Department of Physics and Astronomy

Ithaca College

Senior Project Report

Diffraction Grating Spectrometer with an Arduino

Submitted by

Thy Doan Mai Le

April 15, 2020

ITHACA COLLEGE DEPARTMENT APPROVAL

of a Senior Project submitted by

Thy Doan Mai Le

This senior project report has been reviewed by the senior project instructor and has been found to be satisfactory.

_______________________________ _______________

Dr. Kelley D. Sullivan, Date
Senior Project Instructor

_______________________________ _______________

Dr. Matthew C. Sullivan, Interim Chair, Date
Department of Physics and Astronomy

I authorize the Ithaca College Department of Physics and Astronomy to place a print copy of my project report in the Society of Physics Students student room and to retain a digital copy. I understand that my project report may be distributed within the Department or College for educational purposes.

_______________________________ _______________

Thy Doan Mai Le Date

Acknowledgments

I would like to thank Professor Daniel A. Briotta for dedicating his last semester as a Professor towards guiding me throughout the project. Professor Briotta is not only a constant source of support, inspiration and encouragement, he has taught me to believe in myself even when nothing seems to work. From Professor Briotta, I have learned that as long as you believe that it would work, it will work. Thank you, Dan!

I would also like to thank Prof. Kelley Sullivan for helping me throughout the writing process, Prof. Matthew Sullivan for his assistance in the project proposal as well as his helping me purchase replacement parts for the components that I burned, and Prof. Toby Dragon from Computer Science for showing me how to correctly configure my Python environment. Without all of the professors above, this project would not have ended in such a success.

In addition, I would like to thank Professor Ron Armale, Professor Christina Plett and Professor David Nusbaum from Cypress Community College, as well as Professor Brian Shotwell from University of California, San Diego. I still remember learning about Newton's laws of motion from Professor Ron Armale's Physics 221 course in Fall of 2016. Without his constant encouragement, I would not have considered myself capable enough to change my major to Physics. Thanks to Professor Plett and Professor Nusbaum for teaching me all of the math I would ever need to know, especially to Professor Plett's "tabular" integration method. Apparently, it isn't very well known on this coast. I still remember being yelled at by Professor Shotwell in Physics 222 when I didn't know what to plot against what on a lab. Don't worry, Prof. Shotwell. I have so far learned to plot and analyze just about anything, in just about any language out there (except COBOL). But without Prof. Shotwell, I wouldn't have such a strong foundation in physics, so thank you!

Lastly, I would like to dedicate a special thank you to my friends at IC Physics & Astronomy. I would name all of you but I fear that such a list would bore my reader even further. Thank you for adopting me into your department. Without your support and your listening to my complaints whenever my code stopped working or I burned another op-amp, I would not have felt so at home here.

To my reader, thank you for picking up this project report. If you are reading this, that means that you must be involved with the spectrometer somehow. The spectrometer is very special to me, so please treat it with absolute care. It may be old, but it works just like new. Thank you for your interest in taking care of something that I hold very dearly to me, and have fun!

Abstract

In 1985, Professor Dan Briotta acquired a grant that made the purchase of a spectrometer possible for the Department. At the time, the spectrometer was a brand new model, complete with a scan controller and auxiliary optical parts, and many students had their first experience in spectroscopy with this very spectrometer. Over the years, the NI-DAQ card that used to control the spectrometer began to wear down, as is customary for electronic components that are at least 30 years old. As a result, the spectrometer entered a period of hibernation and students began to lose interest in the spectrometer due to a series of mechanical and software problems. In 2004, the spectrometer took its last set of data before going into obsolescence for the next 15 years.

In this project, the spectrometer's NI-DAQ card is replaced by a more modern microprocessor called an Arduino and a new graphical user interface is written to replace the old Igor program that used to run the spectrometer. It is my hope that this project would restore the spectrometer's capabilities and renew the students' interest in spectroscopy at IC Physics & Astronomy.

Contents

1. Introduction

1.1 History

In 1835, the cynical French philosopher August Comté confidently claimed that humans would never be able to understand the chemical composition of stars [6]. Just 15 years later, when Joseph Fraunhofer put a prism in front of the objective lens of his telescope, he saw that the sunlight exiting his telescope produced unexpected dark lines on a spectrum that was previously thought to be continuous. Unfortunately, Fraunhofer passed away before he had the chance to fully understand the dark lines that are named after him, without knowing that these lines would later reveal the chemical composition of our sun and invalidate the French cynic's claim entirely.

Fast forward to 1859 in Heidelberg, Germany, chemists Gustav Kirchhoff and Robert Bunsen successfully developed the very first Bunsen burner [6]. In order to test their burner, Kirchhoff and Bunsen would drop various solids into the flame. Without any solids burning, they witnessed that the flame would produce light without any dark lines upon passing through a prism. With a solid burning on the flame, however, they observed that the flame would produce light with distinct dark lines, and that upon switching out the burning solid, they would get a different set of dark lines. Bunsen and Kirchhoff were unknowingly observing the absorption spectra of different gases, similar to the absorption spectra of stellar gases.

White light, despite being colorless to our eyes, actually consists of all wavelengths visible to the human eye, ranging from about 4000 to 7000 Å. When white light is allowed to pass through a prism, different wavelengths will experience a different index of refraction, with longer wavelengths experiencing a higher index of refraction and shorter wavelengths, a shorter index of refraction. Therefore, longer wavelengths will incur a smaller refraction angle while shorter wavelengths will incur a larger refraction angle. The refraction angle will

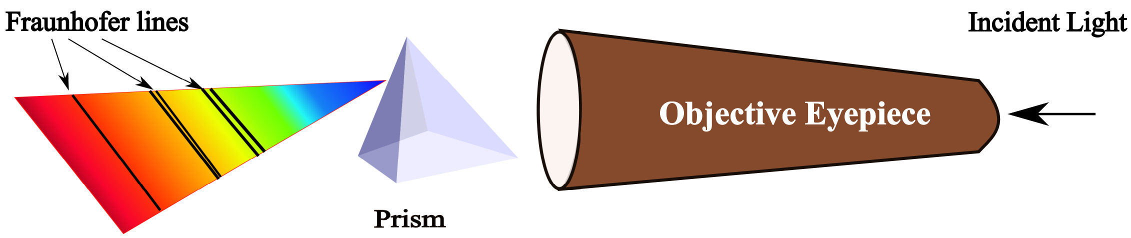

Figure 1: A simple diagram of the apparatus that Fraunhofer was using when he accidentally discovered the dark lines in the absorption spectrum of solar rays.

then determine how much "bending" each wavelength will experience upon exiting the prism. Instead of exiting the prism as white light, the exit light rays experience a dispersion because the longer wavelengths would "bend" less than the shorter wavelengths. This dispersion spreads out the individual wavelengths, which leads to the appearance of a rainbow spectrum that is similar to the output spectrum in Fig. 1. Therefore, white light passing through a prism will always show a rainbow spectrum due to dispersion.

