required waveforms automatically when provided with the necessary operating voltage and current. These parts are convenient to use when no oscillator is present, but this convenience is not necessary since the MCU in use can produce PWM waveforms. Producing these waveforms requires a dedicated timer in the software, but does not otherwise reduce the performance of the MCU. The price of buzzers without oscillator circuitry is generally lower since there is no electronics required, so these buzzers are ideal.

## Buzzer Realization

The MCU is capable of driving most indicator buzzers directly using TTL levels. This is not a good solution because there is no protection for the MCU from overdraw or fly back voltage from the inductive behavior of the piezo element. A better solution is to use a small voltage buffer to act as a switch. This is realized using a BJT. Figure shows a BJT switching buffer with a small bypass diode to prevent damage to the transistor during fly back. Rb is a 1k resistor used to limit the current through the base. The impedance model used for the buzzer was derived from the datasheet current consumption at the indicated voltage. While this model does not accurately reflect the inductive nature of the buzzer, it is sufficient for quiescent current analysis. All components consumed less than 250mW, so no extra thermal considerations are needed.

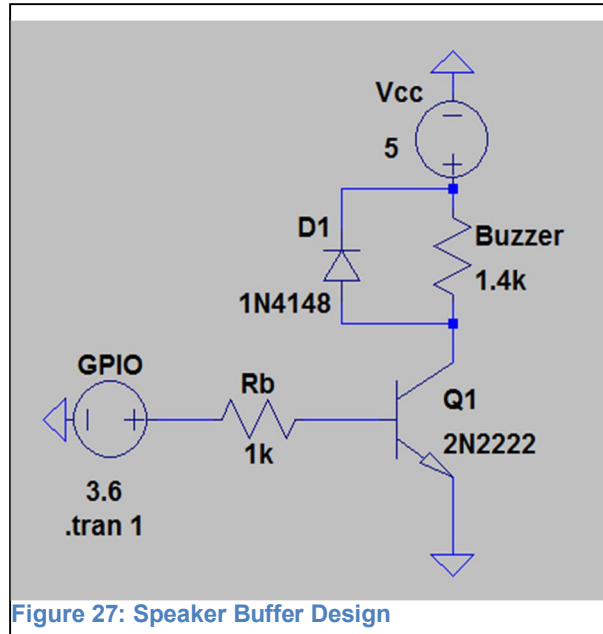


Figure 27: Speaker Buffer Design

The BJT acts as a switch in the case of the PWM waveform applied to the base. A BJT was chosen over a FET implementation since the current is low and the load is relatively light. The sensitivity of operation is not very high for the BJT, but one must be selected with a transistor frequency above the desired 4 kHz. Table shows the tabulated maximum current and power values for this schematic.

Component	Max voltage (V)	Max Current (mA)	Power Dissipation (mW)
Rb	2.85	2.84	8.09
Buzzer	4.97	3.56	17.7
Q1	5.00	6.4	2.21
D1	5	0*	0.001
GPIO pin	3.6	2.84	10

Table 22: Speaker Buffer Component Stress Values

#### d. Micro Control Units

The microcontroller unit is a key part of any electrical system. It allows the system to complete multiple tasks, and has many components to it. This device will be the center of the system, and will control the majority of other components in the system. It will do this by connecting to external hardware, and reading code loaded in either the device it's connecting to, or from code loaded onto it. For our particular system, we chose to use a MSP430F5528. The reason for choosing this controller is because all of us have experience with MSP430s from previous classes. This ensures that we can get right to work on it, and we don't need extra time to learn how it works. We are also provided resources for this component on campus, and through teachers who have helped us use this device. There

are many choices of microcontrollers to choose from. Some of the more popular devices are Arduino's, and while there seems to be a good amount of support for these devices, it would be similar to using the MSP430. One of the main differences between these two devices is price. The Arduinos can cost anywhere from \$20.00-\$40.00, while the MSP430s cost from \$1.00-\$6.00. The catch here is that the MSP430s require a development board which can cost anywhere from \$10.00 - \$100.00. In order for us to save money here, we will be using the development boards from the TI lab on campus. Most microcontrollers have the same ports, but are usually structured differently or have a different number of ports. This means that most of the external devices needed for this project will be compatible with the MSP430, just as it is compatible with the Arduino's and other microcontrollers.

The ports needed for our microcontroller are the SPI, I2C, general pins, ADC, UART, and USB. These ports will be supporting the power to the light, the LCD, the buttons for the interface, and will receive information from the optics portion. One variable to consider for our microcontroller is the memory. There will be two types of memory used, non-volatile, and the RAM. The non-volatile memory is saved on the controller, while the RAM is lost once the system is turned off. This means we need a microcontroller that has enough memory to support our external components and to run the code stored on it. This is why we chose a controller with a large amount of memory, this way we will have enough in case we need it. One factor in choosing the microcontroller was the ADC channels. For our system we need at least one ADC 12 bit channel. ADC stands for analog to digital converter, and it represents data given to it and converts it to bits. A 12 bit ADC will be able to convert data will be able to represent 2 to the 12<sup>th</sup> power numbers. This will be useful for connecting to our optics portion of the system and CCD in order to read in the data. The majority of MSP430s viewed had a slope ADC. This just means they have 2 to the 1 power of numbers to be represented. This limited our choices, but thankfully, since there are so many MSP430 microcontrollers, we were able to find one that fit with all the other specifications that we needed.

Another variable to consider is the operating temperature and power. This has to be considered with the operating temperatures of the other external devices in order to keep the heat of the whole system at a minimum to avoid overheating. And another threat to the system could be if the heat creates a problem with the box holding everything. The MSP430F5 series are all low power microcontrollers. This means that they use a low amount of power compared to other microcontrollers, and will produce less heat, and need less power from the power supply to run efficiently. All these factors are reasons why we chose MSP430F5 series. This series of MSP430s has a low operating temperature, has a USB port so we can easily connect devices with USB, and has enough SPI and I2C channels to connect our external devices. The biggest differences between the specific MSP430s within the 5 series are the ADC channels and the amounts of memory given to each microcontroller. This is why we chose the MSP430F5528, since it has enough memory and the correct ADC channels that we need.

## i. MSP430F5528/Other Options

After considering many choices, the microcontroller we chose to use for our system is the MSP430F5528. For selecting the correct microcontroller for our system, we needed to consider many different factors. We needed the correct ADC, SPI, and I2C channels to support our system. We then need a controller with low operating temperatures, good amounts of memory, and a low price. One option we considered was the MSP430FR4xxx series. This microcontroller had LCD support, but no USB compatibility, and not enough memory for the code and the rest of the system. In order to connect a USB for this system, we would need to connect it through SPI/I2C pins, which are already being used, and we would need to code it, taking up more room than we already have. We would also have to adjust the clock speeds, since there would be an issue with that when trying to connect an external USB and have it connect to another device.

Then we considered the MSP430F6xxx series. As seen in Table, this controller had LCD support, USB compatibility, and a good amount of RAM. The reason we didn't choose this one was because it had too many general purpose pins. Since the LCD support given would have taken too much wiring, it wouldn't be able to support an LCD with a good enough resolution to display a graph. The way the LCD support works on the MSP430s is that they give you a LCD port. In that port, there are many segments, each of which will end up connecting to your screen, depending on the resolution. In this case, the biggest support we could find was for 160 segment LCDs, which are big enough to display a clock and simple letters, but does not have a big enough resolution to draw a graph. Another problem with this LCD support is the wiring. In order to connect each segment to the screen, we would need as many wires as there are segments. The benefit in this is that the user is able to draw to each segment easily through the code, and won't have to look too much into the libraries given with LCD. This is feasible when looking at very small LCD screens, but when the resolution of the screen needs to be bigger, it becomes a lot easier to just connect it through the SPI and I2C ports. Then we can use the libraries given to us by the LCD, or lookup how to draw the graphs we need to it through the given data sheet.

We then considered some of the Arduino products, which use AVR Atmega as their microcontroller. These products were very similar to the MSP430s. One of the main differences was that the Atmega used its own software library, different than that of the MSP430. They did, however, have the same output channels, such as ADC, SPI, and I2C. They also had USART, while the MSP430s had UART. USART supports asynchronous and synchronous receivers/transmitters, while UART only supports asynchronous. This doesn't make much of a difference since each microcontroller has both I2C and SPI, which are synchronous either way ["Using USART or UART"]. One of the main reasons for choosing an Arduino device would be for the LCD support it has available on the internet. It is very easy to find examples of connecting an Arduino with an AVR Atmega to a LCD, which would be helpful if we were to get stuck when trying to connect them. MSP430s, on the other hand, have few resources on connecting to LCDs. The LCDs that MSP430s are typically shown being connected to are just simple 140 segment LCDs, which are

normally used for displaying numbers and letters, and would never be able to display a graph. Despite all of available resources, both microcontrollers will have the same output ports, such as SPI and I2C. This means that connecting both of them will be the same, the difference might occur when there is code needed to use the LCD. And since the MSP430s are cheaper, and we already know how to write some of the code for it, this helped us to choose from the TI brand of MSP430s.

The last option we considered for microcontrollers were PIC microcontroller units. They functioned the same way as many of the MSP430s and Arduinos, but there were fewer options available. The MSP430 was able to give us a large list of microcontrollers to choose from, whereas the PIC failed to do so. The only benefit was that there was some support available on the internet for connecting PIC to LCDs, but not as much as the Arduinos. We also would have to learn the code and software for PIC. One thing learned while researching the PIC was the similarities it had to the Arduino products. It seems as if the Arduinos are the most popular, while PIC is right behind them. We believe that the MSP430 we are using is less talked about because it is generally newer.

So as an available option, it was good to view each microcontroller just to see what they have to offer, but in the end, the MSP430F5528 seemed to be the best to us. The hardest part in finding a suitable microcontroller was finding one that left no part in abundance. Most choices we found had a large amount of general purpose pins. This is why the MSP430F552x series, seen in Table-1, was the best for our system. It contained enough general purpose pins to not be in abundance, and the series was advanced enough for it to have USB compatibility. This saves a lot of memory for the code, and allows us to use the SPI/I2C ports for other external devices. It also didn't contain the extra LCD support that was not needed, and would have just been an extra feature not used. Another variable to consider was coding. Each microcontroller we looked at could be coded using the C language. But for MSP430, we already know how to easily code them in Assembly language, and have experience with mapping the pins as well. On top of this, we have done examples of using pins through both C and Assembly, which will most likely be done differently through other microcontrollers.

Part Number	MSP430F5527	MSP430F5528	MSP430F5529	MSP430F6635
<b>Description</b>	16-Bit Ultra-Low-Power Microcontroller, 96KB Flash, 6KB RAM, USB, 12Bit ADC, 2 USCIs, 32Bit HW MPY	16-Bit Ultra-Low-Power Microcontroller, 128KB Flash, 8KB RAM, USB, 12Bit ADC, 2 USCIs, 32Bit HW MPY	16-Bit Ultra-Low-Power Microcontroller, 128KB Flash, 8KB RAM, USB, 12Bit ADC, 2 USCIs, 32Bit HW MPY	MSP430F663x Mixed Signal Microcontroller
<b>CPU</b>	MSP430	MSP430	MSP430	MSP430
<b>Frequency (MHz)</b>	25	25	25	20
<b>Non-volatile Memory (KB)</b>	96	128	128	256
<b>RAM (KB)</b>	8	8	10	18
<b>GPIO Pins (#)</b>	63	47	63	74
<b>I2C</b>	2	2	2	2
<b>SPI</b>	4	4	4	4
<b>UART</b>	2	2	2	2
<b>DMA</b>	3	3	3	6
<b>ADC</b>	ADC12 - 14ch	ADC12 - 10ch	ADC12 - 14ch	ADC12 - 12ch
<b>Additional Features</b>	USB, Real-Time Clock, Watchdog, Temp Sensor, Brown Out Reset, IrDA	USB, Real-Time Clock, Watchdog, Temp Sensor, Brown Out Reset, IrDA	USB, Real-Time Clock, Watchdog, Temp Sensor, Brown Out Reset, IrDA	USB, LCD, Real-Time Clock, Watchdog, Temp Sensor, Brown Out Reset, IrDA
<b>Operating Temperature Range (C)</b>	-40 to 85	-40 to 85	-40 to 85	-40 to 85
<b>Package Size: mm2:W x L (PKG)</b>	196 mm2: 14 x 14(LQFP)	See datasheet (DSBGA)	196 mm2: 14 x 14(LQFP)	49 mm2: 7 x 7(BGA MICROSTAR JUNIOR)
<b>Approx. Price (US\$)</b>	3.47   1ku	3.55   1ku	3.58   1ku	4.79   1ku

Table 23: Comparison of MSP430Fx family. Courtesy of Texas Instruments.

Once we picked our microcontroller, we needed to look at how to program it. TI offers a development board called the MSP430F5529 Launchpad. This product costs \$12.99, and we have an option of seeing whether or not it is available in the TI lab. This development board will allow us to program our microcontroller by connecting it to a computer via USB. This is done by coding everything in their given software, Code Composer Studio, and saving it to the microcontroller. Along with the development board, TI also offers booster packs. These are external devices which can be connected to the development board. Booster packs will allow us to easily use an LCD for the system.

Another option that led to our final decision was the ADC. This analog to digital converter, as mentioned in section 4 g, needed to have at least one channel at 12 bits. Most of the MSP430s we saw had an ADC, but it was categorized as slope, meaning it converted and represented 2 bits of data. The Arduinos and PICs usually came with the 12 bit ADC channels we needed. But since there were such a variety of choices from the MSP430 family, we were able to find a microcontroller with such an ADC that also met all of our other requirements.

## ii. Memory

Within the MSP430F552x series most of the microcontrollers have the same number of ports, but differ in general purpose pins and the amount of memory. We ended up choosing a microcontroller with 8KB of RAM and 128KB of non-volatile memory. We needed a controller that would have enough memory to hold the code of our system. In Assembly, each line of code or instruction is anywhere from 2-6 bytes. Which means it could take 512 instructions to get to 1KB of non-volatile memory. When looking at the 128Kb of non-volatile memory we have, it can seem to be too much. We believe that not all the memory will be used, but that it is necessary to have in case we end up needing more than what we originally planned for. It is better to plan to need more memory, than to have less and come up short. This would result in us buying a new MSP430 which would have more memory, or buying external memory.

When talking about memory, there are two types. One is the non-volatile memory, and the other is the RAM. The RAM will get deleted once the system is turned off, hence volatile, while the non-volatile memory will stay on the system in storage. The code for our program will be stored in the non-volatile memory section, while the RAM will be used for things such as displaying the LCD screen if needed. Another option of memory is FRAM. This memory will show up as one number instead of two different memory types. The benefit to this is that you can then split up the FRAM into your non-volatile memory and your RAM, depending on what your system needs. This option shows up mostly on the MSP430F6xxx series, PICs, and Arduinos, and seems to be a newer way of approaching memory on microcontrollers. We still chose not to pursue this third option, since the microcontroller we found had enough non-volatile memory, and enough RAM. For a backup precaution, we have considered using external RAM. This will be used if the LCD

we choose needs more memory than expected. Depending on the resolution of the LCD, our microcontroller would need to have a lot of memory.

Another option for this problem with our LCD is that the LCD could come with its own memory, which would save us all the memory it was going to use. The only issue would be the price. If we select a LCD with its own memory and a high resolution, it would cost a lot more than if we had just gotten a LCD with no memory and a high resolution. Now the code for the system will be able to store data from an input, and create a graph, in order to display the data to the user. This will be done in C, and not assembly, which will be more difficult to determine how much memory will be taken by this code, until it is written. This is another reason why we chose to go with a lot of memory, that way if it turns out we needed more memory, we will have it.

The usage for our memory will be for many different components. The CCD will interact with some of the general pins we have, giving the microcontroller input. This will then be held on by the system and used to create graphs displayed onto the LCD. Our code will be doing this by taking given numbers, plugging them into equations and formulas given to us from resources specific resources, and then plotting this information on a graph. There will be a part in the code that responds to the light, which will turn it on and off quickly for the CCD to read in its data to the microcontroller. There will then be memory set aside for the code to display this graph computed earlier to the LCD display. Computing the graph and displaying it will be the portions of our code which will take up the most memory. The memory will also have space for code which will interact with the USB port. We will be able to pull the graphs from the USB onto a flash drive by saving the image of the graph, or onto a computer by saving the image of the graph.

One option that can be used for dealing with the light is using the PWM port. PWM stands for Pulse-Width Modulation. There a signals being passed through this port, and the user can specify on which signal that the port will give power to a unit. If we use this to turn on our light, we will be able to easily turn on and off the light without having to go through too much code. This will save us some memory, and will also be better for power usage, since there is already a signal being pulsed through the port. Although seen as a good option, the voltage needed to power the light source will be much higher than provided from this port, and the power will come directly from our power source. It is good practice to view each option made available to us, but they are not always the most feasible choices.

### iii. SPI/I2C

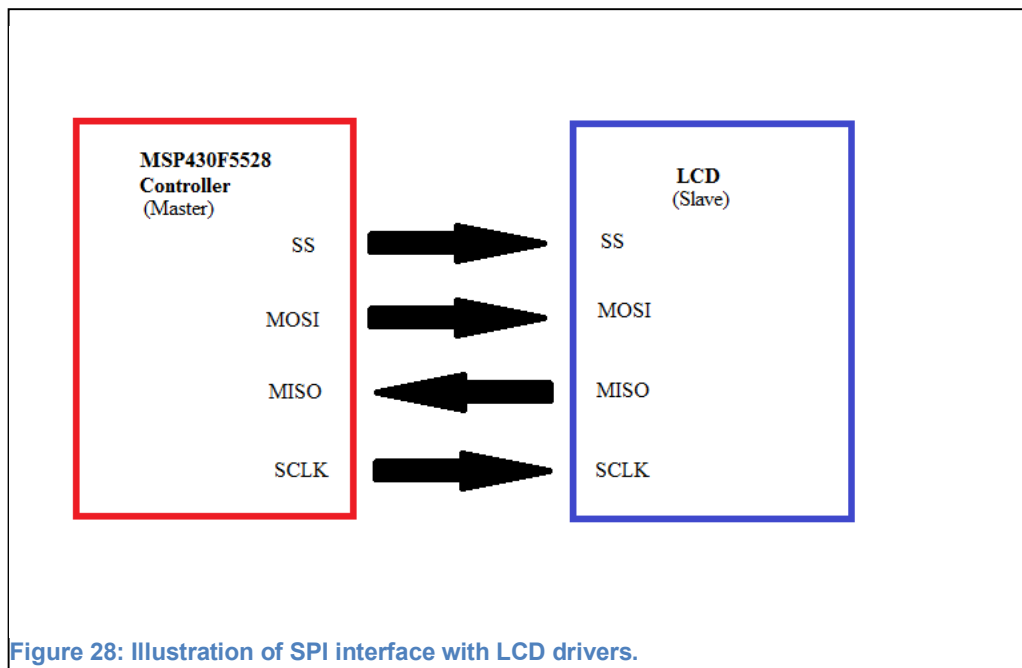
The SPI and I2C ports are used for the connections between the microcontroller and the LCD. They are serial ports under the USCI\_A and USCI\_B. USCI\_A are the asynchronous serial ports. For our microcontroller, this controls the SPI. This asynchronous communication means that the two devices connected will not be able to transmit and receive data simultaneously, but will not have to synchronize their clocks, allowing for certain operations to happen faster. The synchronous serial port on our



microcontroller, which is the USCI\_B, contains an I2C port and another SPI port. This means that while the devices are connected to each other, they will run on the same clock speed, thus they can transmit and receive data simultaneously [London].

The SPI, or Serial Peripheral Protocol, has 4 ports on it. In these 4 ports are the synchronous clock, a slave select signal, and two data lines, one from master to slave and another from slave to master. The master, which will be the microcontroller, sends data or requests data from the slave. The slave in this case will be the LCD screen. This data will include the graphs that need to be displayed onto the screen. The slave select will be the system choosing which external device it needs to be transferring the data to, which in this case it would choose the LCD as the slave. And the clock is needed since the microcontroller and the external device need to be running on the same clock when they are passing data through the data lines [“Protocols”].

The image below, Figure, shows the connections made from the master, our MSP430F5528, to our slave, the LCD. The SS is the slave select, and the SCLK stands for the Serial Clock. The MOSI stands for Master out Slave In, and is where the master sends data to the slave. The MISO stands for Master in Slave Out, and is where the slave sends data to the master.



In the I2C, there are only two ports on it. These are the serial data line, and the serial clock line. The serial data line will be used for transferring data, while the serial clock line will be used for connecting the clock between the two systems [“Protocols”]. I2C is allowed to connect to multiple slaves, which gives it an advantage over SPI, since SPI can only connect to a limited amount of slave devices, depending on how many are connected. The main differences between the two systems are shown when needing to connect to two or

more external devices. For I2C, it will only take two wires to connect to an external device, and has the ability to connect to even more devices with fewer wires than SPI. SPI needs up to 3 to 4 wires to connect to an external device, and when connecting to more than one, will require more than 4 wires. A problem that will appear here is transferring data to the correct external device. For the SPI, we will need to set each master to the slave in order to transfer data to the correct device, without any interference. The I2C, however, will detect which slave to give its data by finding the address and connecting the transfer.

The main reason to use SPI over I2C here would be for a faster data transfer. Since the I2C will have to look up the address to confirm whether or not to transfer the data, the time it takes to actually transfer data will be longer than the SPI data transfer. The SPI also has a better data transfer rate than the I2C port. The LCD option of the booster pack uses SPI communication. This will be simple for us since we are planning on connecting only one device to the SPI. This will leave open the I2C connection for us to use for other external devices. This connection will require 4 wires, and use an 8 bit SPI. Most other smaller LCD screens require the use of SPI, while the larger screens which use a higher resolution tend to use I2C. Some optional LCDs even use a UART connection or require any 4 pin interface with any serial host. This gives us a lot of room to work with, and allows us flexibility in choosing our different external devices, knowing that there will be more options, and we won't be limited to only choosing one connection port.

The microcontroller we are using for this system contains 2 I2C, 4 SPI, and 2 UART ports. We used the same mentality for choosing memory here. We went with more ports than we originally required, just in case something happens to come up. If we need to go with another LCD screen, or if too many external devices need to use a certain port, we will have plenty of them. This will allow us to expand our search for these external devices, so we are not limited by searching for devices with specific ports. The majority of the microcontrollers we viewed had 1 I2C, 2 SPI, and 2 UART ports. This allowed us to consider many microcontrollers, since they all had roughly the amount of ports we would end up needing. But in the end we decided to be safe and choose a controller that had more than just the necessities.

In order to determine if our SPI connections were working, we came up with a method for testing each part in our system. If we had just connected everything together, if something didn't work correctly, the only way to test and find out what the actual problem was, was to test everything individually. Technically speaking, we could have tried fixing the problem, and then running everything together through trial and error. But in order to find the actual problem, narrow it down, and fix it, we decided to go through these debugging methods instead. In order to test if the screen was actually working, we would plug it in to the launchpad, and try to run our code from there. If it didn't work, then we knew the problem was with our LCD not working, especially since we have more than one launchpad to test this with. At this point, the problem could also be the code itself, but since TI comes with example code for the launchpad and the LCD we were using, we could just revert back to that old code to determine if that was the issue. After checking to see our LCD

screen works with the launchpad, we can then test the voltages of the 3V3 and 5V port we have on our PCB, and make sure those are running correctly. Now that we know the LCD is working, as well as our power on the PCB, we now need to test if the SPI ports on our PCB are working as intended. To do this, we need an oscilloscope with four wires. We then connect each wire up to the PCB, one at a time. Each time we connect a wire, we will power the PCB on, run some example code, and configure the settings on the oscilloscope in order to read the correct data we are trying to see. Now, if we run the code on the microcontroller, and the oscilloscope is showing us actual data, since it should be sending data through the SPI ports, then we will know the problem is that we are not connecting our LCD through SPI correctly. However, if the oscilloscope is not showing us anything useful, than that means the two options. One option is that the code that is initializing the USCIA and the SPI is not correct. Or the second option is that our PCB is set up wrong. If the answer is the second option, we would need to order a new PCB. But if it's the first option, then we just need to initialize the code correctly. So we will first try and fix the code by searching back through the user manuals that TI gives us for our specific microcontroller, and try and see if anything has been initialized incorrectly. If we try this to no avail, then we must try and order a new PCB, this time either trying to correct the system by using USCIB, since the launchpad uses this connection type, or by trying to use 3 wire SPI mode instead. This is how we would fix any issue with our SPI connectivity.

#### iv. Temperature

A big concern for our system was the operating temperature that will be produced by the microcontroller. The MSP430F5xxx series is a 16 bit ultra-low power microcontroller. The operating temperatures listed for this controller are anywhere from -40 to 85 degrees Celsius. The absolute maximum temperature listed for this item was at 95, where the device could sustain permanent damage. According to the datasheet given for the MSP430F5528, temperatures tend to reach around 80 degrees Celsius when it is operating at around 3 volts. This is expected, and there are a couple of ways to deal with this problem.

The first solution would be to incorporate fans into our system. This will help keep the microcontroller cool, especially while it is inside of a box. This MSP430F5528 comes with a temperature sensor that we could use to tell us when we need the fans to speed up. This can be done during the testing of the hardware. It is impractical for us to require the user to keep track of the temperature and adjust the fans accordingly. Instead, during the testing, we should keep track of the temperature of the microcontroller, as well as the temperature of our other external devices, and adjust the fans for each specific task the system will use. This can be done in junction with the results provided to us in the datasheet in order for us to adjust the fans correctly. We can also choose to adjust the fans by checking to see how long the system has been in use for. The control for these fans can be implemented in the code stored in the microcontroller. This is another reason having the extra memory on the microcontroller is a good idea, should an issue such as this arise.

An issue with the fans would be space. Since this system is to be used in labs, we don't want the box to become too big just to fit in fans. The benefit to our system size, besides the LCD and CCD, is that not much else of the system will be too big, thus we won't require having a big box. This means we can get a bigger case for our system for the fans without worrying about going over on size. We will find out these measurements by determining the size of all of our components, drawing them out on schematics, and trying to come up with an optimal design for the system. This will show us how much space we have to work with, and will let us create room for fans in case they are needed.

Another solution to this heat problem would be to create a case that will have many vents on it. The case will already have some vents on it, but we will make more vents, and adjust the casing so that the system runs a good temperature. These vents won't be huge openings, but rather small little holes that will allow air to travel throughout the case. The other benefit to this style of casing is we can add this idea with our fans, and the case might be better than just one of these ideas. If we add in fans to the system, and allow for there to be many vents on the case, this will allow for the maximum air flow in our system.

In comparing temperature options to other microcontroller units, the results were usually the same. For each MSP430FR and MSP430F series looked at, they each had the same operating temperatures, and maximum operating temperatures. They differed slightly in the specific temperatures in each data sheet, but were very close for the temperatures listed in the MSP430F5xxx series. Even the Arduinos and PICs that were viewed would usually list their operating temperatures to be anywhere from -40 to 85 degrees Celsius, exactly the same as the MSP430. The only difference was that the Arduinos provided a clearer display of their operating temperatures and examples of them than the PICs on their datasheets. And the MSP430 datasheets contained very detailed information on these values, specifying graphs to view.

## v. USB

The MSP430F5528 that we picked for our microcontroller comes with USB compatibility. USB stands for universal serial bus, and is one of the main ways devices connect to each other now. The most up to date version of USB is 3.0, but the USBs we found on many of the microcontrollers only supported version 2.0, such as the microcontroller we chose. The biggest difference between USB 2.0 and USB 3.0 is the data transfer. The data transfer is 10 times faster for the 3.0["Difference between USB"]. This issue will not directly affect the way our system works. For flash drives, our USB drive will be sending small files and pictures of graphs, so the data transfer there will not be very noticeable. However, when it comes to connecting to a computer and transferring the data, this is where USB 3.0 could be better. We are still going to be sending these files to the computer, but there will be more options for the computer to use instead of just saving these files. Another thing to note is that USB 3.0 is fairly new, and up until recently, everyone was using the USB 2.0. This means that while the technology can be upgraded in the future to make this connection

faster and more efficient, there won't be a big noticeable difference in the way USBs interact with our system.

In order to have this USB connection, we will be purchasing a female USB, which allows connections to regular USBs. We will connect this using the mapping diagram from the datasheet of our microcontroller, and map it through the PCB board. Then, we will have to go in more depth when coding for the USB. This code will have to state the connections made to the USB and how to interface with them. It will also determine what kind of functions it will have with each device. For a flash drive that is connected, through buttons on our system, the user will be able to save a picture of the graph on a file. Through connecting from USB to a computer, the system will be able to access the graphs shown on the LCD. The USB connections to the actual microcontroller unit are detailed in Figure.

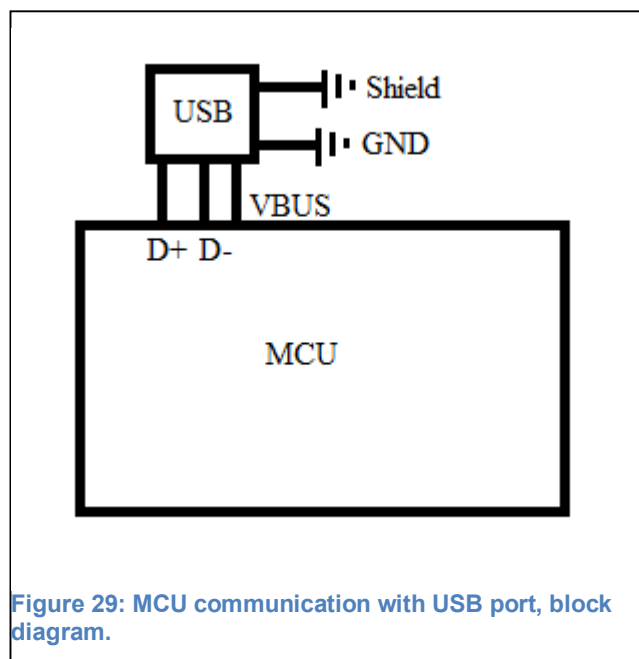


Figure 29: MCU communication with USB port, block diagram.

To understand more about how the USB actually works, we viewed the IEEE Standard for a High-Performance Serial Bus. This standard showed us the inner workings of serial buses, such as an USB. Although we are not constructing the USB itself, this standard shows us minimum requirements for how they are built, and what is needed from data a data transfer standpoint. From this standard, we can see that the minimum requirement for the data transfer needed from the cable is 98.304 Mbits/s. This means that our USB port should support a greater data bit transfer than that value [1.6.2, Standard Serial Bus]. The durability needed from our USB is also specified. The minimum amount of connections between the USB and its port, before performance is hindered, is 1500 cycles [4.2.1.8, Standard Serial Bus]. The USB on our system will be more advanced than the minimum,

and should be able to perform for more than double this amount of cycles. The testing done on these serial buses shows us the limits and how much stress our USB will be able to perform under. It will be able to withstand a voltage of around 500 Vdc, and can maintain performance under high humidity [Table 4-5, Standard Serial Bus]. We will not be putting our USB under such extreme conditions, but these tests show us the durability of USBs, and allow us to proceed without us having to test the USBs. We can safely assume now that our USBs will have no problem working under the conditions of a lab, and will not be disrupted by the amount of data we will be passing through it.

This standards document also provides us with a list of code operators for the programming language C/C++. These operators are commonly used in C code, and are known to many programmers. The purpose of this list is to provide the programmer with a list of usable operators, and to include their definitions, so that there is no confusion when programming the USB. This list will be used by us when we code the USB functions. Although we will be using USB developer's package from TI, our code will be done using the C language. One benefit to using this is the compatible examples that will come along with the program. This means that we might not have to explicitly type out every piece of code that contains these operators, but using this table given to us, we will be able to understand more thoroughly how each line of code works. This better understanding of our code will allow us to expand on these examples given to us, and allow us to create better functions suited for our project.

This list can also be helpful to us when debugging our code. We will use the list as a reference when we need to know what a specific segment of code is doing, and will allow us to correct mistakes by finding the definition of these operators, and learning how they act with the rest of the code. Not only are there definitions given to us explaining each operator, there is also an order that the machine will go through, showing us how the code will work. When processing read and write requests, the system will go through the order of first finding the definition of the code written. Then it will find the initial values given to the data, and will then read the values. Finally, if it needs to write, it will write at the end of the process. This order gives us a better understanding of how the read and write functions work for USB, and will be helpful to us when connecting our flash drive, since the main purpose of this USB will be to write information to a connected flash drive. Also related to the code, towards the end of this standard there is a section on commonly used code and functions. These functions and definitions include code examples using the C language. They are all commented, and provide a lot of detail and explanation into what action each line of code is performing. As already mentioned, even though we will be utilizing more of the coding examples given to us from TI, this is a good back up resource to have in case the examples do not provide all that we need for our program. These listed examples go over anything from functions on port connections to useful variables being defined. There are also functions that can be used for testing, giving us a way to see the results of our code explicitly, which will be helpful for debugging our code if any errors occur.

Transactions that take place while the USB is in contact with the system are also gone over in this standard, and show us a more in depth look at how some of the operations are dealt with. The transactions gone over in the standard include reading, writing, and error handling, just to name some. Each data transaction communicates with a transaction data request. The transaction type is what determines the next steps in the system. If the transaction type is READ, the system will go forward with the steps to handle reading from the USB. If the type is WRITE, the system will go ahead with the steps for writing to the USB instead. These requests contain a fair amount of data for the system to move on. This data includes a destination address, the data to write, if it's meant to write, and the data length. Once the data is confirmed, the data is then linked to the system, and the appropriate responses take place. The biggest difference is in the error handling portion. Instead of data being passed through, the transaction layer will handle the errors. This means no data is passed, and there is no need to link to the system [7.3.1, Standard Serial Bus].

Another Standard from IEEE for Discover, Authentication, and Authorization in Host Attachments of Storage Devices explains to us how USBs are to connect and receive data. They specify four layers of communication. There is a functionality layer, which holds the code that will be used to interface with the USB. There is then an IEEE 1667 layer that contains a number of silos. These silos then interpret the code, and find the appropriate responses for the application. This layer is where much of the actual data communication seems to be made. The next layer is the transport, which will "define the method used to communicate IEEE 1667 Silo commands between the host and the device." [Figure 3, Standard for Discover] The last layer, called the physical layer, is the actual device. This is where our USB will be, and will finally receive the data transferred from the previous layers. [Figure 3, Standard for Discovery] This helps us to better understand how data communications are actually made through the USB, and shows us why the code being used to interface with our USB will be important.

## vi. LCD Connectivity

For our system, we will need a LCD in order to display the data on a graph to the user. The LCD we need must have a display resolution of at least 300x200, and be a large enough size for a graph to be easily seen from it. The best way to connect these LCDs will be through the SPI/I2C ports, meaning we will need to find a screen that supports these serial ports. Since there will be buttons on our system around the screen, we do not require the LCD to have touch screen technology, but since most modern screens have this feature, we will implement some functions for it if necessary. The screen must also come with an understandable datasheet and other resources to help us connect and program the LCD.

The most readily available LCDs we were able to find screens that came with Arduino functionality. These screens came with tutorials and guides on how to connect their LCD to the ports of an Arduino. While this option seems viable, we have already chosen to use a MSP430 instead. One thing to note, is that the majority of these screens that connect to Arduinos, can also connect to a MSP430. The way these screens originally connect are through either the SPI or I2C port, which MSP430s have.

Another option for our screen is through the TI booster packs. These booster packs, as mentioned above, are supported by TI, and have the ability to easily connect to many MSP430s using pin connections. Additionally, many of them come with very specific datasheets which provide us with tools we need in order to connect them to a MSP430 and show us how to draw to the screen. This would be the best option to go with from a connection point of view. These packs will knowingly provide support to MSP430s, have been proven to work with them, and will give us the resources we need in order to complete our system. One issue with these packs is that booster packs are not only for LCDs. This will limit our number of choices, since there are more booster packs without LCDs, than packs with LCDs.

Alternatively, we can seek a third option of LCDs. These are the independent screens that don't directly support a specific microcontroller. The benefit to this is the vast amount of information in the datasheets, and how specific this information is. The only set back would be the trouble finding resources helping us connect to our specific microcontroller. These screens will connect through the SPI/I2C ports, and usually carry their own memory, separate to that of the microcontroller. The only concern with these LCDs is the price. These screens usually cost more than the other two screens mentioned above by about \$10.00. This price difference may seem small now, but it is important to keep in mind that the bigger we make the screen, and the higher the resolution of the screen, the bigger that price difference becomes.

One issue in finding LCDs is the size. Most screens we found are anywhere from two to four inches diagonally. For a reference, this would be the size of the screen of a smart phone, or a bit smaller. This is a problem since we want our screen to be big enough that the user can easily read a graph from. Although the resolution would be high enough in



order to display a graph, we would like the user to not need to zoom into the screen in order to see the data clearly. This will create a demand for our LCD to have its own on board memory. When researching, we find that the bigger the screen is, the higher the resolution is. It is nearly impossible to find a screen that is the perfect size for our system, with a resolution of 320x240. Most screens past 5 inches have a resolution greater than or equal to 720x480.

We ended up connecting the LCD to the PCB through SPI connection. We chose 4 wire SPI instead of 3 wire SPI. The reason for this was to allow for any other devices we would need to connect as a back up. We already planned to connect the keypad just through wires individually to 4 pins. But if this had failed, or the code for this keypad was getting to complicated, we would be able to hook it up through the SPI. In order for us to do this, we would need to have connected the LCD screen with 4 wire SPI, and not 3 wire SPI. 3 wire SPI is used for when you know you are only using one device through the SPI connection. In this mode, data will just travel directly to the microcontroller, and nothing will have to be selected for the device to know where the data should be sent. When you add the extra wire, making it 4 wire SPI, you are essentially adding in a wire that decides where the data will be headed. If we have more than 1 device connected with SPI, then this 4th wire will allow our microcontroller to decide which data goes where. In this scenario we have, the data for the LCD will go to the display, while the data for the keypad, will be transferred to the microcontroller. Once finishing this project, the implementation of the keypad just through setting up the wires to the pins, ended up working out fine. It was simple and straightforward. The code proved to be simple as well, and there was no need to set this up through the SPI connection. It is still best to always have a back up plan, which is essentially why we chose 4 wire SPI for the LCD.

## e. Software Design

### i. C Software

For the software aspect of this system, there are two levels of language. There is the higher level language, and the lower level language. The higher level languages are defined as being independent of the system itself. This means that the language C can be compiled and used on many different microcontrollers, where the lower level language, such as Assembly, will be more specific to the microcontroller. For most microcontrollers, C is the higher level language that is supported amongst them. The MSP430s are supported by a program called Code Composer Studio, which allows the programmer to write and compile their C code for the microcontroller. Code Composer Studio also allows us to write in Assembly language. This software also provides the user with various libraries. As students of UCF, we have taken courses that use this program in order to program MSP430s. This familiarity makes it easy for us to continue using the same brand of microcontrollers and the same program in order to program it.

The higher level language we will be using is C. In this code, we will program the majority of the system. Here we will need to create a graphical user interface. In order to do this easily, we will be implementing certain libraries, such as the MSP Graphics Library, to help. These libraries will contain tools for us to easily design a feasible user interface. This user interface will be displayed on the LCD screen, and can be used by either touching the screen, or by using the buttons provided. The C code will also handle programming the buttons, as well as displaying the graphs. We will be using special libraries to handle the data given to the microcontroller via the CCD. This data will then be translated into a graph and shown on the display screen. This C code will also give the USB a function to save pictures of the graph and export it as a saved file onto a flash drive. We will also be implementing the same thing done when connected to a computer via USB.

Another reason why C is our preferred choice of language to write the code for our system in is because of its compatibility. Almost all microcontrollers can be coded using the C language. This will help us in case we need to switch our microcontroller to a back up option, we will still have to change some parts of our code, but not all of it. This makes the higher level language of our code almost universal amongst other microcontrollers. Some other higher level languages include Java and Python. While our experience with Python is small, the majority of us working on this project are knowledgeable in Java. Java is an object oriented language, and is taught here at the University of Central Florida in at least two classes. One great advantage to using Java is for the user interface. Java comes with libraries that make it easy to create a user interface. Some of these functions include the JFrame and JPanel resources. With these resources, the programmer can simply call these functions that show up on the screen as a commonly seen user interface. Within these functions, the programmer can customize the interface, in order for it to fit the view model we will be attempting. While Java seems to be the best choice for higher level languages, C is still the stronger option. Most microcontrollers, especially low powered controllers, do not support these languages. The reason for this is because these languages take up a lot of memory, and their compilers are too complex to be made available for a low powered microcontroller. It is always a good idea to compare each programming language to see what each language can offer. But in this case, we will still be using C since the functionalities for microcontrollers are more available, and allow us to have a better selection of microcontrollers to choose from.

## ii. Assembly

The lower level language we will be using is Assembly. This language can be used to save space in our memory. If parts of our C code become too big for our memory to hold, then we will optimize the code, and program it for Assembly. The Assembly code can then be used for some of the buttons, as well as connecting to other devices, and assigning ports. Although it is generally easier to write certain code in C, Assembly can be more efficient in terms of memory and speed. This is where the datasheets will be helping us the most. The datasheets will provide to us the specific pins that we will need to assign in our code. We will also need the datasheets of the LCD screens in order to display our graphs and

user interface. Some of these LCDs will come with their own libraries, allowing us to use those to create the graph and display easily.