Similar to a smooth, rainbow spectrum, absorption spectra are also continuous spectra but with characteristic dark lines at certain wavelengths. These dark lines occur at wavelengths that correspond exactly to the energy levels that the electrons inside the hot gas cloud require to jump to higher energy levels. Because every element has a unique atomic structure that defines the energies necessary for an electron to become excited, every element is expected to have its own "energy signature", a set of allowed energies that can be absorbed for orbital jumps. After repeating the same heating and observing procedure with different solids, Bunsen and Kirchhoff realized that different materials produced hot gases with their own distinct set of dark lines. In addition, when they heated an elemental source, such as hydrogen gas, and allowed light to pass through with a prism on one end, the reverse of an absorption spectrum occurred: a discrete spectrum with bright lines at very specific wavelengths and no bright lines at the remaining wavelengths. This type of spectrum is known as the emission spectrum of an element. Sometimes, these bright lines of the emission spectrum fit right into the dark lines of an absorption spectrum. When that happens, the source of the absorption spectrum must contain the element which produced the fitting emission spectrum. Using this method of overlaying the emission spectrum on top of an absorption spectrum, astronomers can then determine the elemental composition of the hot gases confined within stars. Over the next 10 years, Bunsen and Kirchhoff dedicated their time to discovering the signature emission lines of numerous materials and found 16 different chemical elements with emission lines that matched with the dark absorption lines on our sun's absorption spectrum. With this observation, Kirchhoff and Bunsen were able to determine the solar chromosphere's molecular composition as well as the sun's structure.

Fast forward to 2018, spectroscopy, which is the study of materials' emission and absorption spectral lines, has become an essential for scientists across multiple disciplines. Spectroscopy has enabled chemists to understand the chemical composition of substances. With spectroscopy, astronomers can stay on Earth and still be able to determine the chemical composition of distant cosmological objects, from aging red giants to bustling hot stellar nurseries. Most importantly, spectroscopy has defeated the baseless claims of Comte the French cynic, propelled the human spirit of discovery, from Earth all the way into outer space, and it is my hope to revive that same spirit through the renovation of the spectrometer that we have here at IC Physics.

1.2 Motivations

Since 1859, spectroscopic devices have become much more robust and versatile than Fraunhoffer's crude apparatus in Fig. 1. A schematic for the previous diffraction grating spectrograph that IC Physics currently owns can be found in Fig. 2. Unfortunately, even though the spectrometer is still functional almost 30 years after its initial purchase, it is currently in need of a serious upgrade.

The spectrometer was previously connected to an NI-DAQ card inside of the attached desktop in Fig. 2 but around 10 years ago, the NI-DAQ card began to fail and has been unable to capture data correctly. The scan controller, which is a device that has a keypad for the user to enter numerical values for properties of a scan, such as scan rate, initial position, final

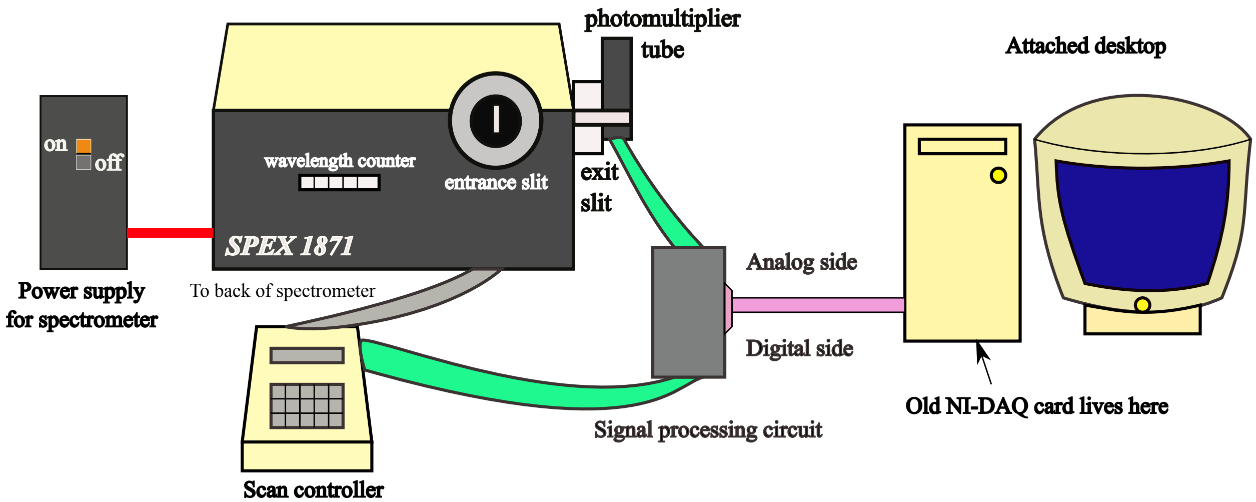

Figure 2: The diffraction grating spectrograph that IC Physics currently owns. The apparatus was purchased circa 1985 and despite being roughly 30 years old, it is still much more modern than what Fraunhofer had to use.

position, etc . . . , in Fig. 2, is also showing signs of wear after many years of operation. Due to the culmination of the above problems, student interest in the spectrometer has dropped over the years. If the spectrometer's setup is simplified and the historical components are replaced by modern components, the spectrometer could potentially regain its popularity as one of the "cool" experiments on the block and students would once again have the chance to widen their laboratory experience as they use the spectrometer for fun and exciting spectroscopic experiments.

1.3 Diffraction Grating Spectrometer

A spectrometer is a device that is typically used to differentiate individual wavelengths within a continuous spectrum with several wavelengths in it. Currently, there are three kinds of spectrometers available: a dispersion (prism) spectrometer, a diffraction grating spectrometer and an interferometer. The spectrometer that IC Physics purchased circa 1985 is a diffraction grating spectrometer, so I will proceed to give a brief introduction to the diffraction grating spectrometer in the paragraphs below.