Most of the assembly code for this system will be written for the LCD. In order to correctly match the pins needed for our LCD to be drawn to, we will have to specify them in the lower level language. Since the Kentec Booster Pack we are using utilizes the MSP Graphics Library, this code may not be necessary. This option of setting the pins in order to draw to the LCD will be used if there is a problem with the booster pack. Otherwise, there will be no need for this portion of code for the LCD.

The Assembly language is taught to us students here at UCF. There are multiple classes which involve students using Code Composer Studio in order to program MSP430s using Assembly. This is helpful to us in both our writing of the code and debugging of it. We will be able to use the software, Code Composer Studio, to show us registers in use and which microcontrollers we are functioning for. The TI lab on the UCF campus will also help us in these processes. With debugging and finding out if our code works properly on the microcontroller, the TI lab offers a wide variety of tools. They have desktops with Code Composer Studio installed on them, allowing us to make changes to our code there. There are also teaching assistants within the lab that can assist us, should an error occur. They also make development kits available for us, so we can easily run our code to these kits, and quickly be able to test our program.

### iii. Display

For our LCD, given the Kentec Booster Pack, we will use the MSP Graphics Library to display information to our screen. This graphics library is a free library given from the TI website. It is compatible with the Booster Pack, and available to use on many of the MSP430 low powered microcontrollers, such as ours. This library comes with code examples, which will help us easily display our information. Using this will allow us to easily create a user interface, where a menu will be displayed. This menu will give the user options to choose from, where they can run the test, display the graph containing the data recorded, and lets the user exit, start a new test, or save the image of the graph. In order to save the image of the graph, we will allow the user to insert a flash drive via USB port. The user will then have the option to save the graph image, and give a name to the image, or save it with a default name that the system will give to it.

If we choose to work with another LCD, then this is where the code will become complicated. We will have to find an LCD which has a graphical library that is compatible to MSP430 microcontrollers. This way the library will set each pin location for us, and give us functions to use so we can draw lines and other objects to the screen. Then in our code, we will just have to call these functions, and the process to creating the images on the screen will be similar to using the MSP Graphics Library mentioned above. If we want to use a LCD screen that has an incompatible graphical library, we will have to manually code and recode some of the library. We will have to change the functions so that they

contain the correct pin connections, which will take a lot of time and extra testing. We will also have to go through each function and determine which pieces of code don't work with Assembly, and change these lines of codes with similar commands that are accepted by Assembly.

Another option for our LCD is to just grab a screen with no graphical library. The problem with this option is that we would have to manually code all of the pin connections. We would then develop code that will draw to the screen. This sounds easy, but requires us to create functions that draw to the pin, functions to draw lines, and other functions to draw different objects to the screen. Since this would take up too much of our time, our best option is to go with the Kentec Booster Pack, since it allows us to just simply connect our LCD, and then write code given their functions from the MSP Graphics Library.

The code written for the graphics library will be done using the Assembly language, if there needs to be code developed for the LCD. There will also be code changed and written in the other graphical libraries using Assembly if we cannot find a compatible graphics library for our microcontroller. The MSP Graphics Library, however, will be controlled through C. We will call the functions from this library, and draw our images by using the C language. The only problem with this will be memory. Since C will take more memory than Assembly, we will have to be careful not to use up too much space, and might have to optimize our code once it has been written so we can use the least amount of memory. Comparatively, if we had used Assembly to create our own graphical library, the code will most likely take up more space than if we had just used the MSP Graphics Library. This is because of our lack of experience with developing graphical libraries. Our software would not be optimized, and would most likely take up too much memory.

One issue to think about when talking about the software required for the system is the memory. Using the Kentec Booster Pack, we will just have to use the MSP Graphics Library, and we can easily code the images using their functions. This will require the least amount of memory for our system. If we have to develop our own library in order for the display to work, we might consider expanding our memory using an external drive to hold some of the code. Since we would have to develop the code ourselves, the code wouldn't be as optimal as the MSP Graphics Library, and would have to use a lot of memory, which our microcontroller might not be able to handle. Initially we chose a chip that gives us an excellent amount of memory, but if we are forced to choose this option of developing our own graphical library, then we might run out of space.

The Kentec QVGA display we used as our LCD proved to be very durable and easy to use. The only issue we ran into with this display, was when we called a new function of our code to draw to it, it ended up taking a couple of seconds to show the output. This didn't make our break our project, but it would have been better from the consumer's standpoint if everything popped up at the screen at the same time. It would also make all our functions look more fluid and aesthetically pleasing. The price remained \$24.99, but thanks to the TI lab on campus, we were able to get it for free.

In order to draw to the screen, we had to map the whole LCD. We had to find coordinates by their pixels, and set up the rectangles and text according to the pixels. This was not difficult to do, but proved to be time consuming, and included a lot of trial and error. The way around this would be to just use the example projects given by Code Composer Studio. But we chose not to do this, since we needed to draw a graph anyways, and the menu screens we planned on creating looked different than anything seen in the example projects. We were also able to test if the 3 wire SPI worked through the launchpad given to us to use from TI. We simply connected the corresponding wires we were planning on using into the ports for LCD, and were able to see that it worked. The only real changes we needed to make were in the initialization files that the LCD came with. We needed to change them from USCI B to A, since our PCB set everything in USCI A. We also needed to change the pin settings since we were moving from the MSP430F5529 used on the launchpad to the MSP430F5528, which is the chip we are using on the PCB. This is explained in more detail inside of the Software Planning section of this paper – Section 7. On the same topic as the code, we decided to run it with less functions than originally planned for. We did this to save on code, and to increase the time it took draw to the screen. This time only differed by a couple of milliseconds in the end, but every second counts, and can make a big difference in seeing the screen act fluidly. In order to test the main screen, we ended up just enabling the touch features of the screen. This was just to see if we could really move back and forth from different screens. Once tested and completed, we ended up removing this, and implementing the keypad. The keypad was used since it was preferred by the people in the labs, and since it took up less memory than using the touch features of this LCD.

## f. Power Supply

The power supply is one of the most important parts in a project; most power supplies are developed to convert alternating current (AC) to direct current (DC). The most common circuit configuration in electronics is AC to DC power. Electric power distribution systems mainly provide AC power, but most devices operate under DC power. A few examples of AC power would be big power plants that distribute electricity to homes, the power supplied by power outlets, washer, refrigerator, and etc. As for DC powered supplies would be cell phone, flashlights, laptop, and etc.

There are many circuits constructed to convert AC to DC power, some are very easy to design as for other circuits are complex and safer for projects. For example, there is the half wave circuit shown in Figure which is developed of one transformer, one diode in series with a capacitor. This circuit only allows for one-half wave of DC to flow through the load. This circuit is not very convenient for many applications. The harmonic content of the circuit's waveform is extremely large and difficult to filter. In addition, this design only provides power to the load every half wave every full cycle.

One mistake usually made when constructing the half wave rectifier is the diodes repetitive peak reverse voltage rating is overlooked. The rating voltage of the diode must be greater than the peak voltage of the output voltage produced by the transformer. If the diode's peak

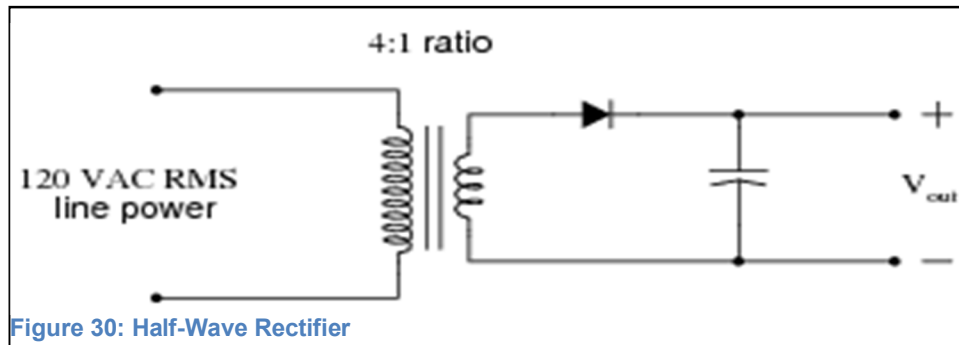


Figure 30: Half-Wave Rectifier

inverse voltage (PIV) rating is insufficient, the diode will not be able to withstand the demands of this circuit. In order to calculate the PIV rating for the circuit's diode, you will have to analyze the peak output voltage from the transformer's secondary windings. Measure each half-cycle of the AC waveform, realizing the voltage loss across each component. After completing the full cycle analysis, the necessary diode result should be obtained.

Another example of a power supply circuit and its capability is transformer-less direct rectification of AC to DC power, as shown in Figure. This design is only constructed of a switch, fuse, and four diodes. The reason for this AC to DC conversion not having a transformer is to save in costs. The transformer is a well needed component for a power supply circuit. Not only does the transformer provide a ratio of voltage and current, it also provides electrical separation between AC and DC circuit lines. This issue becomes very dangerous for operators/technicians to work on.

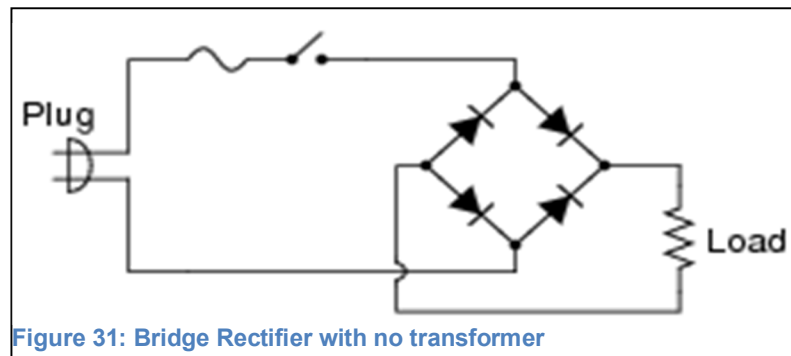
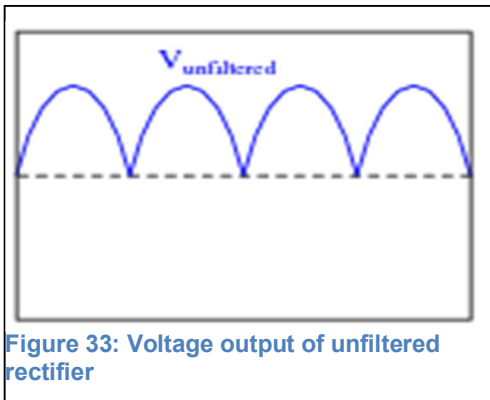
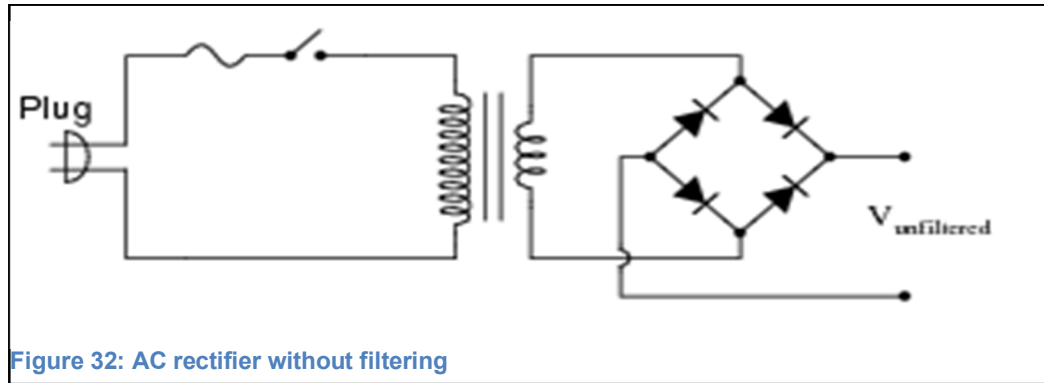


Figure 31: Bridge Rectifier with no transformer

It is very important to have a filter implemented into power supply circuits. The filter helps eliminate the ripple AC voltage from the DC voltage output. By eliminating the ripple in the circuit, the ripple can damage components or your final experimental results. There are multiple types of filters, such as low pass, high pass, band pass, or band stop. The filter

needed to stop the ripple voltage is the low pass filter. The parameters needed to determine the ripple frequency for linear power supplies are the line frequency and rectification pulses. Figure shows an unfiltered full-wave rectifier, while Figure shows the expected output of the circuit. The modified circuit is shown in Figure, with a lower voltage ripple as shown in Figure.



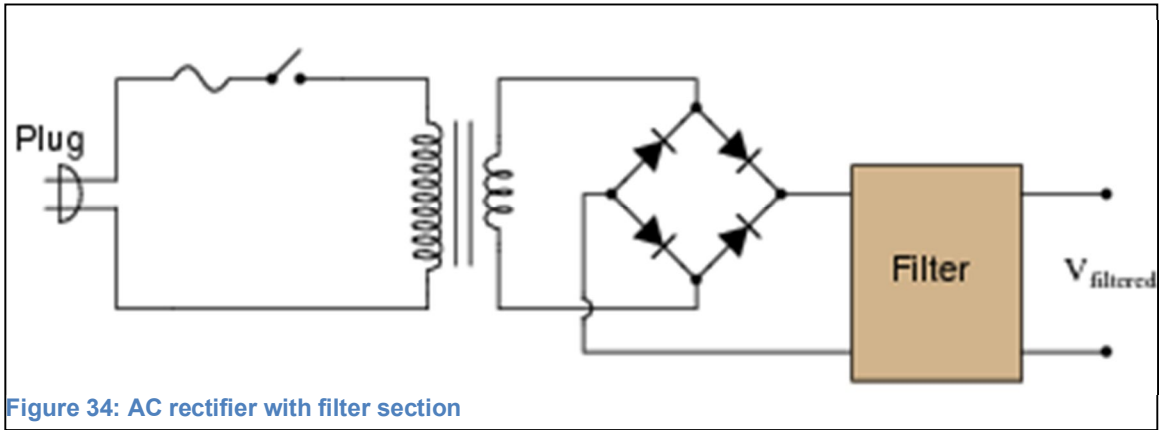
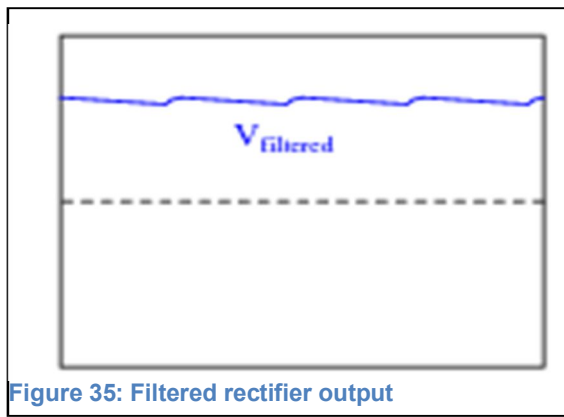
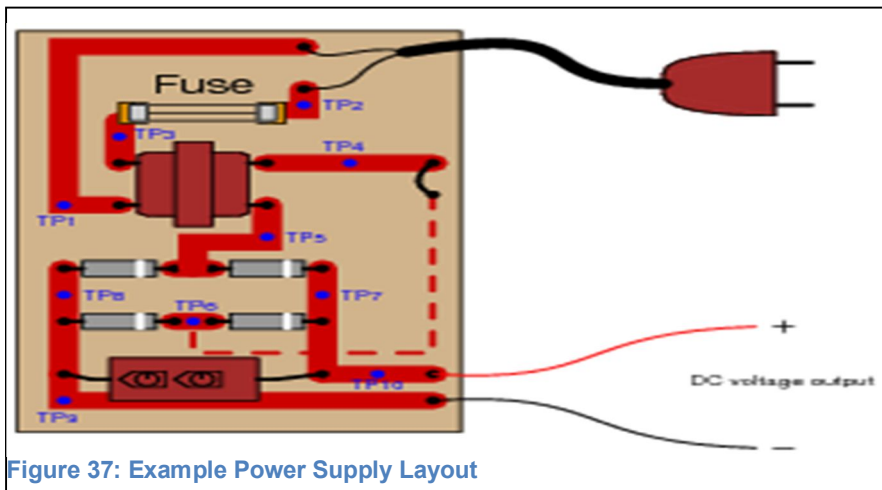
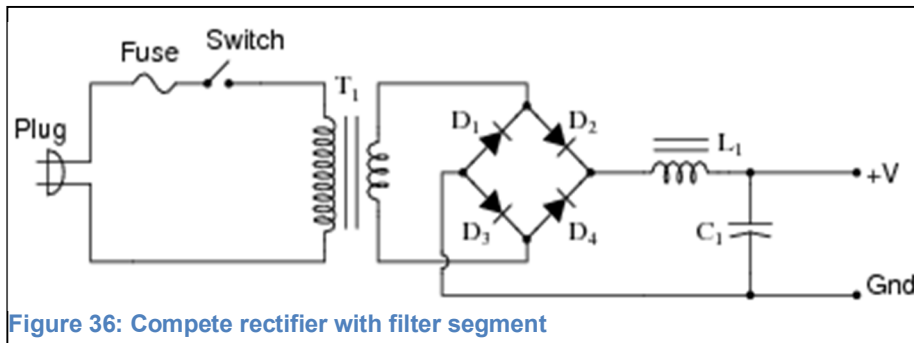


Figure 34: AC rectifier with filter section



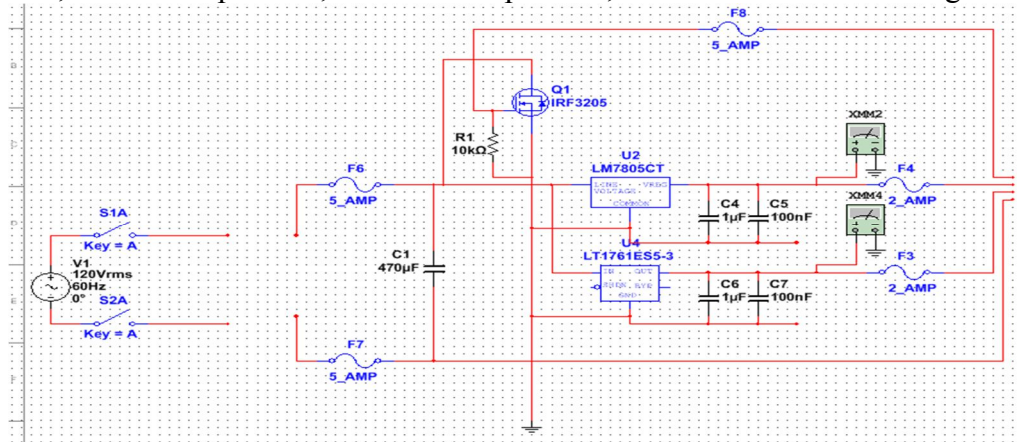


The final chosen design is shown in Figure. Part values are derived in the part selection area.



## Final Power Supply:

The final power supply constructed and used in the design is as it is shown below. The most difficult part of the power supply is giving us the right amount of watts needed for the devices in the design. In order to achieve the amount of power needed for the design we needed to take into account how much power the transformer was liable of providing. By doing so the transformer originally purchased was only providing 12 watts of power, when the design needed 60 watts. The transformer was replaced with a Quans transformer as shown below in the image below. The Quans transformer provided the design with either a -12V or +12V supply, only needing to add a +5V regulator and +3.3V regulator the final schematic of the power supply. By adding the Quans transformer, it eliminated fuses, regulators, resistors, and capacitors; eventually reducing the price of the final power supply schematic. The price reduction to the power supply was nearly nine dollars. The picture below is the final power supply schematic, consisting of +12V, +5V regulator, +3.3V regulator, two 10 F capacitors, two 100nF capacitors, and an IRF 3205 switching MOSFET.



Final Power Supply Schematic: Figure above.



Quans Transformer: The figure above was the transformer used in our power supply design.

## Design Protocol

The power supply of the project will be constructed by following a few basic steps as the following:

- I. Cautiously measure and record all component values before constructing the circuit. For best results, choose high values for resistors (anywhere from 1k to 100k).
- II. Draw and label all parts for the schematic diagram.
- III. Now begin to build the circuit on the breadboard.
- IV. Check for continuity on the circuit's construction. Each wire should lead to its destination, verifying each location by looking at the schematic.
- V. Using a multi-meter make sure all voltages and currents are correct compared to the analytical answers.
- VI. If everything is correct, now cut and solder components to breadboard.
- VII. If there are any significant errors (larger than 5 percent), cautiously check your circuit to make sure there is enough separation between components. Cautiously measure voltages and currents again and compare results.

Table below shows the full part list for power supply construction.

Quantity	Components	Information
1	Vero board	
4	Diodes	1N4001
1-2	Regulator(s)	LM7812
1	Transformer	5v - 35v
1	Capacitor	1000uF - 4700uF
1	Capacitor	1uF
2	Capacitors	100nF
1	Jumper Wires	
1	Heatsink	
1	Fan	Climate control
2	Fuses	
1	Filter	Low Pass
1	LED	Red
1	Resistor	1k ohm
1	Power Switch	

Table 24: Material requirements of power supply

All the components will be test individually for accuracy. Extras will be purchased in case of malfunctions or failure due to human errors. As for the construction, the four diodes will be combined to make a bridge rectifier, which prevents the circuit from only receiving half-wave voltage, but instead suppling it with full wave. The regulator will keep the voltage at

a constant 12-volt output and if a 5-volt output is needed just replace the LM7812 regulator with a LM7805 voltage regulator. The transformer will provide 5-35 volts depending on the amount of voltage the design needs, Jumper wires will be implemented to connect the power supply to the rest of the project. Heat sink will protect from electric shock and from components getting to close together and causing an outage. The fan will provide cooling the entire system. While fuse one will protect the high side voltage (AC side) and fuse two to will protect the low side voltage (DC side), that way nothing in the gets burned up. The power supply will also have a low pass filter to help get rid of all the ripples from the AC voltage waveforms. Some of the most common filters that will be needed is the LC ripple or the pi filters. An LED light will be needed to signify weather the power supply is on or off. The resistor is needed to help aid the LED, and the power switch will be turning on or off the power for the design.

The final produced schematic is shown in Figure. Test values are shown in Multisim to assess the quality of the network.

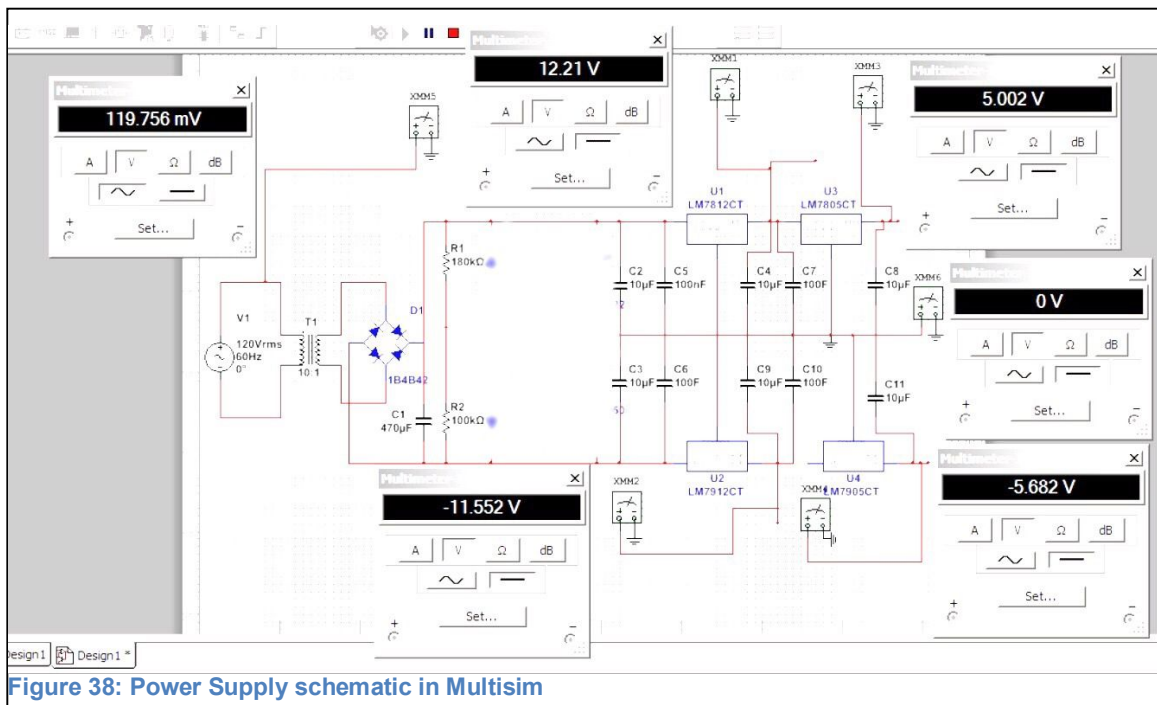


Figure 38: Power Supply schematic in Multisim

## h. Summary of Design

The final schematic is presented in the prototype design section. This design, though aimed at presenting a stand-alone device, utilizes mostly 3.3 and small footprint components. This design could easily be adapted to hand-held systems that interface with a computer for data readout. The higher voltage light source could be powered with a buck converter for burst lighting, or even just replaced with a window for sunlight.

The limiting factors in the spectral resolution are predominantly the optics, which can quickly scale the price of the product up for small improvements. The selected components were deemed acceptable for this application, though the UV response is somewhat sub-optimal for the desired application. Financing the device is discussed in the administration section.

Modularity became an important consideration for this project since many components must be spread out around the enclosure. The optics in particular require great precision in mounting. Using a number of cables and discrete connections will improve the placement of these components with good cable management. This is discussed further in the final schematic and PCB layout sections.

The decision was made to separate the power and computer boards to improve the grounding of high current elements from the sensitive ICs used for signal processing and control. This also improves the modularity of the final design layout at the cost of a larger enclosure. Estimates still put our project healthily under the limitations defined in the initial product pitch so this was not deemed a significant downside.

## 7. Prototype Design

### a. Full Schematic

The first draft of our final schematic is presented in Figure and Figure, utilizing two separate board structures for better ground isolation and modularity. This schematic was constructed using Eagle to facilitate easy PCB layout once the design is tested and satisfactory. The design utilizes several cables for improved modularity and easy interface in the event that upgrades or hardware failures require device replacement. It also improves the ability to mount the components ideally in the enclosure and ensure good optical resolution.

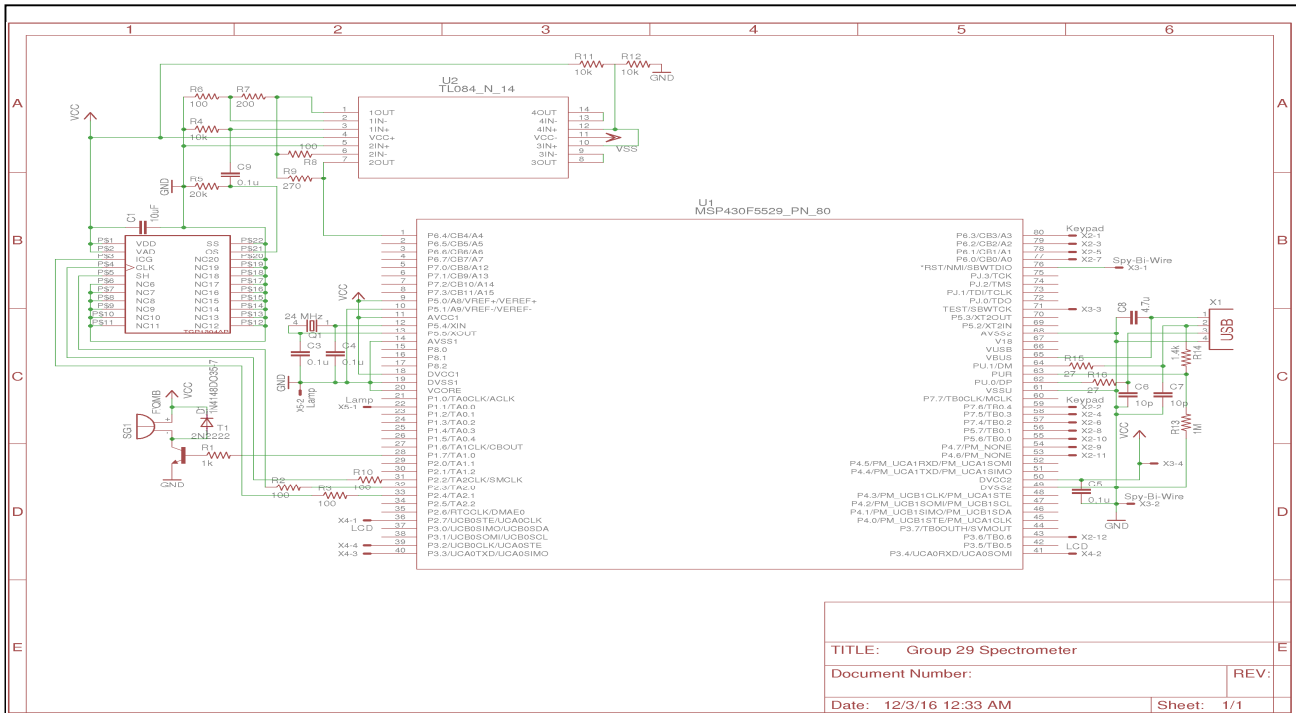


Figure 39: Complete Computing board schematic. First draft.

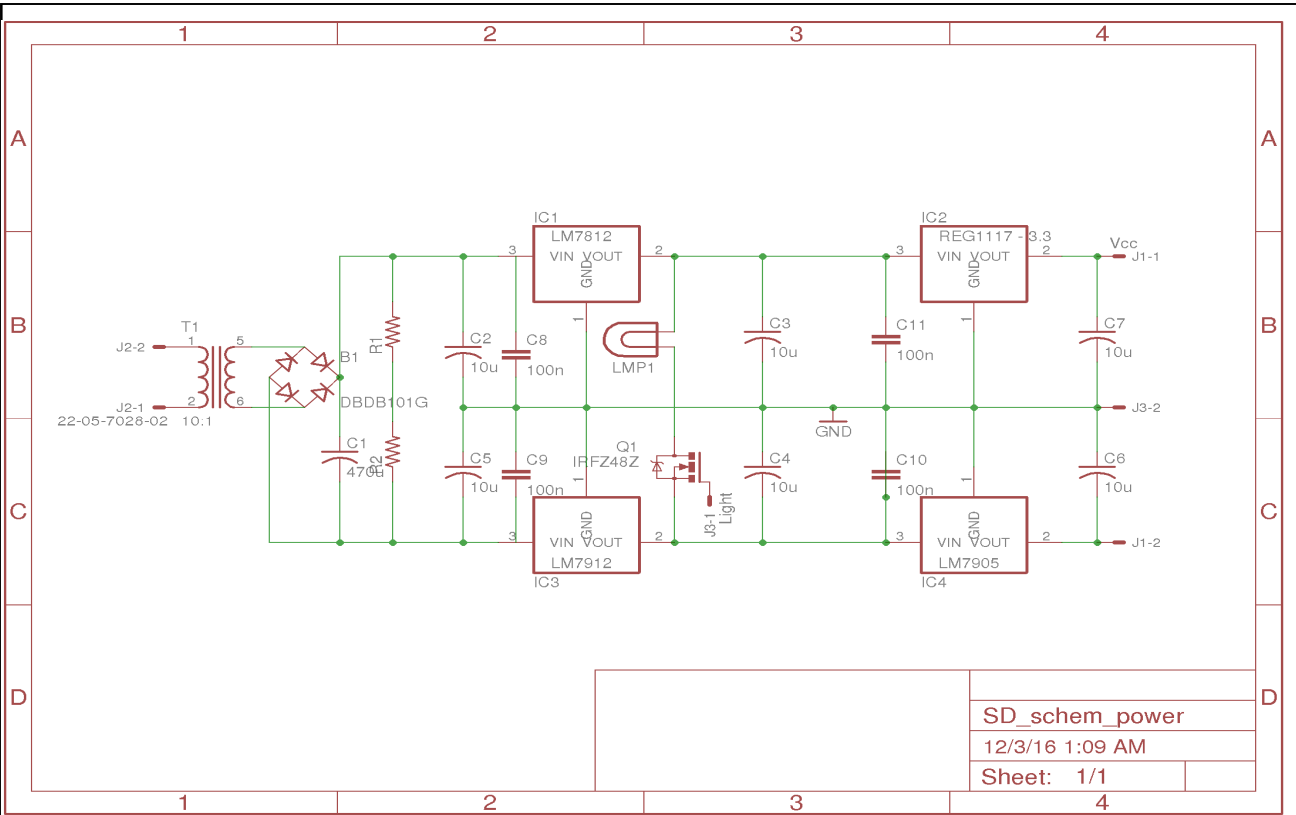


Figure 40: Complete power supply schematic. First draft.

The schematics will form the basis of the initial PCB designs for this project, though focus will initially be placed on constructing the optics and testing the final software using the development environment. A JTAG interface is included in the board to allow for hot prototyping and memory flashing without assembling a new board or attempting to remove the surface-mount microcontroller.

The schematic utilizes a large number of filter capacitors to improve power delivery to the ICs before the power supply can respond. Due to the low voltages of most of these devices, exhaustive regulation was not deemed necessary, though this may be a consideration during testing if transient errors are noticed during high load.

The two boards will be tested individually before introducing the sensitive computing components into the power supply. This separation also allows for a more robust ground plane with less possibility to experience ground spikes through the IC ground leads. Ground plane considerations will be heavily considered during layout.

The BoMs in Table and Table were generated from these schematics.

**Table 25: Computing BoM.**

Qty	Value	Device	Package	Parts	Description
1		PN61729	PN61729	X1	BERG USB connector
1		TCP1304AP	DIP	U3	Toshiba CCD Linear Image Sensor
1	0,1u	C2.5/5	C2.5-5	C5	CAPACITOR
1	0.1u	C22/6	C22.5B6	C9	CAPACITOR
2	0.1u	C5/4.5	C5B4.5	C3, C4	CAPACITOR
1	1.4k	R-US_0204/7	0204/7	R14	RESISTOR, American symbol
2	100	R-US_0204/7	0204/7	R2, R3	RESISTOR, American symbol
2	100	R-US_0309/10	0309/10	R6, R8	RESISTOR, American symbol
1	100	R-US_0309/12	0309/12	R10	RESISTOR, American symbol
2	10k	R-US_0204/5	0204/5	R11, R12	RESISTOR, American symbol
1	10k	R-US_0309/10	0309/10	R4	RESISTOR, American symbol
2	10p	C2.5/5	C2.5-5	C6, C7	CAPACITOR
1	10uF	C5/3	C5B3	C1	CAPACITOR
1	1M	R-US_0204/7	0204/7	R13	RESISTOR, American symbol
1	1N4148DO35-7	1N4148DO35-7	DO35-7	D1	DIODE
1	1k	R-US_0204/7	0204/7	R1	RESISTOR, American symbol
1	200	R-US_0309/10	0309/10	R7	RESISTOR, American symbol
1	20k	R-US_0309/10	0309/10	R5	RESISTOR, American symbol
1	24 MHz	86SMX	86SMX	Q1	CRYSTAL
2	27	R-US_0204/7	0204/7	R15, R16	RESISTOR, American symbol
1	270	R-US_0309/10	0309/10	R9	RESISTOR, American symbol
1	2N2222	2N2222	TO18	T1	NPN TRANSISTOR
1	4.7u	C2.5/5	C2.5-5	C8	CAPACITOR
1	F/QMB	F/QMB	F/QMB	SG1	BUZZER Source: Buerklin
1	Keypad	332-12	332-12	X2	12 Pin - 2mm Dual Row
1	LCD	332-04	332-04	X4	4 Pin - 2mm Dual Row
1	Lamp	332-02	332-02	X5	2 Pin - 2mm Dual Row
1	MSP430F5529_PN_80	MSP430F5529_PN_80	PN80	U1	
1	Spy-Bi-Wire	332-04	332-04	X3	4 Pin - 2mm Dual Row
1	TL084_N_14	TL084_N_14	N14	U2	

**Table 26: Power BoM**

Qty	Value	Device	Parts	Description
1		LAMP	LMP1	LED HOLDER
2		R-US_0817/22	R1, R2	RESISTOR, American symbol
4	100n	C5/2.5	C8, C9, C10, C11	CAPACITOR
1	10:1	XFORMER	T1	TRANSFORMER
6	10u	C-US050-024X044	C2, C3, C4, C5, C6, C7	CAPACITOR, American symbol
1	22-05-7028-02	22-05-7028-02	J2	CONNECTOR
1	470u	C-US050-024X044	C1	CAPACITOR, American symbol
1	DBDB101G	DBDB101G	B1	Single Phase 1.0 AMP Glass Passivated Bridge Rectifier
1	IRFZ48Z	IRFZ48Z	Q1	N Channel HEXFET® Power MOSFET Logic Level
1	LM7812	REG1117F	IC1	800mA and 1A Low Dropout (LDO) Positive Regulator
1	LM7905	REG1117F	IC4	800mA and 1A Low Dropout (LDO) Positive Regulator
1	LM7912	REG1117F	IC3	800mA and 1A Low Dropout (LDO) Positive Regulator
1	Light	22-05-7028-02	J3	CONNECTOR
1	REG1117 - 3.3	REG1117F	IC2	800mA and 1A Low Dropout (LDO) Positive Regulator
1	Vcc	22-05-7028-02	J1	CONNECTOR

## b. PCB Assembly and Vendor Selection

The PCB will be attempt to be designed using eagle cad to build and construct the layout before printing it. Procedures are as follows down below. If attempt fails, we will purchase a PCB from the following website: <http://www.elecrow.com/services/pcb-prototyping.html>

The website offers great services and offers; it provides easy ways to receive 5, 10, and 20 pieces of PCBs for testing and prototyping. Some of the services are as follows:

- i. Elecrow provides 2 layers and 4 layers PCB prototyping, in different sizes.
- ii. Multiply colored PCBs without extra charge
- iii. The LED is free no extra charge is needed
- iv. PCB insurance, all PCBs are tested
- v. If the PCB is damaged, there is a replacement free of charge.

The following list shows a few steps on how we would construct our own printed circuit board, though the vendor service is likely a more robust option.

- i. Download software ( Eagle Cad)
- ii. Design schematic for PCB
- iii. Convert file to PDF
- iv. Print on glossy paper and make sure it's a laser printer
- v. Toner must be on max (very dark)
- vi. Purchase copper plate for PCB
- vii. Paste printed paper on copper plate
- viii. Dip in water for 10 minutes
- ix. Remove glossy paper



- x. Put it in Ferric Chloride for 15 minutes
- xi. Rinse in water
- xii. Then let it sit in water with baking soda for 5 minutes
- xiii. Final step is to remove left over ink using nail polish remover

### c. Firmware Coding Plan

The firmware represents all of the hardware-level manipulations that must take place to provide the high-level software with data to format and present to the user. The specifics of timing and pin connections have been covered in other parts of this document, but the collection of functions that must be produced and integrated on the microcontroller will be addressed in this section.

The most complex sub-routines regard taking in the spectral data from the sensor and interpreting it as a spectrum instead of a large group of data points. The first important function is the readout function for the CCD, which forms the basis of all other methods. This function will accept no arguments and returns an array containing all samples taken from the ADC during the pixel readout. Most likely this array will overwrite the previously read array in the memory, as these raw values are only needed to produce the calibrated data.

A closely connected function is the binning function, which will iterate through the raw data and average the values of each predetermined set of pixels to produce the binned pixel, with larger average weight on the middle pixel that most closely corresponds to the required wavelength. This function could require no arguments but should store its result array separately from the raw data and from other spectrums. The binning function reduces the impact of the photon shot noise that forms one of the major limiting factors of the data integrity. It also smooths out any read noise that may remain in the signal.

The selected sensor has 3648 functional pixels, while the spectral range that will be presented to it has 600 discrete nanometer values. This means each pixel will account for approximately 164 pm of granularity. To match the 1.8 nm spectral resolution of other spectrometers on the market, up to 11 pixels can be averaged per individual wavelength selection. The binning function will accept a variable array as an input, since the smaller spectrum size will accommodate storing more than one spectrum. This will allow comparisons for spectrum blanking and allow the high-end software to run scans while interpreting a previous scan for graphing or direct comparison.

A sub-binning for-loop block will form the basis of this binning scheme, which will select the center pixel for a wavelength, average the adjacent 11 pixels, and return a single wavelength value for the output table. This function is subject to calibration, and potentially even a further exploration of digital signal processing to produce an optimal averaging scheme. For the first prototype drafts, a simple 90% confidence interval on the standard bell curve will be overlaid on the 11 values to produce the average value. The

coefficient's can be hard-coded into the function or stored in a small table for easy adjustment.