Before we begin to understand the diffraction grating spectrometer, it is crucial that

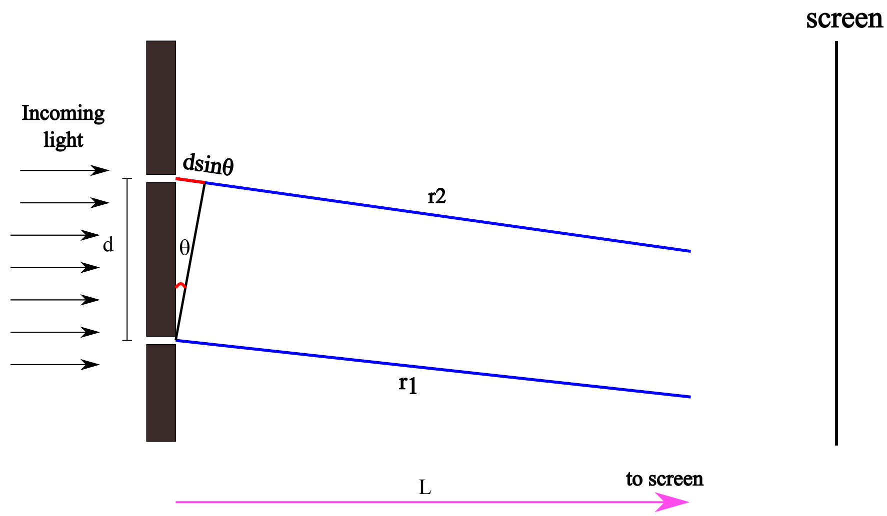

Figure 3: Diffraction occurs when light rays from a single slit interfere with one another and destructively interfere due to their path difference being an integer half of their wavelengths.

we review the theory behind Thomas Young's double slit experiment that has the setup of Fig. 3. When light encounters a wall with two open slits, each of the same width, d distance from each other, light that exits travel through different paths towards a detector screen placed some L distance parallel to the wall. Because the screen's distance from wall is often much larger than the distance between the two slits, we can assume that the paths of light from both slits are parallel to each other. Suppose that light that enters both slits were traveling to a location in the lower half of the detector screen. From Fig. 3, light from the top slit travels a distance r_2 towards the screen while light from the bottom slit travels r_1 distance to get to the screen. Distance r_2 is $d \sin\theta$ longer than r_1 due to the position of the top slit. This distance is called the path difference between light from the top slit and light from the bottom slit.

Because light consists of electromagnetic waves, light from the top slit will interfere with light from the bottom slit due to their path difference. When two waves with the same frequency and wavelength have half-a-wavelength path difference, they will cancel one another, because the crest of one wave falls exactly on the trough of the other. This phenomenon is known as destructive interference. On the other hand, when two waves with the same frequency and wavelength have a whole-wavelength path difference, they add on top of one another and create a new wave whose amplitude is the sum of the original waves' amplitudes. This is known as constructive interference.

In the double-slit experiment of Fig. 3, light from the top and bottom slits have the same frequency and wavelength, because they originally came into the slits from one single light source. If their path difference $d \sin\theta$ is any integer half of their wavelengths, light from the top and bottom slits will destructively interfere with one another and produce a dark region on the detector screen. If the path difference is any whole integer of their wavelengths, light from the top and bottom slits will constructively interfere and produce a bright region on the detector screen. Therefore, the conditions for bright regions on the screen are:

$$d \sin\theta = m\lambda \tag{1}$$

while the conditions for dark regions on the screen are:

$$d \sin\theta = \left(m + \frac{1}{2}\right)\lambda \tag{2}$$

where $m = 0, \pm 1, \pm 2, ...$ are the orders of the maxima.

Instead of a double-slit setup, our spectrometer operates upon the theory of multiple-slit interference, which is similar to the double-slit but instead of 2 slits, the barrier now has multiple slits spaced closely together. Even though the setup sounds different, the fundamental principle remains unchanged: when light from neighboring slits have a path difference that is some integer multiple of their wavelength, a bring region appears due to constructive interference. A slight deviation from the double-slit occurs when the path difference is not an integer multiple of the wavelength and the conditions for dark regions become more unpredictable. There are approximations [?] for their conditions but for the scope of this project report, I will focus on the conditions for bright regions because that is what the spectrometer will try to detect.

So far, we have only discussed interference with incoming light that has only one wavelength. White light and other spectral sources typically have more than one wavelength, with visible light having a continuous spectrum of wavelengths ranging from 4000Å to 7000Å. Since the condition for bright regions is wavelength-dependent, according to Eq. 1, each individual wavelength within a spectrum is expected to arrive at the detection screen at slightly different angular positions. From observational inspection of Eq. 1, the angular position of a bright region on the detector screen can be determined by:

$$\sin\theta = \frac{m\lambda}{d} \tag{3}$$

with θ never exceeding 90°.

If a light wave has a longer wavelength, Eq. 3 predicts that the outgoing light from the multiple-slit barrier would be found at a larger angular position on the detector screen, away from the center of the screen. For a shorter wavelength, outgoing light would be found closer to the center of the detector screen. This wavelength-based "spreading" of light causes a dispersion of wavelengths at the detector, with "blue", shorter wavelengths arriving closer to the center of the screen while "red", longer wavelengths arrive further outward from the center. A diffraction grating spectrometer operates primarily by taking advantage of this dispersion phenomenon. Specifically, the spectrometer can take a continuous spectrum of different wavelengths, such as white light, and separate out the wavelengths using a diffraction grating, which is essentially multiple slits on a barrier, to produce a dispersion of all wavelengths present in the input spectrum. The dispersion of wavelengths can be recorded and used to determine different properties of materials, such as a star's atmosphere, its age or the molecular structure of a chemical compound.

1.4 Arduino Uno

An Arduino Uno is a small, portable control board that is often used in managing electronic devices. There are 6 analog input pins and 13 digital input/output pins on an Arduino, as

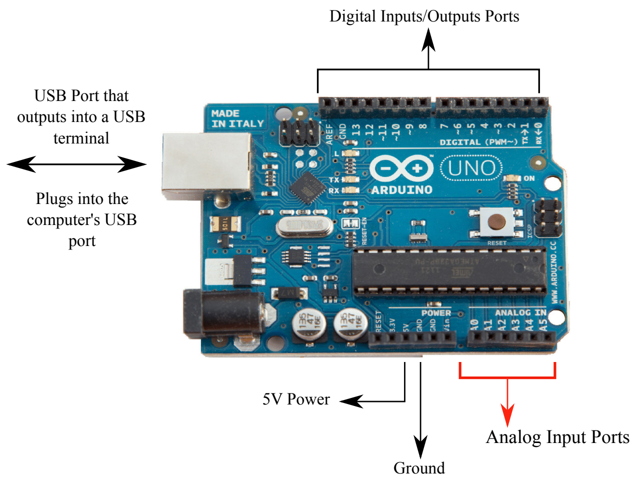

Figure 4: [1]A diagram of the basic parts of an Arduino Uno board. The USB port allows the Arduino to be connected to a computer via a USB cable. There are 13 digital ports that can be used for both digital inputs and outputs. There are 5 analog ports that can be used for analog inputs, however, the Arduino does not actually read analog signals but it converts analog signal into a digital one by an Analog-To-Digital converter that is embedded inside of the board. The Arduino has a port for Ground and for 5V power. From Ref. [1]

referenced in Fig. 4. Analog signals are electrical signals that are smooth and continuous, like a sinusoidal wave. Digital signals are signals that are discrete and step-like with shapes that resemble square waves. The Arduino is able to process both digital and analog signals that originate elsewhere but is able to output only digital signals. The Arduino is controlled by a program on a computer to which the Arduino is connected. This program must be written in the Arduino's specific text editor and in C language. Once a program is ready to run, it must be uploaded onto the Arduino from the Arduino's text editor. A snapshot of the Arduino's IDE is referenced in Fig. 5. An IDE is an Integrated Developing Environment that typically combines a text editor and a compiler, or interpreter, together into one package. This way, the user can write the program and compile it right when they are done with writing. The pre-written program includes a setup routine that initializes the Arduino (set certain pins for output, input and any other necessary internal settings such as timers) and a loop routine that runs indefinitely until the Arduino is disconnected from the computer, or until an instruction in the loop routine tells the Arduino to stop.The Arduino communicates to the laptop that controls that Arduino by a USB cable to the USB port in Fig. 4. The

USB connector is plugged into a USB port on the laptop that is controlling the Arduino. The Arduino is versatile in the sense that it acts as the mediator between a central command center (i.e.: the computer) and external electrical components. It does not need to know what these external components actually are. As long as the external components can be controlled using voltage signals, the Arduino can be programmed to control these components.

One important capability of the Arduino is its interrupt routine, in which the Arduino halts its current task and performs a special routine that only runs during an interrupt. This special routine can either be a library function or a function that was written by the user. Once the special routine is completed, the Arduino will return to whatever it was doing before the interrupt and keep on doing that task, until an interrupt is triggered again. The condition for an interrupt (both external and internal) can all be determined by the user according to the user's specific needs. A very common way that interrupts are utilized is for blinking a light (typically an LED) upon the push of a button: the Arduino can interrupt when it receives a voltage signal from the push button and turn the LED off and back on again. Once the blink is completed, the Arduino goes back into its normal loop routine until the next push of the button. For the purposes of this project, the Arduino's external interrupt ability is used to monitor the stepper motor of the spectrometer and this use will be further discussed in a later section of this report.

1.5 Project Goals

Previously, the spectrometer was controlled and monitored by a NI-DAQ board that had so far began to malfunction. An IGOR program, which ran on a very old Mac computer situated next to the spectrometer, used to communicate with the scan controller in Fig. 2, via the NI-DAQ card, and plot the data as it came in. The NI-DAQ board would send instructions to the scan controller, which would then send instructions to the stepper motor. The stepper motor is an internal mechanical part that is used in rotating the diffraction grating, which will be discussed in a later section. The NI-DAQ card was also used to communicate with

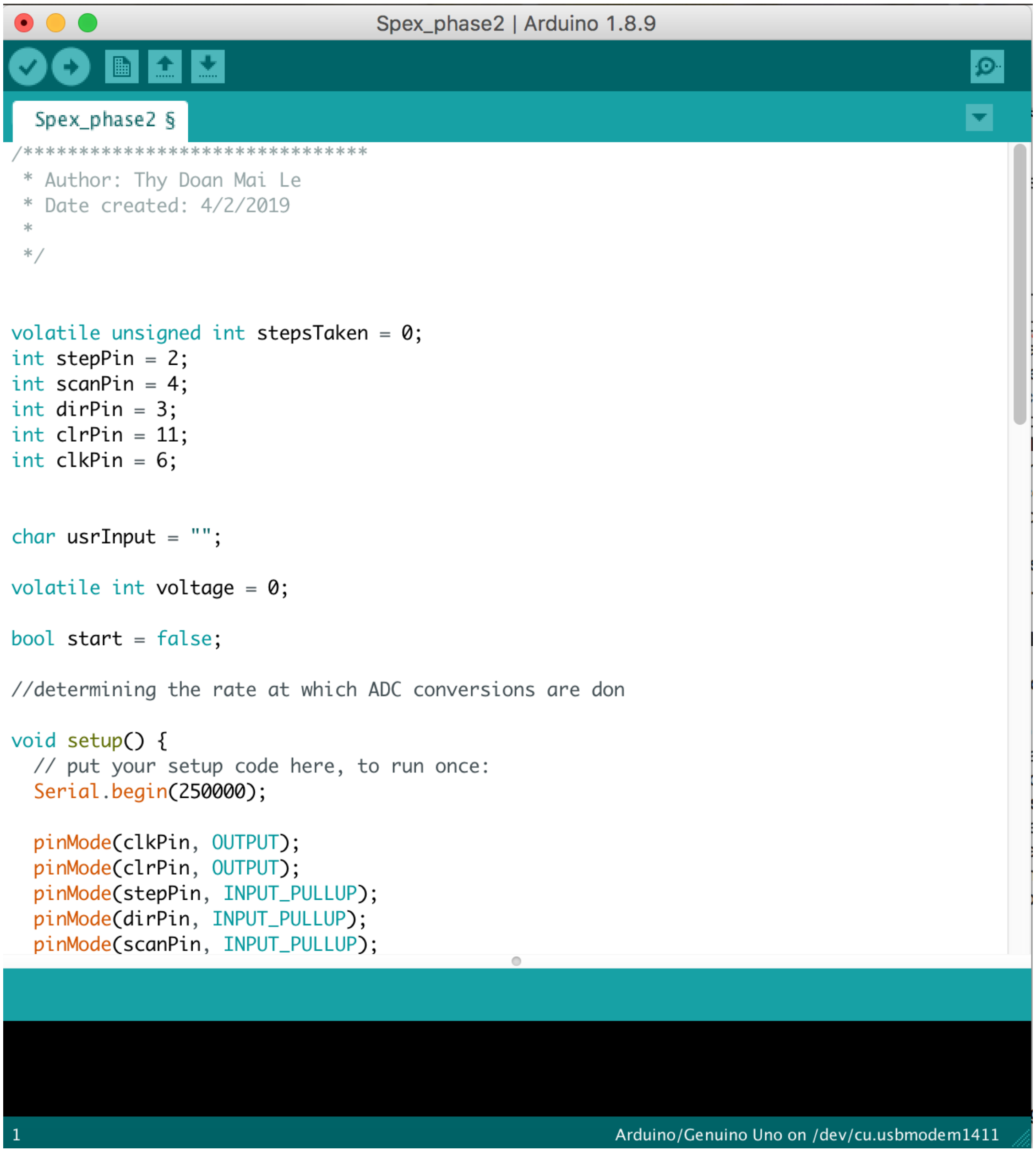

Figure 5: The Arduino IDE has a text-editing space where the program for the Arduino can be written prior to uploading to the board. When the program is ready, the program would need to be compiled by clicking the checkmark button on the top left corner. Because the program is written in C language, the compiler would need to compile the C code, generate the corresponding assembly language file and the processor in the Arduino will translate this file into machine-code instructions. The reason for using C and compiler technology, instead of an interpretive language like Python, is primarily due to speed. Compiler languages are always faster. Once compilation is completed, the program needs to be uploaded onto the board by clicking on the arrow button, next to the compile button, before the program can be executed by the Arduino.

the photomultiplier tube, which is a device that outputs a photocurrent every time a photon is detected at the spectrometer's exit slit.