Once the values have been collected, the resulting binned list can be multiplied by a new function to adjust for quantum efficiency calibration. Since the detector has varying quantum efficiency but static integration time, the values of pixels corresponding to less reactive wavelengths will be limited by a proportion of this efficiency element. A look-up table can be stored in the memory of the MCU containing coefficients for quantum efficiency as displayed on the data sheet for the selected sensor. In a practical build, this curve could be precisely calibrated using an automated assembly tester that displays a known steady-state spectrum and returns the saturation values for each pixel. For this experiment, the look-up table will simply contain data sheet values that can be corrected as necessary. The proof-of-concept prototype will still function adequately.

One last adjustment is a dark current calibration function, which subtracts a static dark current value from each pixel. Since both the integration time and the dark current through the sensor are not controllable and should not vary significantly during a single scan, this can be a single value that is calibrated by covering the scanning input and reporting the resulting full-sensor pixel average. This last function finally presents the best quality data that can be derived from the known characteristics of the sensor, and is ready for software organization into tables and figures.

In practice, many scans will only request one value to be displayed predominantly. A carrier function will be made for this purpose to sift through the resulting calibrated array and report the final value. This function will also accept a register as an input to store the final value and improve readout time if called additional times by the software.

Other assorted hardware functions of this device include a function to simply activate the light source during a scan. Since the integrator of the sensor is cleared and then stored in a shift register, there is not a strict timing requirement for this function. Most likely, this function will simply turn the required GPIO pin on for a static time value, which will begin as 15 ms for a 5 ms timing buffer for clearing the CCD and beginning integration. If this value is found to produce too much heat for the system specs, it can be trimmed as necessary during debugging.

The buzzer will utilize a hardware timer to produce the PWM waveform to drive the inductive element. The function will accept a sounding time and a volume adjustment as input and return nothing. The timers on the MSP430 line are clocked separately from the processor itself, and utilize one register that continuously counts up, one register that stores an end point and a third register that stores a value. The timer can be configured to adjust a pin to high or low depending on the timer function, which in this case will toggle a pin on during timer restart and off when it reaches the stored midpoint value. The counter in mode 07 of the data sheet will then continue counting to the maximum in the second register and then restart. This produces a waveform with frequency controlled by the cap

register and duty cycle controlled by the midpoint register. The volume level will set the proportion of this midpoint to reduce the sounding volume of the buzzer.

Dealing with the user interface and USB communication can be done in high-level software, and the power is controlled by a simple switch on the supply board, so no further firmware coding is necessary.

All necessary calibration arrays will be stored in standalone database files and linked to the main project to allow for easy generation and adjustment without editing the entire source.

To recap, here are the expected firmware prototype functions:

```
function ReadOut () { return int *RAWDATA[];}
// RAWDATA size is current 3648

function BinOut ( int *SPEC1[] ) { return int *SPECTRUM[]; }
// SPECTRUM size is 600

function Quantize ( int *SPECTRUM[] ) {
    for (p = 0; p < SPECTRUM - 1; p++) {
        SPECTRUM[p] *= QUANTIZE[p];
    }
    return;
}
// Adjusts SPECTRUM[] data with pre-defined values

function DarkSub ( int *SPECTRUM[] ) {
    for (p = 0; p < SPECTRUM - 1; p++) {
        SPECTRUM[p] *= DarkData[p];
    }
    return;
}
// Adjusts SPECTRUM[] data with calibrated values

function DarkCal () {
    ReadOut();
    BinOut();
    Quantize(int *DARKSPEC[])
    for (p = 0; p < SPECTRUM - 1; p++) {
        DarkData[p] = DARKSPEC[p]
    }
    return;
}
// Block light input, generate dark current data
```

```

function LightOn () { return; }
// Flips on the light for 11-15 ms, turns light off with an interrupt to not affect scanning

function FullScan ( int *SPEC1[]) {
    LightOn();
    Readout();
    SPEC1 = Binout();
    Quantize(SPEC1);
    DarkSub(SPEC1);
    ConvAbs(SPEC1);
    return;
}
// Convenient way to generate a calibrated spectrum with one call.

function GetWavelength ( int *SPEC1[], int WL) { return int Abs }
// Simple front-end to report a single absorbance value from a given spectrum

function Buzzer ( int TIME, int VOLUME) { return; }
// Sounds the buzzer at 4 khz with the selected volume for the selected length of time

```

#### d. Software Coding Plan

The simplest way to structure our code is with the use of functions, as seen in Figure. Each piece of code has its main function, where everything is called. The other functions in the code are called from the main function, and allow the programmer to easily find what each piece of code is doing. The reason this is helpful for us is because the functions will explicitly say what they do. So instead of us having to look through the main function of our code and finding what each line of code does, we will be able to just look through the functions. This also helps for readability purposes, as well as debugging. The code will be much neater and easier to read through.

```

int main()
{
    function1();
    function2();
    return 0;
}

//Function1 will draw the UI to the LCD
void function1()
{
    //Code here to draw to the LCD
}

//Function2 will collect the data
void function2()
{
    //Code here to collect the data
}

```

Figure 40: Namespace and prototyping of software design

For debugging, this will allow the programmer to find mistakes easily. If there is a problem with the code, instead of going through what we wrote line by line, we can figure out what the problem is, and associate that problem to its corresponding function. We can then fix the problem within the function, without having to change a lot of the other code. Had we solved this problem by checking each line of code, it would take a large amount of time, and we would most likely have to change a lot more code than just a couple lines in a function. There will be a good amount of code written in the main function, just so we don't run into problems with the compatibility of functions. This means we will try to avoid having problems with functions interacting with other functions by keeping the number of functions to a good amount. Each function will have a comment attached to it explaining its purpose, and what other functions it interacts with. We will also include comments on pieces of code within the functions where there is complexity. This way we won't have to comment every single line of code, since some of it will be self-explanatory, but we will have enough comments to be able to easily read through our code and tell what each function is doing.

The specific functions we are going to have in our software will be for each different component of our system. We will have a function that displays the user interface. There will be a function for the data received from the CCD. This function will read in the data, and configure it into a graph that can be displayed on the screen. There will be a function for saving the image to a flash drive through a USB port, mentioned below using the MSP430 USB software, and a function for user options. These options will include a back button, a button to perform the test, a power-off button, and a button to zoom in or out of the graph. There will also be functions to choose how to display the data. This includes the exact numbers for how the data on the graph is displayed, and the parameters the user wants to be displayed. If not specified, there will be a default set of numbers and parameters displayed based on the data received by the system.

In order for our system to utilize the USB port, we will be using MSP430 USB software. We will download the Descriptor Tool from TI, and this will allow us to develop our USB interface. This tool will allow us to view the MSP430 USB API, which is a software library provided from TI, specifically for the use of USBs. The most notable feature about this API is the use of coding examples. The main example we will be using will be the MSC examples. These examples will give us a means to store data onto a flash drive through the USB. There are many other examples, ranging from allowing the user to implement an external device such as a keyboard, to being able to interface with a computer. The clock for this USB can be controlled by a crystal at 32 kHz. The clock can also be powered from the clock used by the actual CPU. The latter will be the easiest way to provide a clock signal to the USB. This is because we won't have to rely on a crystal, and can instead use the clock known to be working for the CPU.

Another feature in our coding plan involves dealing with variable names. As seen in a lot of short code, simple names such as "x" or "y" are given to variables. This will not be acceptable for a large amount of our code. Since each function and process in our program will cater to specific needs, there will need to be specific names given to each variable. For loops and simple local variable, "i" can be used, as well as "j," "k," and so on for subsequent loops within the first loop. Local variables within functions will be named explicitly to what they are referring to, and the same rule will be applied to function parameters. For example, if we are drawing a picture with our code, and we need to call two coordinates, x and y, then the variable names for these coordinates will be "coordinatex" and "coordinatey" respectively. Abbreviations will only be used with variables that seen in almost every function, and should not be utilized for every variable. Abbreviations, while helpful to the programmer, can cause some confusion when reading through the code. These abbreviations will also need to have extra comments along with them, creating more documentation which may not be necessary if the variable is only used once. These abbreviations will be mostly seen for names that are too long. We don't want to fill our code with useless names, since calling these variables in equations or functions will end up taking up too much space. Variable names with 15 or more characters in the name will be considered for an abbreviation, and will be commented with a description pertaining to it.

When discussing variable in code, there are many different types. Two of these types are local and global. Local variables are variables only used within a specific function, and outside of that function they hold no value. This means you can use a variable named "x" in one function, and have the same variable named "x" in another function, without them conflicting. The rules we implemented for these variables are discussed above. Global variables however are the opposite of local variables. They are defined outside of functions, and their data can be seen by each function. If one function changes the data of a global variable, the data is permanently changed for the rest of the code that uses this variable. Global variables are less commonly needed and used than local variables, therefore there won't be too many of them in our code. Since there will be a few number of global variables in our program, we will start by naming each variable with the prefix "g\_." For

example, if we have a variable named “counter,” and we are making a global variable, we will name this variable “g\_counter.” This task of putting the “g\_” in front of each global variable might seem tedious, but since there aren’t expected to be too many of them, this task will be easy. This also allows us to easily know when we are changing the value of a global variable, since the importance of this variable is greater than that of a local variable.

During the process of making this device, we found the code to be straightforward and simple, just as we had planned for it from the beginning. Code Composer Studio was still used, and we used the graphics library from TI called MSP430Ware. TI had many example projects listed, that helped show us how to start our design correctly. The code consisted of a main function, and in the main file, we included the majority of the other functions. Pin settings and calls for the LCD to given to us in different files. We needed to change a lot of the settings. For one, we needed to switch the pins that connected the LCD through spi, since our launchpad was for a MSP430F5529, and the pins on the MSP430F5528 are different. We were just able to look through the data sheet and correct these. For our PCB, we used USCIA for the connection, and the launchpad used USCIB for its connection. So we needed to also change the connection from USCIB to USCIA. This was done fairly easy, since there was a lot of documentation on how to set these, and the correct values on how to define them. The main difference between USCIA and B is that B contains just SPI and I2C connections. USCIA contains UART, 2 other calls we did not intend to use, and SPI. We chose to go with the USCIA since we weren’t going to use I2C, and there could be a chance of us using UART in the future, in case something had gone wrong, or we thought there was another possibility of configuring the system.

USCIA and USCIB can contain two or more modules within them. For example: for A, there is A0 and A1, and for B, there is B0 and B1. We chose A0 for the USCIA module. There was no need for a different number of A modules for this project, but in planning for backups, we decided to just save another module in the event of the first one not working. This is what we considered to be our backup plan for anything going wrong with the first module. If we couldn’t get the first module to work, we would then try switching over to the second module, and continue testing from there. The main problems that would have arose from these modules stem from calling them in the code. Inside the code, where you call these functions, you must first set up the USCIA. Within this process of setting up, you must also tell the system which module you are using, and complete filling in the other settings required. This was all specified within the user manuals provided for us by the TI website, corresponding to the MSP430F55xx series. There are certain differences between USCIA and USCIB, but they did not factor in as much as we initially thought they would. These differences were just involved in the calling of these modes. You were not allowed to use the same interrupt vectors for both of these modules in SPI mode, and each time you called it, the system needed the different specific settings to be changed. There needed to be multiple lines of code changed and adjusted, just changing the USCIB to USCIA was not enough, and would just result in your device not working properly. This was one error we encountered. This error stemmed from the fact that if you initialized everything for USCIA for example, but did not set some settings correctly, instead of Code Composer

Studio giving you an error since it can't find these specific settings in your system, it will just allow the code to run, but your device won't actually be working. This puzzled us at first, since we would run the code, yet our device would just not be working correctly. After switching around to different modules, and testing the SPI on an oscilloscope, we determined it had to be with how we were setting up the initializations for the USCIA mode in the code. After finding where the root of our problems were, we were quickly able to fix this problem, and get our LCD screen up and running.

In the main file, we included the rest of the pin settings. We had a timer initiation function, which set up all the correct timers, clocks, and set the pins. We originally had two functions for this start up, and would call both at the beginning. The timers were initialized in one function, the pins set in the other. We realized that we would be able to save on memory usage if we just set up everything together in one function. Now it's not feasible to just run the program all in one function, so we ended up using multiple functions. The keypad worked through interrupts. We would set timer A to listen and wait for a keypad button to be pressed. Then once it is called, we set up an interrupt service routine so that once the key is pressed, the code inside the routine will be called. Inside the routine, it would save the row of the button pressed, and then search for which column it is in. Once it got the column, it would search for the corresponding value in a 2D array. We initialized a 2D array in the beginning as a global variable, and it searched through the array, using both the row and column, this way it could find which button was pressed. The values in this array were all ASCII characters, and we would then run if loops inside the main function in order to see which button was pressed. These if loops were all contained inside a do while loop, which stated that if a specific button was pressed (the back button) it would exit that while loop, and start it up again. This made it so that if the back button was pressed, the user would just go back to the main menu, and it would act as if they were back at the home screen. We did not implement a button to go forward, so the user has to act with caution when pressing the back button, if they are already viewing precious data that they don't want to lose.

For the CCD sensor, we also set everything up inside the initialization function called at the beginning, and called it through interrupt timers. We used the ADC and timer A1. When the function was called, timer A1 would be interrupted and go to the interrupt service routine. Inside here, the code would then call the ADC, and the ADC would take in the picture, and read it out to an array with values inside it. Our biggest struggle with the CCD was getting the right output values. Many times we would end up seeing junk values, just because the timing wasn't correct. But we soon found out this was a problem with our delays. Once we fixed our delays, by increasing some at the start, and removing some at the end, we were able to capture the correct values. This turned out to be our most time consuming problem. At first we thought it was how we were calling everything, even the function itself. We had to keep checking the pins in order to make sure they were all called and set correctly. The datasheet for the CCD wasn't as detailed as others, and wasn't the best, but in the end helped us find our problem, along with some videos on the same subject.





## 8. Testing and Verification

### a. Hardware Environment

The figure in Figure below shows the components purchased in order to build the project design. Items in the plastic bags hold capacitors of different sizes, transistors, diodes, voltage regulators, transformer, inductors, and resistors all of varies sizes. In plastic enclosed containers there are analog devices, mirrors, and a light bulb. As for the pcb in the picture will not be used, it will be used for parts in order to build the power supply for the spectrophotometer. Components that will be DE soldered will be transformer, capacitors, heat sink, voltage regulator, and fuses. The black box contains the MSP430F5529 launch pad. Each device is a key component to building the project design.

Figure below shows the equipment used for testing every component purchased. The results obtained after testing passed. On the top left of the figure shows the oscilloscope and function generator, which are used to measure the voltage output and input through wavelength and frequency. The equipment on the top right are multi-meters that test for voltage, current, and continuity. The computer has a software called multisim, which is used to build our circuit design and used for simulation.



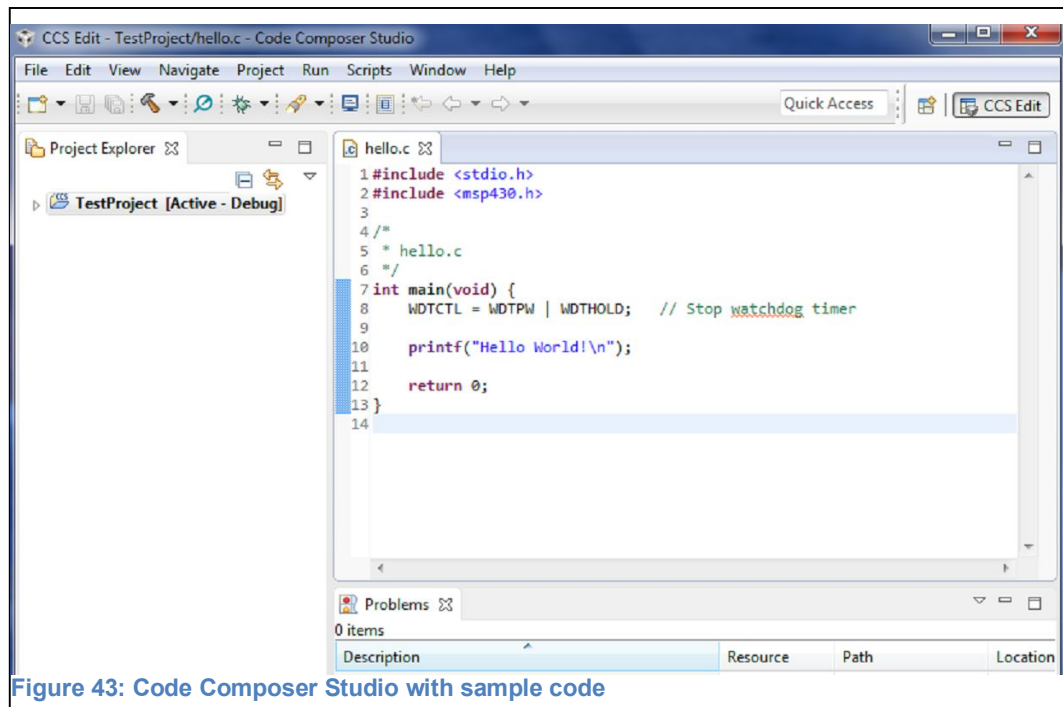
The major requirements for testing the subsystems and prototype hardware include the following:

- Function generator capable of up to 24 MHz PWM output
- 2 channel oscilloscope
- Steady-state power supply with up to 24 V amplitude output
- Digital Multi-meter with 3 digits of accuracy
- Computer with Multisim and LTspice circuit simulation

Fortunately, the senior design lab provides all of these materials and will suffice for hardware level testing before the final product is completed.

## b. Software Environment

The software environment being used to develop the system will be Code Composer Studio, seen below in Figure. This is a free program offered by TI, which allows the user to easily code and debug for MSP microcontroller products. This program will be easy for us to use since we have seen it in many of our classes before. It has been used to allow us to program MSP430s, and will help us achieve that same goal in this system. Although the code will certainly be more complicated for this project, the usability of this software will allow us to continue working without having to research and learn different software. This program will allow the programmer to create either Assembly language files or C language files. This is useful to us since we are allowed to compile and test our code without having to change the program we are in. These features, combined with the compatibility for specific MSP430 microcontrollers will allow us to quickly and easily develop the code for our system.



There are many helpful features included with this IDE called Code Composer Studio that isn't offered from any other program. There is an option to view what registers are being used, and what data is in them. This will be extremely useful for the code we write in the Assembly language portion of our code. This feature will mainly be used when debugging and seeing why data isn't being transferred correctly, or being displayed correctly. Another reason why this program is the best for our system is for the compiler. We will be able to compile and run all of our code through the program, and we can see on our screen whether or not it is working. This feature might be seen in many other programs, but the difference

is that we are able to then connect our microcontroller to our computer with a USB connection, and we will be able to transfer the code and test it from the controller. This process will only be allowed through Code Composer Studio. Another great feature from this program is the ability to create projects. Within the project you can have multiple files, and they don't have to be the same language. For example, a programmer can create a project which contains multiple C language files and multiple Assembly language files. This allows us to use a large variety of code to test the file sizes, and to test the speed at which the code will run at. When creating a project, this program will give the programmer a list of available microcontrollers, specific to TI. We can then choose our microcontroller, the MSP430F5528, or the MSP430F5529 for launch pad testing. This will ensure us that when we compile and transfer our code to these specific microcontrollers that it will work with no problems. Without having this feature, we would have to take extra time to test and ensure that our code would be working correctly on each device.

### c. Final Product Environment

Once the prototype has been assembled and verified to pass major hardware and software tests, the accuracy of the device must be tested using the materials that end users will expect. The environment for testing the final product will ideally represent a traditional molecular biology lab to store and provide protein or DNA samples. This environment will also prevent spills from causing contamination in an environment that is unprepared for potentially volatile proteins. The teaching laboratories in the Burnett School of Biomedical Sciences are excellent test environments for this device, provided the faculty will allow this strange experimental device into their lab.