In order to eliminate the need for the NI-DAQ board in data collection, I built a signal processing circuit to process the signal from the photomultiplier. To read the processed signal, I programmed an Arduino to measure this voltage at intervals that can be specified by the user. In order to capture the data on a computer for further analysis, I wrote a Python program that can receive data from the Arduino's serial line and output a graphical user interface window to live-plot data as it comes in from the Arduino.The NI-DAQ board is then completely bypassed. The user is expected to enter the initialization values for a scan, such as starting position, stop position and data acquisition rate, on the left sidebar of the Python window to initialize the program, and on the scan controller to initialize the spectrometer prior to every scan. Further details on the spectrometer's scans will be discussed in a later section.

2. Design

2.1 General Description of the Diffraction Grating Spectrometer

A spectrometer is a device that takes advantage of the principle of multiple-slit diffraction in order to separate individual wavelengths of light within a continuous spectrum. In multiple-slit diffraction, the pattern's maxima, or bright fringes on a screen, are known to occur when

$$d \sin\theta = m\lambda, \tag{4}$$

where θ is the angle between the incident light ray and the normal to the plane of the screen (where the pattern illuminates), $d \sin\theta$ is the path difference between two incident rays from adjacent slits and m is the order of the maximum.

A diffraction grating is required to produce a multiple-slit interference pattern. The diffraction grating is a planar surface with grooves that are edged on closely together and are equally spaced from one another. When light is incident on a grating, each groove on this grating acts like a single open slit and forces the incident light to have path differences, which ultimately leads to an interference pattern. For a spectrum consisting of multiple wavelengths, this pattern is a spread of the component wavelengths of the spectrum.

There are two types of gratings, transmission and reflective, that are used in spectrometers. The transmission grating allows light to pass through as it diffracts and the resulting pattern is often projected onto a screen. The spectrometer at IC Physics has a Czerny-Turner mount on its optical table, which requires a reflective grating instead of a transmission grating. Therefore, I will dedicate the remainder of this sub-section to discussing the details of the mount along with how a reflective diffraction grating changes the geometry of diffraction.

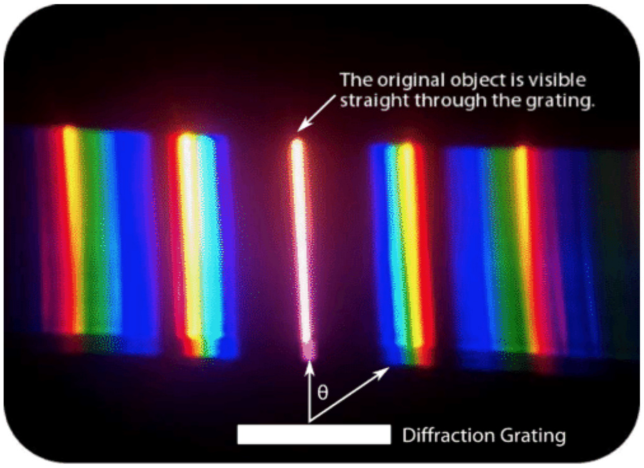

Figure 6: [5]When a spectrum of continuous wavelengths passes through a diffraction grating, each wavelength will undergo its own diffraction pattern as seen here in this figure. For example, the wavelength for the red line underwent diffraction to show 1st-order maxima on both sides of the zeroth order, and fainter 2nd-order maxima after the first. Because the incident light comes in as a band of wavelengths, the diffracted light also comes out as a band of wavelengths but each diffracted wavelength has its own angular position so our eyes are able to see the spread of the outgoing light.

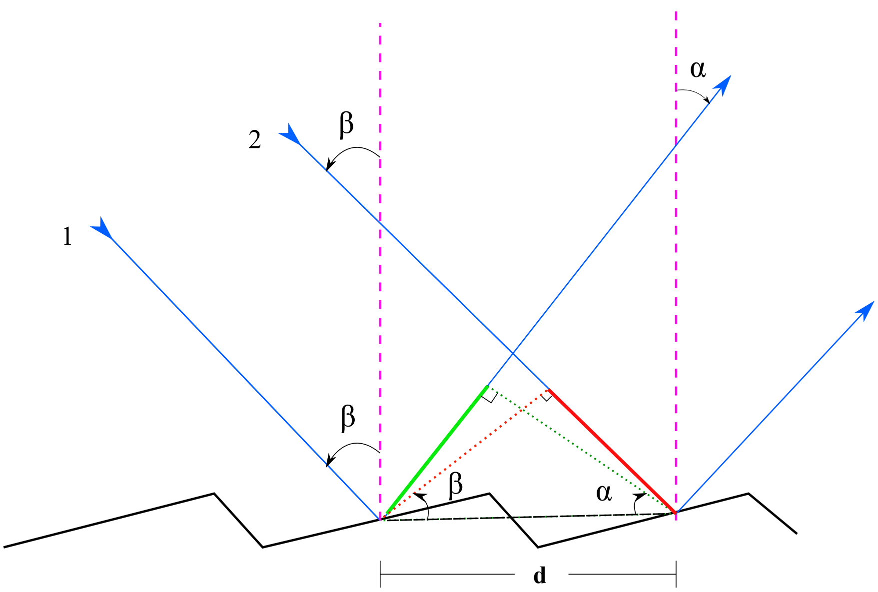

Figure 7: A side view of the path of travel of incident light upon hitting the surface of a reflective grating. β is the angle between the incident ray and the normal of the grating. α is the angle between the diffracted light ray and the normal to the grating. The path difference between ray 1 and 2 is now the sum of the path difference between the incident rays and the path difference between the diffracted rays, which is the sum of the red and green segment.

The Czerny-Turner mount, as referenced in Fig. 8, involves a collimating mirror, a rotating, reflective grating and a focusing mirror. Light entering the spectrometer is first guided towards the collimating mirror, which ensures that incident light on the grating would always be collimated, then towards the rotating grating, then to the focusing mirror before it exits at the exit slit. Because the focusing mirror could only reflecting a small part of the diffraction pattern that is coming from the grating, there is a stepper motor underneath the grating that rotates to turn the grating, so that other parts of the diffraction pattern can also reach the focusing mirror and exit the optical table.

A reflective grating is different from a transmission grating in that it reflects the diffraction pattern instead of allowing the pattern to pass through. Thus, when a reflection grating is used, the geometry of diffraction deviates slightly from Eq. 4 with the maxima occuring under the conditions of:

$$d\,(\sin\alpha + \sin\beta) = m\lambda \tag{5}$$

where $m = 0, \pm 1, \pm 2, ...$; d is the spacing between rulings on the grating, α is the angle between the grating's normal and the diffracted light ray and β is the angle between the grating's normal and the incident light ray. The grating normal changes as the grating is rotated during a scan and thus both α and β change according to the rotation of the grating.

The stepper motor, however, does not operate according to α and β. It operates based on an angle θ, shown in Fig. 8, that is measured from the absolute resting position of the grating to where the grating's current position. From the geometry in Fig. 8, we can let γ be a dummy angle variable and say that:

$$\begin{aligned} \theta + \gamma + \varphi &= 90^\circ \\ \alpha + 2\varphi + \gamma &= 90^\circ \end{aligned} \tag{6}$$

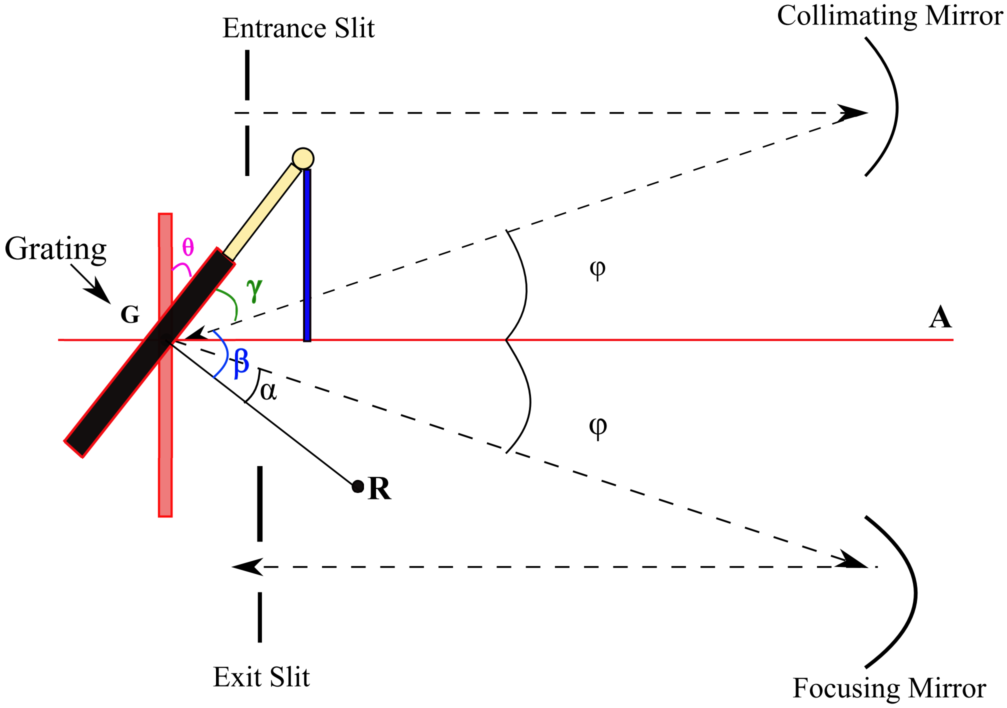

Figure 8: A birds-eye view of the Czerny-Turner mount inside of the spectrometer. In this configuration, light enters the entrance slit, hits the collimating mirror and travels towards the grating at an angle ϕ with respect to line GA in the figure. Line GA is the normal to the grating when the grating is perfectly parallel to the red bar, so this normal GA stays fixed. The blue rod is a screw that can lengthen or shorten itself in order to lower or raise the beige bar that it is holding up. Lowering and raising the beige bar effectively rotates the grating and thus, changes the rotating angle θ. α and β are the same angles as described in Fig. 5 and change with the rotation of the grating. In contrast, φ angles remain constant throughout the rotations because the mirrors are fixed. Angle γ is measured between the grating and the incoming light ray from the collimating mirror and will become significant later during the determination of the grating equation for a Czerny-Turner mount.

By subtracting both equations from each other, we get:

$$\theta - \alpha - \varphi = 0 \tag{7}$$

which, by rearranging, we can extract an expression for α to be

$$\alpha = \theta - \varphi \tag{8}$$

Similarly, for β, we initially have two equations

$$\begin{aligned} \beta + \gamma &= 90^\circ \\ \theta + \gamma + \varphi &= 90^\circ \end{aligned} \tag{9}$$

By subtracting the two above equations, we get

$$\beta - \theta - \varphi = 0 \tag{10}$$

which, from rearranging, we can get an expression for β

$$\beta = \theta + \varphi \tag{11}$$

From Eq. 8 and Eq. 11, we can say that:

$$\begin{aligned} \alpha &= \theta - \varphi, \\ \beta &= \theta + \varphi. \end{aligned} \tag{12}$$

Since

$$m\lambda = d\left(\sin\alpha + \sin\beta\right), \tag{13}$$

we can substitute in the new expressions for α and β from Eq. 12 to obtain

$$m\lambda = d[\sin(\theta - \varphi) - \sin(\theta + \varphi)]. \tag{14}$$

Using the following trigonometric properties:

$$\begin{aligned} \sin(A + B) &= \sin A \cos B + \cos A \sin B, \\ \sin(A - B) &= \sin A \cos B - \cos A \sin B, \end{aligned} \tag{15}$$

we can simplify Eq. 14 to

$$m\lambda = 2d \cos\varphi \sin\theta, \tag{16}$$

where θ and φ are both described and denoted in Fig. 8. Because ϕ is constant, the term $2d \cos\varphi$ can be regarded as a constant k and the above equation can be rewritten as

$$m\lambda = k \sin\theta \tag{17}$$

Thus, by moving the GR arm in Fig. 8, angle θ will change as a result and according to Eq. 17, the outgoing wavelength λ towards the exit slit will also change. This is the mechanism behind what is called a Czerny-Turner mount of a spectrometer and is exactly how the spectrometer at IC Physics operates.

2.2 New Design for the Spectrometer

2.2.1 Monitoring the Stepper Motor

In a spectrometer with a Czerny-Turner mount, the reflective diffraction grating is rotated by a stepper motor in order for the entire diffracted spectrum to sequentially exit the exit slit. The exit slit in the spectrometer is an adjustable narrow opening that only passes a portion of the diffraction pattern through at each angular position of the grating. When the grating rotates, the diffraction pattern will move across the focusing mirror in Fig. 8, wavelength by wavelength, and all parts of the diffraction pattern may sequentially pass through the narrow exit slit. For this spectrometer, every trial, in which the grating rotates through a specific range of user-defined wavelengths, is called a scan. Because a scan can start and end at any wavelengths that are within the limits of the spectrometer, the user is required to specify the start and end, as well as the rate of a scan in Å/s, at the beginning of each scan.