In the event that the device is not allowed into the biology building for some reason (perhaps rampant EMI from the power supply) the major requirements for a suitable facility include the following:

- A well-grounded power outlet in the range of 115-240 VAC
- Stock protein and DNA samples in the range of 1M – 0.1 mM
- Optional: Stable buffer solutions to dilute the solutions
- A cold storage (portable is acceptable) for stock protein samples
- Nitrile gloves for safely handling proteins
- A stock of cuvettes and supporting containers
- A micropipette for exact protein measurement and delivery
- An absorbent and disposable cleanup material in the event of spills
  - Kimwipes, durable paper towels, etc
- Optional: Food coloring for quick specific wavelength testing

Many of these requirements are portable and can be taken to whatever site is desired for testing. The Senior Design lab does not allow fluids that are not directly related to prototype construction, and any spills would jeopardize the equipment – and at minimum stain the

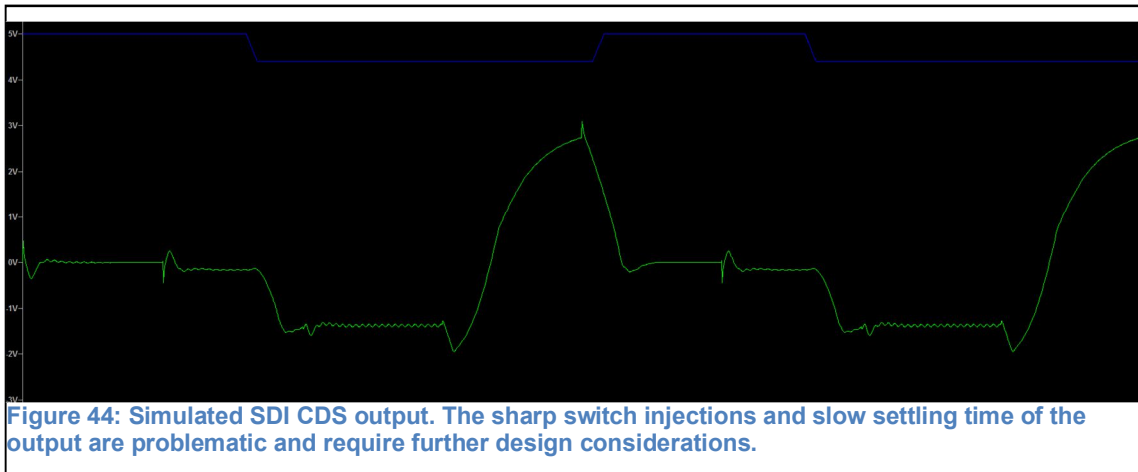
carpeting. The contingency test environment is simply a household garage with a portable cooler to store protein samples. Samples were promised from the Burnett School of Biomedical Sciences regardless of test environment, and many stock solution containers are designed for excellent shelf-life as long as they do not overheat, so any portable cooler is suitable for delivering them.

Since the power supply was assembled from scratch and may malfunction during full system testing, good ground plane dynamics and circuit breakers are required for this test environment. Most households use a very fast breaker for garages since they are likely to overdraw with large yard equipment, so this is a fine environment for this product. Regardless, attempts will be made to get this device into a real laboratory to ensure it is a good fit for our customers.

## d. Subsystem Testing

### i. Sensor Processor

Testing will begin using small systems and proceed to full-scale timing checks. Figure shows the simulated result of the proposed system. The ADC will read the value at the moment before the reset impulse is sent, approximately every 5  $\mu$ s for a 200 kHz data rate. This is seen in the diagram as the tip of the large peak before the sharp impulse. The settling time of the ADC will prevent the irregularity of the waveform from affecting the output. The system is designed around 3V of power for the ADC, but gains can be adjusted in the ADC driver to accommodate larger voltages. The OP37 also does not have the same slew rate as the TL084, so further testing will be done to ensure the data is accurate.



The sensor was powered and tested using only the DC coupling network as shown in Figure and delivered the waveform shown in Figure. This waveform shows an additional peak during the data level, but in practice this is expected and unlikely to affect the data. The

amplifier are powered on a 5V rail to prevent any issues with rail-to-rail characteristics from affecting the performance.

The extended testing plan is to wire all switching outputs to the devboard and test the scanning function under saturation and dark conditions. The resulting data can be inspected on the MCU debugger to assess the input resolution. Instead of the designed supply, the solid-state supply in the lab will be used for all components.

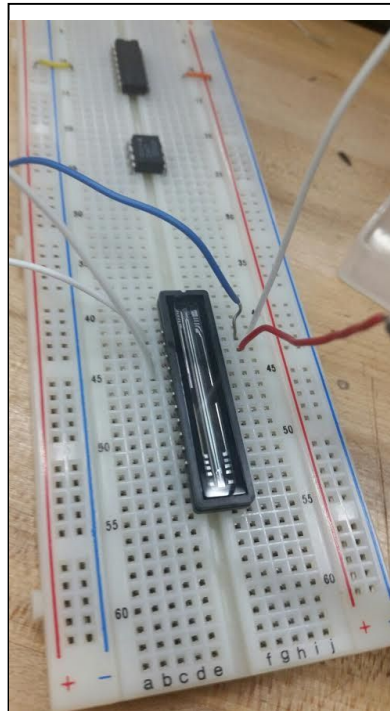


Figure 45: Sensor in testing environments

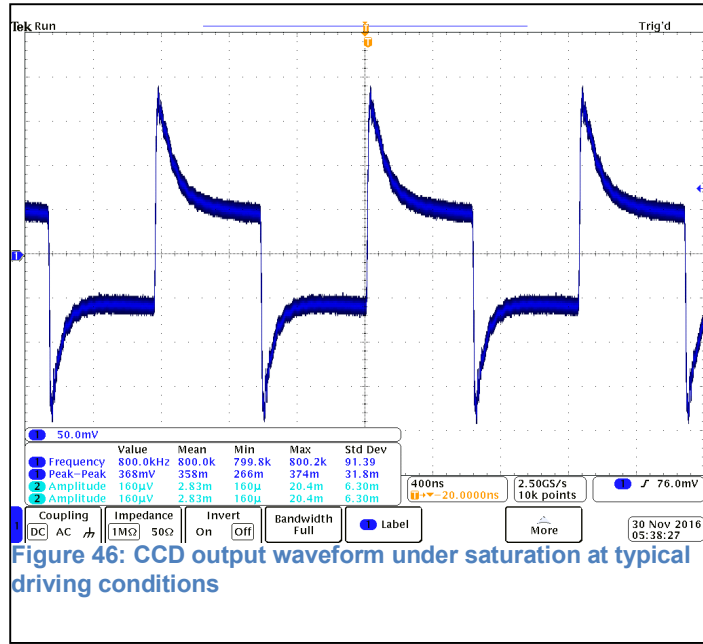


Figure 46: CCD output waveform under saturation at typical driving conditions

## ii. Optics

The optical devices of the spectrophotometer require individual testing. Individual testing allows us to confirm whether the manufacturer crafted a device with the right specifications. The three optical devices that will need to be tested are both spherical mirrors and the diffraction grating. With the spherical mirrors, the focal length is the important parameter that needs to be confirmed. The diameter of the mirror can also be measured easily. The diffraction grating can be tested to figure out parameters such as groove density, the spectral range, and diffraction efficiency. The test for groove density is done with a monochromatic source, such as a laser. Spectral range and diffraction efficiency across the range should be measured by a broadband source, like a lamp.

The test we will use to confirm the focal length of our mirrors is the parallax method test. This is method can be used for both spherical and parabolic mirrors. In principle, whenever an object is placed at the center of curvature of a mirror, the image will be formed at the center of curvature. This can be easily proved from the mirror equation as shown in Equation:

$$\frac{1}{o} + \frac{1}{i} = \frac{1}{f} \rightarrow \frac{1}{f/2} + \frac{1}{i} = \frac{1}{f} \rightarrow \frac{1}{i} = \frac{1}{f/2} \rightarrow i = \frac{f}{2} = |R|$$

Equation 15: Mirror equation derivation for testing

For the parallax method test, you need a flat surface, the mirror, a retort stand, a fixture for the stand, and a skinny object with a pointed end. A retort stand is just a weighted stand with a vertical rod sticking out of it. Place the mirror face up on the surface. Attach the object to the retort stand fixture, and insert the fixture so that it can slide on the rod. Adjust the object such that the tip of its point is in line with the center of the mirror. Then, adjust the height of the object until the tip of the image seems to be in line with the center of the mirror, also. The distance between the object and the center of the mirror at this height is considered the radius of curvature. The focal length can be calculated by halving the radius of curvature.

The groove density of a grating can be measured by taking advantage of the concept of diffraction of multiple orders. For this test, we need a laser in the visible wavelength range, a ruler, a white screen with a sizable hole in it, and the diffraction grating. When a laser hits a diffraction grating, the light is diffracted in multiple directions. These directions correspond with a mode other than the zeroth mode. At the zeroth mode, the angle of reflection (or diffraction) is equal to the angle of incidence. With the diffraction grating equation, the distance between the diffracted orders, and the distance from the grating to the screen, the groove density can be measured. To begin the experiment, set up the screen such that the laser propagates through the hole. Mount the grating within a meter away from the other side of the screen, and the laser should be centered on and normal to the grating. The diffracted orders should be visible on the screen. Measure the distance from the hole to the maximum of the first diffracted order. Then, measure the distance from the grating to the screen. We can calculate the groove separation,  $a$ , with the grating equation shown in Equation:

$$m\lambda = a\sin\theta$$

Equation 16: Grating equation for testing

Where  $m$  is an integer corresponding to the diffraction order. In our case,  $m = 1$ . The variable  $\lambda$  is the central wavelength of the laser. The letter  $\theta$  corresponds to the angle of diffraction of the first order. This is easily calculable by simple geometry since we know certain distances, as shown in Equation.

$$\theta = \tan^{-1}\left(\frac{\text{distance from hole to first maximum}}{\text{distance between grating and screen}}\right)$$

Equation 17: Test measurements to find angle of diffraction

Now that we have all the information, the groove separation can be calculated. As discussed in the diffraction grating research in Part 4 of this document, the groove density is the inverse of the groove separation. In conclusion, this is a simple experiment when it comes to diffraction gratings. The diffraction grating is a mature technology, so none of



this information is groundbreaking. However, it is important to understand the science of optical devices.

Confirming the spectral range and diffraction efficiency of a diffraction grating is a more challenging affair. In order to analyze these parameters, we can use a device similar to the one we are creating! A simple spectrometer for general use will be sufficient enough for our purposes. The source needed to illuminate the grating will need to have a large spectral range, such as the xenon lamp we will use in our spectrophotometer. This is to ensure that the diffraction grating can cover the wavelength range we want to operate within for our project. The procedure for this experiment is as follows. First, couple the light from the source into the spectrometer. This will be the reference point for the grating, which can be set by the spectrometer's software package. Next, install the grating such that light diffracted from the grating can be coupled into the spectrometer. We should be able to see the spectral range the grating is capable of diffracting. In addition to the range, we should be able to see the relative intensities of the diffracted light as a function of wavelength. This graph is a good estimate of the diffraction efficiency of the grating.

### iii. Light and Driver

The wavelength range of the light source is a crucial part of any type of spectrometric system. Without understanding the spectral range of the source, we could not call our project a UV-VIS spectrophotometer. It is integral to check if the xenon light source we acquired covers a wavelength range between 200 nm and 800 nm. The simple way to measure the spectral range of our source is with a simple spectrometer, just like the spectrometer used to approve the parameters of the diffraction grating. The principle is to couple the light from the source into the input of the spectrometer. The software package of the spectrometer will process the light and display the relative intensity as a function of wavelength. The integration time of the detector may have to be adjusted if the relative intensity graph is either too weak or oversaturated. An oversaturation occurs because the detector and the detector processing subsystem cannot process and analyze light that is too intense / bright / powerful. A decrease in the integration time of the detector can fix this issue if it occurs.

### iv. Buzzer

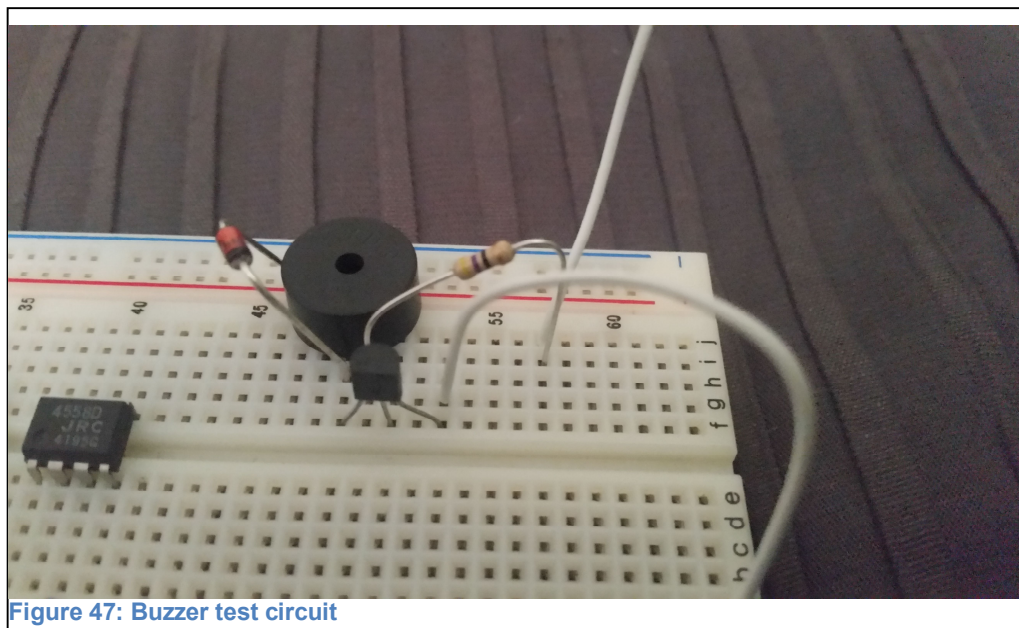
The buzzer is easily tested by constructing the network shown in the design section and applying a PWM waveform from the signal generator. The following waveforms will be tested:

- Perfect square wave (50% duty) at 4 kHz and 0-3 V amplitude
- 25% duty cycle 4 kHz 3V wave
- 10% duty cycle
- 100% (DC)

If these are successful at producing a sound, then the device can be attached to the devboard and tested using the generated buzzer function. The same conditions can be emulated using the input conditions during debugging.

Successful hardware tests can be tested using a guitar tuner or similar frequency device to ensure the output is 4 kHz. The volume should drop to half for the 25% duty cycle test and lower for 10%. At 100% duty cycle, there should be no sound as the piezo element is not being manipulated.

Due to the simplicity of this system, no further tests will be necessary before final assembly. Final product testing will ensure the system can activate the buzzer when appropriate. Figure shows the assembled test circuit.



## v. LCD and Driver

Testing our LCD should be done easily with the use of the MSP Graphics Library. Through the use of their given functions, we will be able to create sample pictures to try and display an image correctly. In order to do this, we will connect the MSP430F5529 Launch Pad to the Kentec Booster Pack, and hook it up to a computer using the USB port on the Launch Pad. This will allow us to test our displays, and see how they look on the actual LCD screen. Before doing this, however, we will be creating the images using the given software. These images will be displayed to us as an example of how they will look on the LCD display. Before displaying an actual graph using data given to us, we will create a test. This test will contain some data that we would see from our CCD, and then shows the data

on a graph. In order to get the graph to work, we will keep testing the functions until we are able to display a graph. Then once we are able to draw a graph to the screen, we will plug in the numbers, which will determine the actual data and lines for our graph.

To make this process easier for us, the MSP Graphics Library gives us example pictures to work with. While some of these are just miscellaneous images we might not use, they will guide us in drawing to the LCD. By letting use these sample images, we will have a better understanding of how the functions work when it comes to drawing an image to the screen. This will give us guidelines we need to structure the correct images. For the User Interface, we will also be sample images that are provided to us, as well as the programming examples given to us from the library. Using these resources, we will construct a UI that contains a list of options for the user to choose from, each in the shape of a button. The user can select one of these options by pressing its corresponding button, which will be connected by pins.

In order to make this process easier for us to test, the Kentec Booster Pack LCD screen van be connected to our microcontroller through input/output ports. This will remove the need to connect through an SPI connection. This may seem harder at first since connecting to multiple pins will require more connections than connecting through the SPI port. But this pin connection will be faster than SPI since we won't have to deal with using the Slave Select wire on the SPI connection. This connection will also be easy for us because the launch pad is compatible with this booster pack. This means that in order for us to test the LCD, we can just line up the pins and ports from both of the devices, and then just connect them together. This allows us to create a connection between the devices without having to use any wires. This option lets us easily connect and disconnect our LCD from the launch pad, and will allow us to easily and quickly test our images on the LCD.

## vi. Power Supply

A few considerations when assessing the functionality of the rectifier are as follows:

- vi. Test every component and write results down including waveforms.
- vii. If any diode fails to open the DC voltage waveform will be a half-wave instead of a full-wave. In addition, there will be less DC voltage across the load and more AC voltage across the load.
- viii. If the transformer secondary winding fails to open, there will be no voltage or current on secondary side of circuit after the discharging of the capacitor through the load. Furthermore, since this fault accorded there will be very little current through the primary windings.
- ix. If the inductor failed to open, there will be no voltage across or current through the load. Also, very minimal current on primary side and no current on secondary side.
- x. If the capacitor failed, there will be increased current flow through both transformer windings, diodes, and inductor. There will be minimal voltage across

and current through load. Feel capacitor and diodes, because they are likely get very hot.

- xi. When trouble shooting the design make sure you know the output voltage you are supposed to obtain and the voltage the circuit is actually supplying after tested. For example we assess the Figure layout. The procedure would be to check/test every component individually, but a faster and efficient way would just to check the fuses. If fuses are fine, proceed to measure every component. After measuring all the components and obtain reading as the following:

VTP1–TP2 = 118 volts AC,  
VTP3–TP2 = 0 volts AC,  
VTP1–TP3 = 118 volts AC,  
VTP4–TP5 = 0.5 volts AC,  
VTP7–TP8 = 1.1 volts DC,  
VTP9–TP10 = 1.1 volts DC.

Then the solution would be that the transformer has an open winding and must be replaced.

### Relevant Equations:

1. Voltage:  $V = IR$
2. Current:  $I = V/R$
3. Resistance:  $R = V/I$
4. Current through Capacitor:  $i = C[dv/dt]$
5.  $P = V^2/R = (I^2)(R)$
6. Efficiency:  $\% = P_{out}/P_{in}$
7. Average Power:  $P_A = (I_{RMS})^2(R)$
8. True Power:  $P_T = (V_{RMS})(I_{RMS})\cos(\psi)$
9. Power Factor:  $PF = \cos(\theta)$
- 10.

### Trouble Shooting:

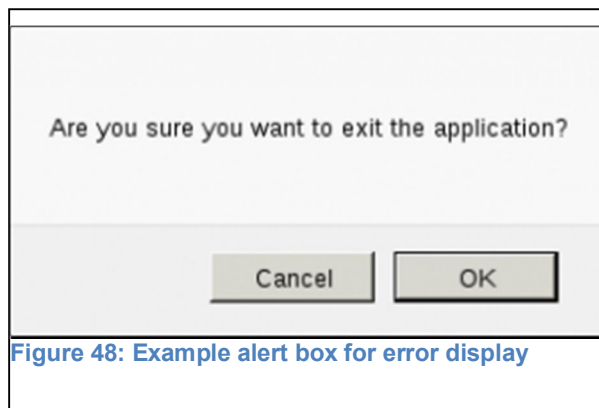
- Check for shorted diodes
- Burnt wires
- Blown fuses
- Shorted capacitor
- Test output voltage with a multi meter
- Open transformer winding (due to overloading)
- Check to see if power cord is plugged in to wall outlet
- Check regulator for correct output voltage

## vii. Final Software

In order to test the functions in our software, we will be using the MSP430F5529 Launch Pad. This tool will be used instead of our actual chip since we can easily connect it to a computer through USB, and this device can easily be written to. If we used our actual chip, we will have to flash the chip each time we wanted to change the code on it, which would take a lot longer than just writing the code to the chip through a USB port. This launch pad also comes with pins and a USB connector so we can test these connections, and see how our code works with them. The biggest difference we will have to keep in mind is the memory. The MSP430F5529 chip contains 10 KB of RAM, whereas the MSP430F5528 has 8 KB of RAM. The MSP430F5529 also comes with extra general purpose pins. Since we aren't planning on using any of these pins or extra memory, we will just have to act within our original guidelines, and ensure we don't go over these limits. Then transferring our code onto the MSP430F5528 will ensure that it works the same as it did on the launch pad.

Each function will be tested individually. This will ensure that each function can work by itself, independent of other functions. If we were to test everything together, not only would it be difficult to spot where our problems are, but going through the code would be a long and tedious task. By testing each function separately, we will keep the errors to a minimum. And if there is an error, we will be able to easily look at the function in testing, and find what is causing the error. For the USB function, we will be testing it by ensuring that a user can insert a flash drive, and have an image saved to it correctly. For the function that draws the user interface, we will test it by confirming the image for the UI is displayed on the screen correctly. We will then test the functions that are connected to the buttons of the system, ensuring that each button performs the correct task. This will be done by putting the code on the launch pad, and pressing the buttons from the UI screen, making sure everything operates correctly. Similarly to how we tested the user interface, we will test the displaying of the graph, and the functionality of the buttons on that screen the same way. The main difference here is the input of data. We will first perform a test with the CCD input function, which will just write to the programmer what data is being seen. We will then write a test function to ensure that all the data is being handled correctly by the specific equations used to display the data. After all of that, we will then combine the functions, having them work together to give us the correct graph, displayed with the data read in from the CCD.