Currently, the stepper motor of the spectrometer is controlled by a device called the Scan Controller that has a keyboard and an information display with six LEDs. The keyboard is used to enter parameters to initialize each scan, such as the starting wavelength in Å, the final wavelength and the scan rate. The LED screen displays the input selection as the user types in their inputs. It also displays the current scanning wavelength based on the angular position of the diffraction grating during a scan. On the back of the scan controller are digital output pins that will go high or low depending on what the scan controller is doing. Further details on the pin numbers and pin names can be found in Fig. 9. By monitoring the digital output at pin 9, pin 10 and pin 11, shown in Fig. 9, the Arduino can monitor how many steps the stepper motor has taken, whether the spectrometer is stepping forwards or backwards, and whether a scan is in session, respectively. For this spectrometer, I found that the stepper motor takes 400 steps for every Å by monitoring the STEP pin at the status port on an oscilloscope. Therefore, I decided to program the Arduino to do a voltage measurement once every 400 steps by default, although the user can increase the rate of measurement through the Python interface for increased precision. To measure the voltage once every 400 steps, the Arduino continuously monitors the STEP pin, records the amount of steps taken and takes a measurement every time 400 steps, or 1 Å, has been stepped through. At the end of the measurement, the Arduino also refreshes its step count to get

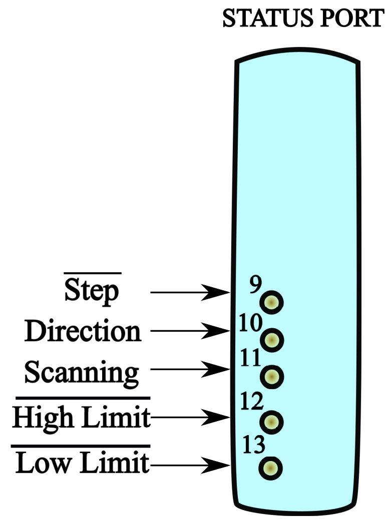

Figure 9: Status port at the back of the scan controller with only significant digital output pins featured. There are 25 output pins on this status port but only 5 of these pins are necessary for the Arduino to monitor the scan process. Pin 9 is an active-low pin that becomes negative every time the stepper motor inside the spectrometer takes one step, regardless of direction. Pin 10 is an active-low pin that becomes negative when the spectrometer is stepping backwards in terms of wavelength, and positive when the spectrometer steps forward. Pin 11 is an active-high pin that becomes positive when a scan begins and stays positive as long as a scan is in session.

ready for the next cycle of stepping. The Arduino is programmed to increment steps taken whenever the STEP pin pulses low, however, the STEP pin pulses low not only during a scan but also during the initialization process prior to a scan. To prevent the Arduino from mistakenly incrementing steps when a scan is not running, the SCAN pin, which remains low when during a scan, is monitored so that the Arduino would increment steps appropriately.

2.2.2 Photomultiplier Tube to Arduino

Outside of the exit slit of the spectrometer sits a photomultiplier tube that is responsible for detecting photons leaving the exit slit. When a photon is detected by the photomultiplier, a photocurrent is generated [3]. The photomultiplier is a vacuum tube with a photocathode that is excited whenever a photon hits it [2]. As a result of the excitation, free electrons are released and then accelerated by a high potential. This potential is supplied by the high voltage supply shown in Fig. 13. These free electrons are then used to generate secondary

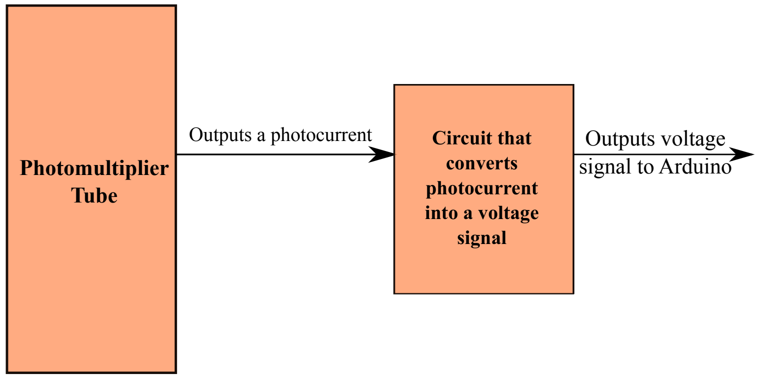

Figure 10: The photocurrent from the photomultiplier first enters an external circuit that converts the current into a voltage signal. Then, this voltage signal is put into the Arduino for measurements. Because this voltage signal is analog (smooth and non-discrete), it will have to go into the Arduino via one of the Analog input pins depicted in Fig. 4

electrons on other electrodes inside the photomultiplier until a photocurrent is created. The amplitude of the photocurrent is proportional to the amount of photons that is detected at the exit slit and depends on the responsivity of the photomultiplier tube. Per the manufacturer's guidelines, the responsivity of the photomultiplier tube of our spectrometer is roughly 50% through the visible spectrum, from 400nm to 800nm. The photocurrent travels from the photomultiplier to the signal processing circuit, where it is processed, before the Arduino performs a voltage read on the converted voltage signal. This whole process is summarized in Fig. 10. During every voltage read, the Arduino will look at the processed DC voltage input from the photomultiplier, do an analog-to-digital conversion on this voltage and send the digital reading result to the laptop that is controlling the Arduino.

2.2.3 Data Acquisition

Once the Arduino has captured the voltage signal from the signal processing circuit in Fig. 11, the Arduino transfers the value of that voltage signal to a Python program on the user's laptop. This program gathers the incoming data from a USB connection with the Arduino, stores the data and performs a live plot of the incoming data.

The Python program consists a procedure to create a user interface window upon startup,