There will also be functions in place for error messages. These messages, as seen below in Figure, are meant to prompt the user to make sure they want to move on. For example, if the user wants to exit the graph, essentially erasing all data displayed, the program will ask the user if they are sure that they want to exit this graph. These prompts will only be displayed at specific times, since asking the user over and over again whether or not they want to move on can become an annoyance, especially if they are using the program a lot.



The functionality of the firmware will initially be tested on a devboard environment, but the final functionality will mostly be assessed through final product testing. Producing usable spectral data from the sensor will be guaranteed through the sensor testing section, while inspecting the debug menu in CCS will show the contents of the spectrum arrays as they are generated. The buzzer and light are tested simply by running the associated programs and visually or audibly inspecting the hardware.

## e. Final Prototype Testing

The assembled spectrometer will be run with the following sample inputs. All samples will be used to generate a lone figure, a table of consecutive values, and a graph of the values over scan count. These figures should show very low variation and will represent the repeatability of the scanner.

- Empty sample compartment: full transmittance, 0 absorbance
- 1M BSA solution @ 280 nm: no transmittance (out of concentration range)
- 10 ug/mL BSA solution @ 280 nm: 1.32 absorbance (test against other machine)
- Green food coloring (unknown concentration) @ 500 nm: (test against)
- 1 min assay of 10 ug/mL BSA @ 280 nm @ 10 sec intervals: (test against)
- Full spectrum of 10 ug/mL BSA
- Green food coloring 1.00 abs. threshold test (add drops between scans)

These tests will be compared against a known working spectrometer where possible to ensure the concentrations are within the expected accuracy figure. All tests are simply suggestions, but they represent a wide range of conditions that are reasonable for the device.

To ensure the usability of the product, this device will be presented to peers in the biology community and they will be asked to complete a simple protocol such as those shown in the above list. A simple user manual will be provided, though ideally the software will be

close enough to standard GUI systems that operation will be intuitive. Customer satisfaction statements will be used for feedback in this stage of testing, as the primary purpose of this design is to improve customer satisfaction.

## 9. Administrative Content

The following list shows the minutes of a meeting that took place on November 1<sup>st</sup> to discuss various elements relating to the project. The results indicate that progress is somewhat behind the initial milestones but not far enough to warrant any panic. Efforts will be focused to complete the remaining project elements and complete initial testing.

### Nov. 1 Meeting

#### Subsystem identification and assignment

##### Optics

- Source – xenon bulb, bulb America, ordered
- Mirrors – research
- Grating – half price or free if rejected; ~\$50, ordering
- Sensor – CCD found, acquiring
- Lenses – none needed

##### Power

- Transformer – nothing yet (research phase)
- Outlet connector – research
- Regulators – already acquired
- Filters – saw on eBay, mostly research
- LED – acquired
- Power switch – possibly acquired, researching better options

##### Electronics

- Amplifiers – researched and ordering
- Speaker – already owned, designed
- Trimmers – research
- Hardware pulser – not required

##### Computing

- MCU – acquiring
- USB interface – research
- LCD – possibly acquired, researching better options
- Keypad – research

##### Enclosure

- Project case – PVC enclosure, research
- Optics – enclosure already acquired
- Board mounting – hardware screws at ground points, acquired

PCB vendor – Jimmy covering options

Standards assignment – three standards completed

Financial discussion – save receipts and reimburse as necessary  
Document collaboration – individual papers are emailed, combined by individual

A second meeting is attached here following the submission of the document draft.

Itinerary – Nov 17 – 2.5 weeks remain

- Finalize understanding of device and operation – good to go
- Produce differential ToC and divide unclaimed sections – completed
- Affirm status of shared blocks
  - LCD and driver
  - Enclosure
  - Light and driver
  - Keypad
- Part acquisition and testing plan
- Hardware Prototype construction
  - -cabling
  - -transport and storage – will address closer to SD2
- Report binding and delivery -
- Page formatting check – Will keep in mind during production
- Figure permissions check – Permissions being checked
- Additional considerations?
  - Memory and storage adequate

The success of the design draft led to our dividing the remaining ToC elements and continuing to produce pages. The main focus was on preparing for our first device drafts and full hardware testing plans. As final components begin to arrive, testing will take place as possible, though the requirement to redesign the sensor processor and slowness of the TI personnel to deliver our development board has led to some unwanted delays. The bulk of testing will have to take place during the break to catch up.

## Initial Project Milestones

Table and Table show the expected time frames for development. These timetables were not rigidly developed or followed but set a good pace to ensure product delivery with adequate debug time and organization.



<b>Senior Design I – Fall 2016</b>		
<b>Description</b>	<b>Duration</b>	<b>Dates</b>
Project Brainstorming	2 weeks	August 22 – September 5
Finalize Project Idea	1 week	September 5 – September 12
Initial Project Documentation		September 9
Research Individual Subsystems	2 weeks	September 12 – September 26
Design Individual Subsystems	2 weeks	September 26 – October 10
Integrate/design subsystems	4 weeks	October 10 – November 4
Table of Contents		November 4
Writing	1 week	November 4 – November 11
Senior Design I Draft		November 11
Order and organize parts for prototype	3 weeks	November 11 – December 6
Continue research and design	2 weeks	November 11 – November 21
Breadboard test of parts	2 weeks	November 21 – December 6
Finalize the design + Writing	2 weeks	November 21 – December 6
Final Senior Design I Documentation		December 6
Begin prototype creation	4 weeks	December 6 – January 9

Table 27: Expected Milestones for Semester 1

<b>Senior Design II – Spring 2017</b>		
<b>Description</b>	<b>Duration</b>	<b>Dates</b>
Begin prototype construction	1 week	January 9 – January 16
Confirm all parts are correct working	1 week	January 9 – January 16
Design and test software	3 weeks	January 16 – February 6
Assemble prototype hardware	6 weeks	January 16 – February 27
Prepare and test for midterm demo	2 weeks	February 27 – March 13
Midterm demo		TBA
Integrate and construct final product	4 weeks	March 20 – April 17
Test final product	1 week	April 17 – April 24
Writing	2 weeks	April 10 – April 24
Final presentation and demo		TBA

**Table 28: Expected Milestones for Semester 2**

## b. Budget and Finance

Table shows a brief summary of the expected costs for research and prototype development. These prices are only rough estimates and a final bill of materials will be provided upon completion.

Item	Cost	Quantity	Total Cost
PCB	\$100	1	\$100
Xe light source	\$200	1	\$200
Output display	\$200	1	\$200
Servos and motors	\$15	4	\$60
Monochromator	\$1500	1	\$500
Breadboard	\$10	1	\$10
Detector	\$50	1	\$50
Electrical components	\$50	Various	\$50
Microcontroller	\$50	1	\$50
FINAL TOTAL			\$1220

**Table 29: Estimated Budget for Development**

Financing the research and development of this project was a major roadblock. Attempts to find sponsors and customers for our construction budget were unsuccessful, and we were forced to find new methods to build our prototype from our own investment. Fortunately, some of the components were donated or discounted for student use, which greatly reduced the cost of development.

### Prototype Finance

One issue we took into consideration is the price of optics for a UV-VIS system vs. a strictly visible system. We were concerned that the optics would be out of the range of a student-friendly budget if we included near to mid UV light in our device. By means of product research, we reached the conclusion that the inclusion of the UV spectrum does not increase the price of optical components by a significant amount. Below is a general price comparison of the two scenarios.

Element	VIS	UV + VIS
Source	\$7.50	\$10.99
Detector	\$10	\$10
Grating	\$53	\$53
Mirrors / Lenses	\$100	\$105
Total	\$170.50	\$178.99

As shown, the price is very similar. However, there are some compromises we made to keep the price similar. One compromise is the quality of the detector array. The detector array we are using is ideal for visible to near infrared wavelengths. It can still be used for UV wavelengths, but the responsivity suffers at these wavelengths. We concluded that this is not a huge problem for a project funded by students. Another compromise is the power of our source. Commercial spectrophotometers use light sources that can be over \$100. We concluded that commercial lightbulbs would supply enough light for the spectrophotometric process in our case. The extra cost for a UV-VIS mirror comes from the UV enhanced aluminum coating explained in the focusing elements subsection of Part Selection and Research.

## 10. Senior Design 2 Insight

### A. Sensor Processing

The most pressing change from the design set forth in senior design I is that of the linear CCD processing system. During initial testing, the behavior of the linear CCD seemed to match that of traditional astronomical CCD arrays in that it returned to reference between each pixel value. Test waveforms collected during the design phase showed a distinct square wave pattern as presented in the sensor processing section and shown in Figure, but it eventually came to light that this was not correct. The master clock signal had been bleeding into the ground of the oscilloscope, causing these erroneous measurements to appear and making us believe we had driven the sensor correctly. As it turns out, the sensor output is a continuous stream of pixels with no reference levels, and these test waveforms were not correct.

The major drawback of this situation is that the drive conditions of the sensor were not consistently correct, a condition for which we were never able to correct. The conditions listed in the datasheet for the CCD require precise timing conditions that must be tempered to the master clock rate. Unfortunately the timers of the MSP430F5528 presented some inconsistencies in practice that would desynchronize the driving pulses from the provided clock drive, which was a fraction of the system master clock. Realistically these inconsistencies might have been caused by hidden system interrupts from the LCD drivers

provided by TI. The end result is that the CCD was never completely functional within our system, prompting the final swap to a 2D webcam array for our final demonstration.

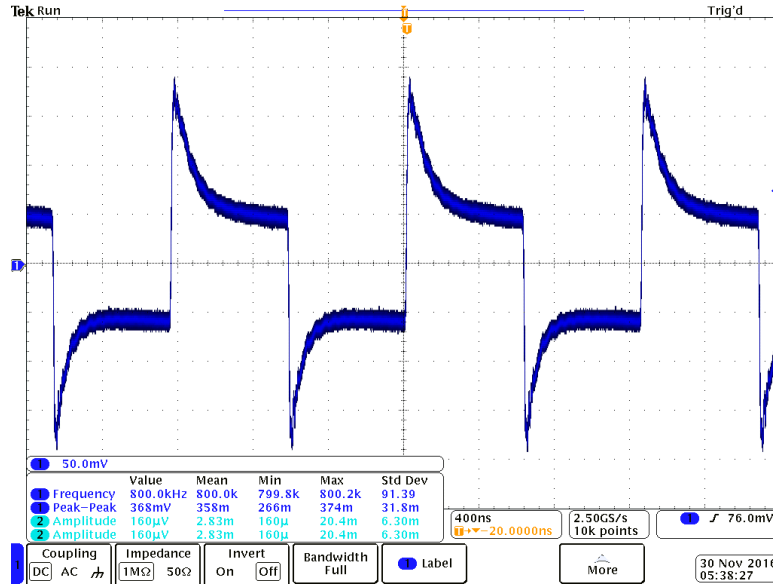


Figure 49: Originally measured CCD response

Surprisingly, the sensor processing hardware developed for the CCD was a better fit for the real pixel output than the understood output. Correlated double sampling was originally intended to resolve the two-cycle reference system of traditional arrays. Straight-forward sampling with an ADC would output values bouncing between the actual pixel output and the DC offset of the reference. This behavior was caused by the output circuitry of the CCD dumping the previous pixel charge before reading the output register. Comparing adjacent levels in the software allowed for noise cancellation without direct filtering and prevented DC restoration systems in low-power applications from losing base reference during repetitive exposures. The final schematic adopted for this project removed the hardware double sampling system proposed in early designs and instead shifted the signal into the range of the ADC without reduction. The software was designed to occasionally measure a reference pixel, but mostly measure pixel outputs directly between periods and adjust the values as necessary. With the new continuous method, this was not necessary, but the existing hardware and software were able to read every pixel of the CCD without adjustment. A fortunate circumstance following an unfortunate situation.

The final demonstration made use of a 2D CCD from a webcam and a few lines of Matlab to translate the resulting images into a spectrum. While this doesn't demonstrate much of the electronic hardware, the designed signal processing calibration schema was still

relevant to the output, and the power supply was still in use for the light source. A transmission spectrum requires an accurate reading from the source light without attenuation to provide reference for the spectral reading. An uncalibrated spectrum is still interesting for some applications, but is unsuitable for our desired application to provide concentration values. The webcam system automatically subtracts out dark signal values, but it turns out not be necessary due to the calibration scheme. The blank frame is also handled by the full signal reference, reducing the previous 4 spectral image system to 2.

During the debugging phase of the linear CCD, several online resources were found regarding the sensor in hobby projects and researched for potential solutions. It became apparent that every engineer had a different idea of how the timing protocol should behave, and implementations were highly variable and contradictory. Most used the integrated shutter function provided by Toshiba in some way, but actual effectiveness was varied. Some fixed the period of the shift register to a multiple of the master clock, as demonstrated in the datasheet, but with some multiple adjustment. This fixes the integration time to some value without variation. While this helps control the very high sensitivity of the sensor, it forces your optics into a narrow range for intensity that we were unable to control with our setup. Other systems run variable shift register periods but align the drive conditions using forced controls. Since the integration time is actually controlled by the time directly previous to the integration clear, this causes unpredictable integration behavior. Our system ignored the shutter function for its complicated requirements and instead opted to alter the light source period for integration. This had the advantage of freeing up a timer and accounting for the unknown characteristics of the light source. Ultimately the conditions were unstable in our implementation, but the theory was sound in preparation.

## B. Microcontroller

Manipulating the peripheral hardware of the MSP430 turned out to be more complicated than expected. The provided documentation is sparse and sometimes unclear on how modules are organized, and hobbyist projects tend to stay away from MSP430 hardware for its difficulty. While similar microcontrollers from other manufacturers have well-documented hardware libraries and simplifying assumptions, the MSP430 gives little help with drivers and provides a huge number of variables that must be set manually. This led to slower development of the firmware than would likely have occurred with a similar arduino or PIC model.

Realistically the MSP430 architecture itself was not the right fit for the project. The hallmark of the MSP430 is the extensive low-power modes and low supply requirements. Our project made no specifications for power consumption, so this was realistically wasted in our implementation. This was unfortunate because the cost could have been lowered through better selection. The C2000 series from TI is essentially a cheaper, faster version of the MSP430 that has no low-power modes, and would have been a good fit if TI was desired. The arduino nano is a popular platform for hobby projects and would have made prototyping simple, but would not have satisfied our desired goal of creating a professional standalone device suitable for mass production.

A surprisingly robust implementation option that was not explored was redesigning the system as a Raspberry Pi HAT system. The Raspberry Pi is a cheap, portable computing system that has been seen in at least two research labs for sensing applications. A simple PCB and shroud could be created and marketed as a successful low-cost spectrometer that requires an additional light source. Even experienced electronic hobbyists are unlikely to set up a spectrometer without professional experience, as the electronics and optics are complicated and expensive. While an unusual application, and not what we pitched in our initial specifications, it would likely have been a good choice for a similar project.

The firmware was developed in C to ease the transition into the GUI software, but a better choice would have been assembly. The timing of the peripherals was important to the functionality of the system, and the unreliable translation of C code into machine code led to errors that required extensive redesign and compiler manipulation. Embedding assembly within a C program is not explicitly difficult, but serves to make debugging more taxing. A module could be developed and linked to the graphic software as a standalone system firmware library, but no one in the group had experience doing so.

Overall the microcontroller selection led to the software development being very difficult and time-consuming. Though much of the care in this project should have been the hardware, the software ended up leading to more issues and ultimately led to the reduced functionality of the demonstrated prototype. As a closing thought, the sheer number of complex disciplines involved in this device made it a poor senior design project. Future projects should likely focus on more traditional electronics or photonics, and no underestimate the difficulty of software development and scientific application.

## C. Final Presentation Description

In the end, our group was not able to fulfill all the constraints and goals that we set during the fall semester. One of the goals that we did not meet is to increase the ease of use of the spectrophotometer. The way we wanted to approach this issue is to implement an LCD screen and a simple keypad for the user. The user would be able to select which test he or she would like the spectrophotometer to perform, and the results would be displayed in an easy-to-understand way on the screen. For our final presentation, we had to resort to using a separate computer to perform our test and analyze the results.

The test we performed in our presentation is not exactly what a real spectrophotometer would do. First, we made sure the spectrometer subsystem will perform correctly. This consists of the light source, the entrance slit, the concave mirrors, and the diffraction. We were able to construct the subsystem and place it in our enclosure. We confirmed the spectrometer was working because a spectrum of visible light could be seen at the output of the spectrometer. The output of the spectrometer is supposed to be incident on the CCD detector. Unfortunately, we were not able to get the CCD working for the final presentation. We resorted to the best alternative we could find: a camera.

The camera we used is the Logitech C270 Webcam. This is a simple off-the-shelf camera for computer use. It is not difficult to setup and the picture quality works for our application. We simply placed the webcam in the image plane. Using Logitech Webcam Software, we were able to capture the spectrum as picture in JPEG format. First, we captured a picture of the spectrum without a sample in the way of the optical axis. This picture would be the full visible spectrum mentioned earlier. Then, we placed a sample inside of our device, along the optical axis of the spectrometer. The sample we used was simply green food coloring diluted in water. We captured the transmission spectrum created by the sample with the camera. Capturing images of spectra is not enough content to constitute our final presentation. So, we had to find a way to analyze the spectra in the form of computer images. We were able to use the program MATLAB to process these images into useful data that a true spectrophotometer can provide.

MATLAB is a very powerful tool, and one of the things it succeeds at is image processing. With MATLAB, you are able to input any computer image and analyze each pixel of the picture. We took this core concept of image processing and applied it to the images of the spectra. First, both images are read into MATLAB. The images are converted into grayscale images. The pixel values of the grayscale image range from 0 to 255, corresponding to the intensity of the pixels of the original images. Then, a function called 'improfile' allows a person to graph the intensity profile of an image across a line of pixels. So, the 'improfile' function was applied to the middle of the spectra on both images. After



the function is executed, MATLAB creates a matrix of intensity values for each pixel and plots the matrix as intensity values vs. pixel number. The total number of pixels depends on the line of pixels you choose when executing the function. This graph is called the transmission intensity profile. This graph is what we wanted to appear on our LCD screen for a user after the test is performed. Nevertheless, we were able to analyze the data we collected from the camera. Finally, we can analyze the two profiles we created. The profile of the full spectrum acts as our reference level for any other profile we obtain. In this demonstration, we can compare the profile from the spectrum with the sample to the reference intensity profile. With the use of math with the profile matrices, the true transmission and absorption curves of the sample can be created.

## References


1. <http://www.chemguide.co.uk/analysis/uvvisible/beerlambert.html>
2. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690723/#b1>
3. <http://www.ruf.rice.edu/~bioslabs/methods/protein/protein.html>
4. <http://cshprotocols.cshlp.org/content/2007/11/pdb.ip47.full>
5. <http://www.life.illinois.edu/biochem/455/Lab%20exercises/2Photometry/spectrophotometry.pdf>
6. [http://www.specinst.com/What\\_Is\\_A\\_CCD.html](http://www.specinst.com/What_Is_A_CCD.html)
7. "All About Circuits: Experiments: A Very Simple Circuit." *All About Circuits: Experiments: A Very Simple Circuit*. N.p., n.d. Web. 10 Nov. 2016.
8. CUI Inc Follow. "Power Supply Safety Standards, Agencies and Marks." *Power Supply Safety Standards, Agencies and Marks*. N.p., 14 Oct. 2014. Web. 10 Nov. 2016.
9. "LATEST." *Voltage Divider Circuits : AC Electric Circuits Worksheets*. N.p., n.d. Web. 10 Nov. 2016.
10. Mar 12, 2013 Dave Johnson. "What's The Difference Between USB 2.0 And 3.0 Hubs?" *Electronic Design*. N.p., n.d. Web. 10 Nov. 2016.
11. "Introduction to I2C and SPI Protocols." *Byte Paradigm Speed up Embedded System Verification*. N.p., n.d. Web. 10 Nov. 2016
12. <https://www.kanda.com/blog/author/kandadmin/>. "Using USART or UART on a Microcontroller." Kanda Electronics Blog. N.p., 09 July 2013. Web. 10 Nov. 2016.
13. London: Wace, 2013. Web.  
[http://vip.gatech.edu/wiki/images/4/46/MSP430\\_data\\_communication.pdf](http://vip.gatech.edu/wiki/images/4/46/MSP430_data_communication.pdf)  
[http://qsimaging.com/ccd\\_noise.html](http://qsimaging.com/ccd_noise.html)
14. <http://www.photometrics.com/resources/technotes/pdfs/snr.pdf>
15. <http://www.analog.com/media/en/training-seminars/tutorials/MT-047.pdf>

16. [http://www.uni-muenster.de/imperia/md/content/ziv/multimedia/downloads/kodak\\_\\_\\_noise\\_sources.pdf](http://www.uni-muenster.de/imperia/md/content/ziv/multimedia/downloads/kodak___noise_sources.pdf)
17. <http://davidallmon.com/pages/ad7667-spectrograph>
18. Mar 12, 2013 Dave Johnson. "What's The Difference Between USB 2.0 And 3.0 Hubs?" *Electronic Design*. N.p., n.d. Web. 10 Nov. 2016.
19. "Introduction to I2C and SPI Protocols." *Byte Paradigm Speed up Embedded System Verification*. N.p., n.d. Web. 10 Nov. 2016
20. <https://www.kanda.com/blog/author/kandadmin/>. "Using USART or UART on a Microcontroller." Kanda Electronics Blog. N.p., 09 July 2013. Web. 10 Nov. 2016.
21. London: Wace, 2013. Web.  
[http://vip.gatech.edu/wiki/images/4/46/MSP430\\_data\\_communication.pdf](http://vip.gatech.edu/wiki/images/4/46/MSP430_data_communication.pdf)
22. "The Importance of 'Grounding' Electrical Currents - Platinum Electricians." *Platinum Electricians*. N.p., 11 Sept. 2016. Web. 03 Dec. 2016.

# Appendices

## A. Permissions

MicroscopyU Figure Permission Request


 Eric Clark <eclark@magnet.fsu.edu>  
Wed 11/9, 5:24 PM

Sean,

You are welcome to use the figure for your project.

Eric Clark  
[eclark@magnet.fsu.edu](mailto:eclark@magnet.fsu.edu)

---

 Sean Pope  
Tue 11/8, 9:09 PM  
eclark@magnet.fsu.edu

Sent Items

Hello,




I am an engineering student with the University of Central Florida performing a senior design project. The project contains a CCD light sensor and I would like to include a short write-up in my design documentation on CCD construction and operation. I would like to request permission to reprint a figure used by MicroscopyU in my report for this course. The report is for educational use only and is not for profit.

The figure in question can be found at this link: <https://www.microscopyu.com/digital-imaging/introduction-to-charge-coupled-devices-ccds>

It has the following caption: Figure 3 - CCD Sense Element (Pixel) Structure

Thank you for your time,  
Sean Pope

**Permission 1: MicroscopyU**

 **Reid Hornberger** 11:24 AM (42 minutes ago) ☆  

to me ▾


Jimmy,

Thank you for reaching out, that should not be an issue as long as you list CUI as the source of your images. What images were you planning to use in particular?




Regards,  
**Reid Hornberger**  
Communications Specialist  
CUI Inc

Direct: [503.612.2336](tel:503.612.2336)  
Toll Free: [800.275.4899](tel:800.275.4899)  
[www.cui.com](http://www.cui.com)




Connect with us:  
[twitter](#) [facebook](#) [LinkedIn](#)



**Permission 2: CUI**

Permission   

---

 **jimmy vallejo** <vallejojimmy69@gmail.com> Dec 2 (1 day ago) ☆  


to sales ▾

To whom it may concern,

I am Jimmy Vallejo and an alumni at the University of Central Florida, I'm currently taking Senior design and needing to get permission for images used in my report. May I be able to use your images?

Thank you,  
Jimmy Vallejo

---

 Click here to [Reply](#) or [Forward](#)

Technical Support Request

We encourage you to contact us with any questions you may have. Our experienced and friendly staff is happy to assist you. To request support, please use our Technical Support Request form below.

First Name\*  
Joshua

Last Name\*  
Beharry

Email\*  
joshbeha@yahoo.com

Company\*  
University of Central Florida

Model #\*  
N/A

Serial #\*  
N/A

Questions/Comments?\*

I am a senior at the University of Central Florida working on my capstone project for graduation. I was wondering if I may use any figures and tables from your website for my documentation. Thank you.

File Attachment:  
Choose File | No file chosen

Address 1  
12306 Eastcove Drive

Address 2

City  
Orlando

State  
Florida

Zip  
32826

Country  
United States

Phone\*  
4073506794

Fax

Security Code:  
XQA6

Subscribe to Our Newsletter?\*

Yes  No

Submit

Permission 3: BWTEK request

Permission

University of Central Florida  
12306 Eastcove Drive, Orlando, Florida, 32826, United States

Attention: Mr. Joshua Beharry  
Appointment:  
Permission for using image items and others of HORIBA

Gentleman:  
This letter sets forth the permission from HORIBA to you for using following items under following conditions.

"Items": Figure of 4 in the diffraction gratings on the HORIBA web site

"Copyright Notice": ©HORIBA, Ltd. All rights reserved.

"Reference number in HORIBA": HOR-C16-031

"Purpose for use": Usage of Figure 4 to illustrate blaze angle and the Littrow condition of a diffraction grating for your project.

Conditions for use:

1. Please indicate the Copyright Notice on the Items.
  2. Do not use the Items for any purpose other than the "Purpose for use" hereof.
  3. Do not make any modification (such as, but not limited to, combine, convert and/or process) to the Items.
  4. Do not copy, distribute, transfer, rent, sale, sublicense, and/or secondary distribute the Items without our permission.
- In the event that HORIBA recognize your infringement of the above, HORIBA may prohibit you from using the Items.

Very truly yours,

HORIBA, Ltd. ("HORIBA")


By: 



Name: Akio Maeno

Title: Department Manager, Corporate Communications Office

Date: November 24, 2016

Request for Permission to Use Website Images ⌵

 **Joshua Beharry**  
 Today, 10:08 PM  
 Thorlabs Technical Support <techsupport@thorlabs.com> ✉

  Reply all | ⌵

Hello,

I am a senior at the University of Central Florida studying Optics and Photonics. At the present time, I am working on my capstone project for senior design. I recently purchased a diffraction grating from ThorLabs, and I would like to know if I can use the efficiency curve on the website in my documentation? On that note, I would like to request permission to use any images on your website in my documentation.

Thank you for your time and consideration,

Joshua Beharry

**Permission 5: Thorlabs request**

\*Interest/Reason for Contact  Sales Support/Customer Service  
 Product Support  
 Website Issue

\*First Name

\*Last Name

Company

Job Title

\*Phone

Extension

\*Email Address

Fax

\*Country

\*Is your application military/defense related?

\*Comments

**Permission 6: Edmund Optics request**



## RE: Figure Permission Request

Barnes, Erik <Erik.Barnes@analog.com>

Thu 12/1/2016 6:40 PM

To: Sean Pope <sean.pope@knights.ucf.edu>;

Hi Sean,

As long as you cite the material, it should be fine. Good luck with your project.

-Erik Barnes

---

**From:** Sean Pope [mailto:sean.pope@knights.ucf.edu]

**Sent:** Wednesday, November 30, 2016 7:03 PM

**To:** Barnes, Erik

**Subject:** Figure Permission Request

Hello,

I am an electrical engineering student at the University of Central Florida working on my senior design project. My project includes a CCD and I would like to include a figure in my report from one of your articles to show the expected wave characteristics. The figure can be found at the following location:

Figure 1: CCD Output Stage

<http://www.analog.com/en/analog-dialogue/articles/integrated-solutions-for-ccd-signal-processing.html>


This report is for educational use only but I require explicit permissions to include any figures that I have not generated.

Thank you for your consideration,  
Sean Pope

**Permission 7: Analog**


## B. Datasheets

first pages only to save space




Jameco Part Number 36038

### 1N/FDLL 914/A/B / 916/A/B / 4148 / 4448



**DO-35**



**LL-34**

THE PLACEMENT OF THE EXPANSION GAP HAS NO RELATIONSHIP TO THE LOCATION OF THE CATHODE TERMINAL.

COLOR BAND MARKING

DEVICE	1ST BAND	2ND BAND
FDLL914	BLACK	BROWN
FDLL914A	BLACK	GRAY
FDLL914B	BROWN	BLACK
FDLL916	BLACK	RED
FDLL916A	BLACK	WHITE
FDLL916B	BROWN	BROWN
FDLL4148	BLACK	BROWN
FDLL4448	BROWN	BLACK

### Small Signal Diode

#### Absolute Maximum Ratings\* T<sub>a</sub> = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V <sub>RRM</sub>	Maximum Repetitive Reverse Voltage	100	V
I <sub>F(AV)</sub>	Average Rectified Forward Current	200	mA
I <sub>FSM</sub>	Non-repetitive Peak Forward Surge Current Pulse Width = 1.0 second Pulse Width = 1.0 microsecond	1.0	A
		4.0	A
T <sub>stg</sub>	Storage Temperature Range	-65 to +200	°C
T <sub>J</sub>	Operating Junction Temperature	175	°C

\*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:  
1) These ratings are based on a maximum junction temperature of 200 degrees C.  
2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

#### Thermal Characteristics

Symbol	Characteristic	Max	Units
		1N/FDLL 914/A/B / 4148 / 4448	
P <sub>D</sub>	Power Dissipation	500	mW
R <sub>θJA</sub>	Thermal Resistance, Junction to Ambient	300	°C/W

1N/FDLL 914/A/B / 916/A/B / 4148 / 4448



**DB Products Limited**  
SINCE 1976

11/F, GUANGDONG FINANCE BUILDING, 88 CONNAUGHT ROAD WEST, HONG KONG  
Tel.: (852) 2548 2098 Fax: (852) 2858 2514, 2858 2520, 2549 3161 E-mail: db\_buzzer@db.com.hk  
www.db.com.hk ISO 9001

### SPECIFICATIONS SHEET

PRODUCT TYPE : EXTERNAL DRIVE PIEZO TRANSDUCER

PRODUCT SERIES : DB-E1548

DESCRIPTION : PZ TRANSDUCER 5V L/W ABS SP RoHS

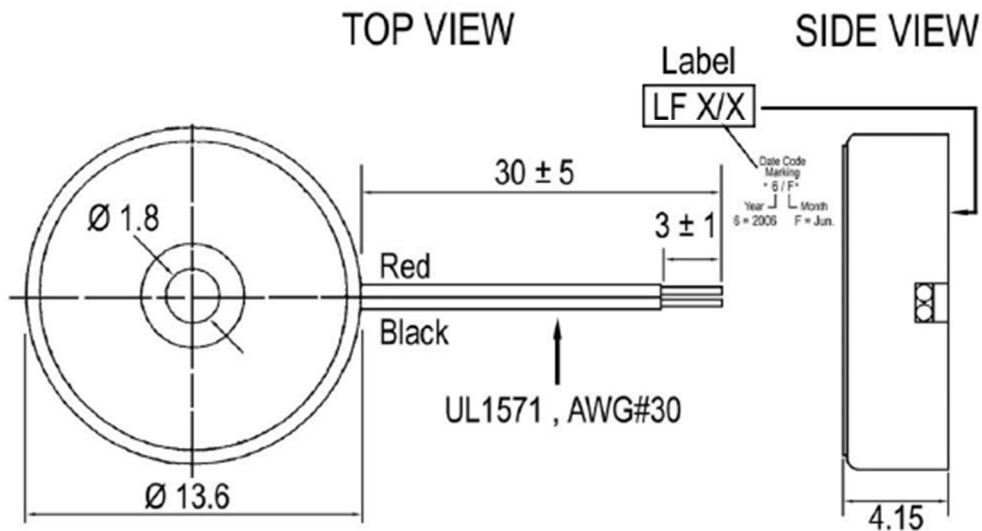
COLOR : BLACK

#### ELECTRICAL SPECIFICATIONS

MODEL NO.	154PZ14480LF - LA
SOUND PRESSURE LEVEL	80.0 dBA min. ( 5.0V p-p Square Wave , 10 cm , 4800 Hz )
CAPACITANCE	12000 pF ± 30 % at 120 Hz
*MAX INPUT VOLTAGE	30.0 V p-p
OPERATING TEMPERATURE	-20 ~ +60 ° C
TERMINAL	LEADWIRE ( UL1571 , AWG#30 )
HOUSING MATERIAL	ABS
*WEIGHT	3.0 g

#### DIMENSIONS

UNIT : mm



Approved By :   
Date : 20-Jun-2006

Prepared By :   
Date : 20-Jun-2006

Modify :	Date :	By :

Ref.:2-001

## TL08xx JFET-Input Operational Amplifiers

### 1 Features

- Low Power Consumption: 1.4 mA/ch Typical
- Wide Common-Mode and Differential Voltage Ranges
- Low Input Bias Current: 30 pA Typical
- Low Input Offset Current: 5 pA Typical
- Output Short-Circuit Protection
- Low Total Harmonic Distortion: 0.003% Typical
- High Input Impedance: JFET Input Stage
- Latch-Up-Free Operation
- High Slew Rate: 13 V/ $\mu$ s Typical
- Common-Mode Input Voltage Range Includes  $V_{CC+}$

### 2 Applications

- Tablets
- White goods
- Personal electronics
- Computers

### 3 Description

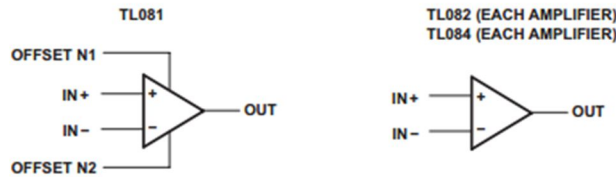
The TL08xx JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
TL084xD	SOIC (14)	8.65 mm × 3.91 mm
TL08xxFK	LCCC (20)	8.89 mm × 8.89 mm
TL084xJ	CDIP (14)	19.56 mm × 6.92 mm
TL084xN	PDIP (14)	19.3 mm × 6.35 mm
TL084xNS	SO (14)	10.3 mm × 5.3 mm
TL084xPW	TSSOP (14)	5.0 mm × 4.4 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

#### Schematic Symbol



### MSP430F552x, MSP430F551x Mixed-Signal Microcontrollers (Rev. M)

SLAS590M – March 2009 – revised November 2015

MSP430F5513, MSP430F5514, MSP430F5515, MSP430F5517, MSP430F5519, MSP430F5521, MSP430F5522, MSP430F5524, MSP430F5525, MSP430F5526, MSP430F5527, MSP430F5529, MSP430F5531

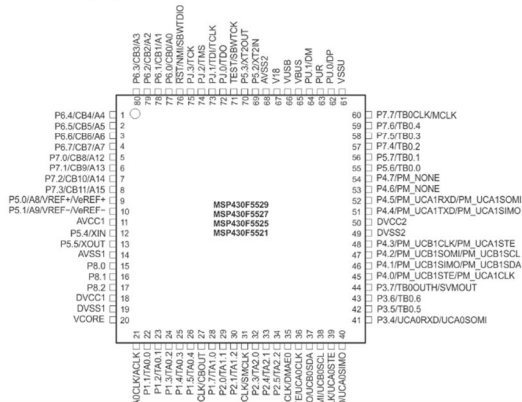
PRODUCTION DATA

See TI store options | Download PDF | Email Datasheet

#### 4 Terminal Configuration and Functions

##### 4.1 Pin Diagrams

Figure 4-1 shows the pinout for the MSP430F5529, MSP430F5527, MSP430F5525, and MSP430F5521 devices in the PN package.



# 1N4001 THRU 1N4007

## PLASTIC SILICON RECTIFIER

VOLTAGE - 50 to 1000 Volts CURRENT - 1.0 Ampere

### FEATURES

- Low forward voltage drop
- High current capability
- High reliability
- High surge current capability
- Exceeds environmental standards of MIL-S-19500/228

### MECHANICAL DATA

Case: Molded plastic, DO-41

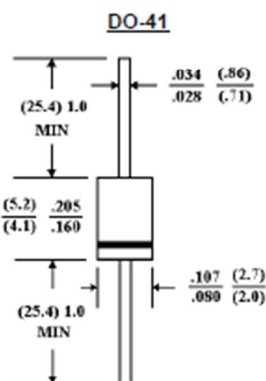
Epoxy: UL 94V-O rate flame retardant

Lead: Axial leads, solderable per MIL-STD-202, method 208 guaranteed

Polarity: Color band denotes cathode end

Mounting Position: Any

Weight: 0.012 ounce, 0.3 gram



Dimensions in inches and (millimeters)

### MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Ratings at 25 ° ambient temperature unless otherwise specified.

Single phase, half wave, 60 Hz, resistive or inductive load.

For capacitive load, derate current by 20%.

	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNITS
Maximum Recurrent Peak Reverse Voltage	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	35	75	140	280	420	560	700	V
Maximum DC Blocking Voltage	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Current .375"(9.5mm) Lead Length at T <sub>A</sub> =75 °	1.0							A
Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load (JEDEC method)	30							A
Maximum Forward Voltage at 1.0A DC and 25 °	1.1							V
Maximum Full Load Reverse Current Full Cycle Average at 75 ° Ambient	30							° A
Maximum Reverse Current at T <sub>A</sub> =25 °	5.0							° A
At Rated DC Blocking Voltage T <sub>A</sub> =100 °	500							° A
Typical Junction capacitance (Note 1)	15							pF
Typical Thermal Resistance (Note 2) R ° JA	50							° /W
Typical Thermal resistance (NOTE 2) R ° JL	25							° /W
Operating and Storage Temperature Range T <sub>J</sub> , T <sub>STG</sub>	-55 to +150							°

#### NOTES:

1. Measured at 1 MHz and applied reverse voltage of 4.0 VDC.
2. Thermal Resistance Junction to Ambient and from junction to lead at 0.375"(9.5mm) lead length P.C.B mounted.

## LM78XX Series Voltage Regulators

### General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

### Features

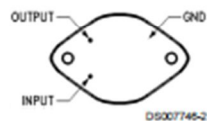
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

### Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

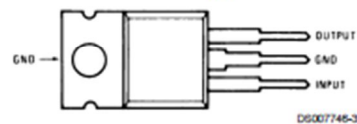
### Connection Diagrams

**Metal Can Package  
TO-3 (K)  
Aluminum**



**Bottom View**  
Order Number LM7805CK,  
LM7812CK or LM7815CK  
See NS Package Number KC02A

**Plastic Package  
TO-220 (T)**



**Top View**  
Order Number LM7805CT,  
LM7812CT or LM7815CT  
See NS Package Number T03B

# MC79XX/MC79XXA/LM79XX

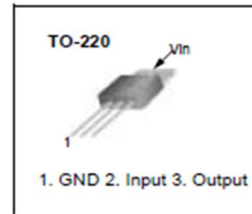
## 3-Terminal 1A Negative Voltage Regulator

### Features

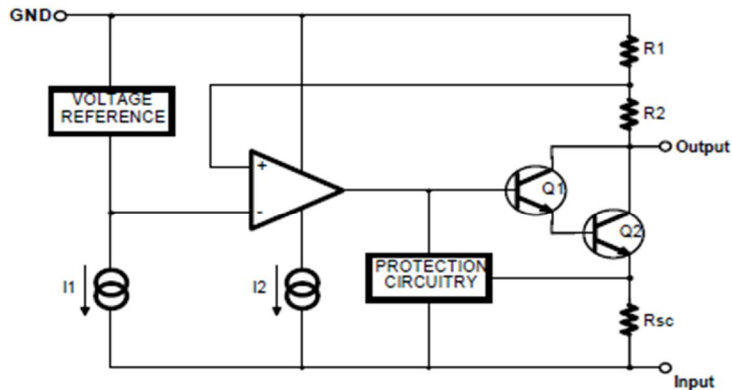
- Output Current in Excess of 1A
- Output Voltages of -5, -6, -8, -9, -10, -12, -15, -18 and -24V
- Internal Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Compensation

### Description

The MC79XX / MC79XXA / LM79XX series of three terminal negative regulators are available in TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible.



### Internal Block Diagram



Rev. 1.0.1

## LM79XX Series 3-Terminal Negative Regulators

### General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of  $-5V$ ,  $-8V$ ,  $-12V$ , and  $-15V$ . These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting, safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of

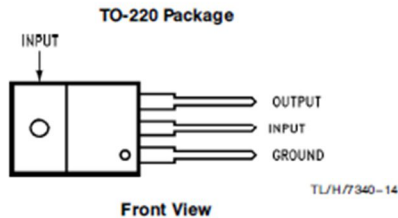
these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

For applications requiring other voltages, see LM137 data sheet.

### Features

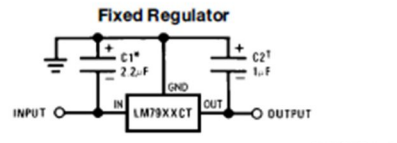
- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

### Connection Diagrams



Order Number LM7905CT, LM7912CT or LM7915CT  
See NS Package Number TO3B

### Typical Applications



\*Required if regulator is separated from filter capacitor by more than 3". For value given, capacitor must be solid tantalum. 25  $\mu F$  aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25  $\mu F$  aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100  $\mu F$ , a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.