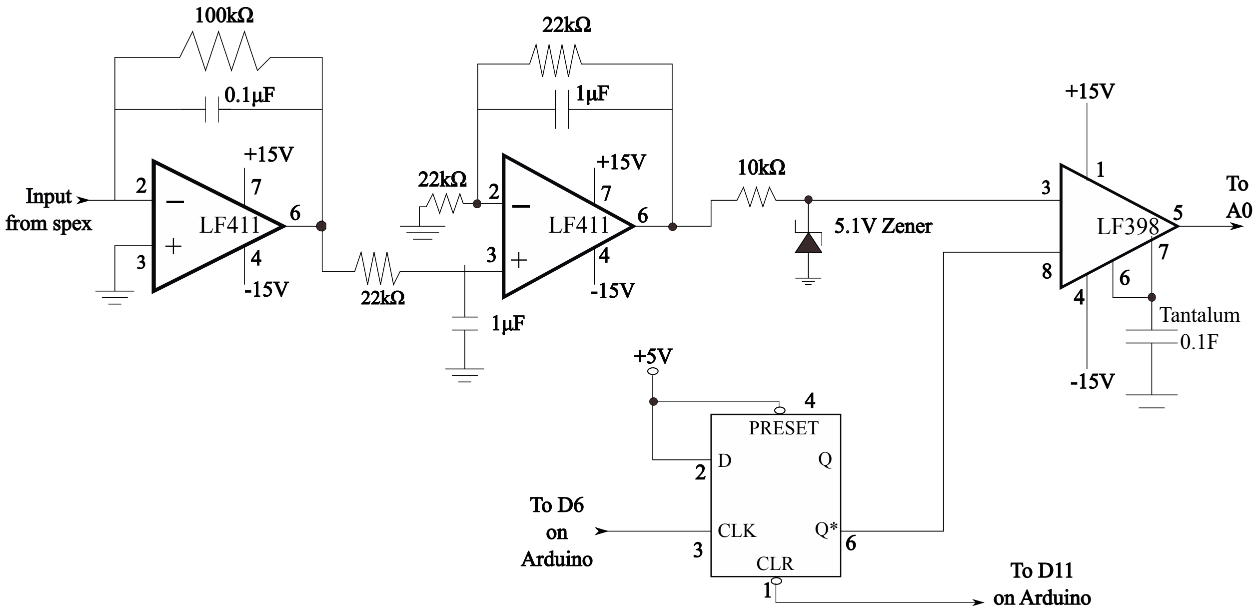

Figure 11: The signal processing circuit includes two op-amp (LF411) circuits, a D flip flop (74LS74) and a Sample and Hold op-amp (LF398). Instead of a voltage signal, the photomultiplier outputs a photocurrent with an amplitude that is proportional to the amount of photons which were detected at the exit slit. This photocurrent is first converted into a voltage signal at the first op-amp (LF411) circuit. Then, the newly converted voltage signal is buffered at the second op-amp in order to clean up the signal. The signal then enters a clipper circuit that prevents the input signal from having a larger-than-5V amplitude, in order to protect the Arduino's analog-to-digital converter. After the clipper, the signal proceeds into the input pin 3 of the Sample and Hold op-amp, whose input pin 8, the Hold pin, has an input from the D flip flop's Q* output. The Flip Flop has DATA and PRESET connected to 5V. When the CLK pin pulses high, pin Q is set to DATA and pin Q* is reset to low. The CLR pin is used to reset pin Q to low and pin Q* back to high. When the spectrometer reaches 400 steps in a count cycle, the Arduino pulses the CLK pin to high, thus setting pin Q to DATA and pulling pin Q* to low. Because pin Q* is connected to the Hold pin of the Sample and Hold, the Hold pin is also low at this point. When Hold pin is LOW, the Sample and Hold maintains its output signal at pin 5 to be exactly as the input from pin 3. This is when the Arduino will perform an Analog Read, which is the Arduino's way of reading voltages, on the output from the Sample and Hold's pin 5. Once the Analog Read is complete, the CLR pin on the Flip Flop is pulled LOW, effectively clearing the Q* output and thus the Hold pin. The Sample and Hold will only exit out of this Hold mode once the Hold pin is pulled HIGH again.

routines that are connected to buttons on the interface window and a procedure to gather data from the Arduino. The interface window has a graph panel where the incoming voltage values from the Arduino can be plotted in real-time against the current scanning wavelengths. On the left sidebar of the user window, there are text fields for the user to enter the start and stop wavelengths of each scan. Below the text fields, there are radio buttons for the user to choose their desired measurement (i.e.: 2 measurements per Å). Lastly, the Python program has a set of buttons for control over the workflow of the program, such as "Start" to start one scan of a spectrum, "Stop" to stop the scan before the final scanning wavelength is reached, or "Save" in order to save the graph of voltage versus wavelength to an image file once a scan of a spectrum is complete. The routines to be included in the Python program are:

- SetUp: sets up the windows, buttons, text fields and connection between the Arduino and the program.
- OnIdle: directs the Python program to capture and plots incoming data from the Arduino during idle, when data will be pouring in from the serial line of the Arduino and no button is being pressed.
- ConnectToArduino: sets up a connection with the Arduino, checks whether the correct Arduino program is open and signals the Arduino program to start. If a connection to an Arduino cannot be made, an error message pops up.
- OnStart: event function for when the "start" button is clicked on the main interface window.
- OnStop: event function for when the "stop" button is clicked on the main window.
- OnSave: event function for when the "save" button is clicked on the main window.
- Draw: function that instructs Python on how to plot the incoming data and how to format the graph panel.

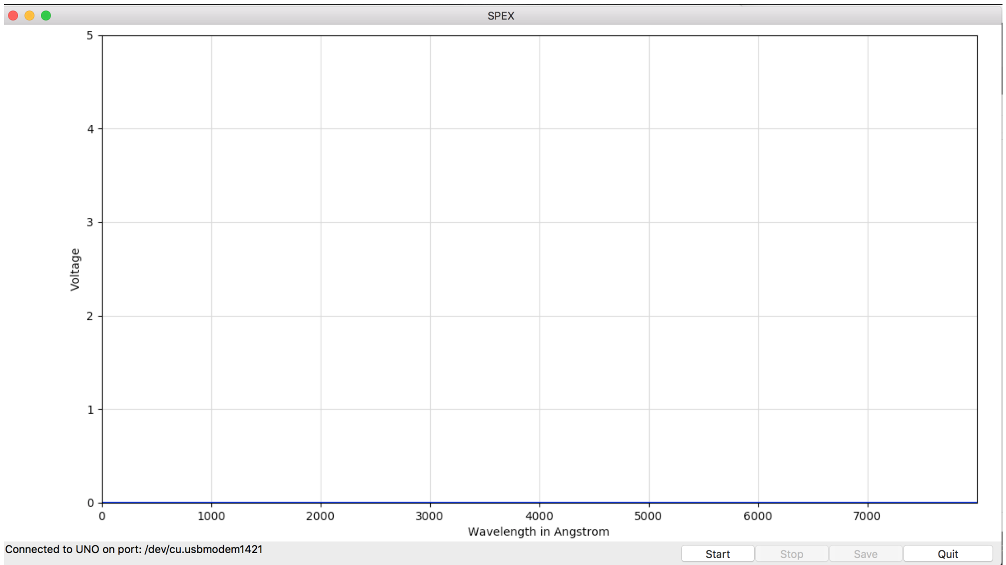

Figure 12: An example of the Python GUI window. The receiving voltage is plotted against the current wavelength in angstrom. The functions OnStart, OnSave, OnQuit, OnStop will be binded via wxPython to each of the buttons that are shown on this window.

- OnQuit: event function for when the "quit" button is clicked.

Each of these routines is required to be bounded to either an event or a button. For example, the OnStart routine is bounded to the "Start" button on the interface window, so that the OnStart routine can be called once the "Start" button is pressed. Routines can also be bounded to events that have built-in definitions by Python. A few examples of these built-in events are IDLE, which is defined as the state in which the program experiences no button pushes or user inputs, or CLOSE, which is defined as the state in which the interface window is closed. A routine can be bounded to a built-in event, so that when the program detects a user-initiated change (i.e.: idle, window is closed, etc. . .) corresponding to a built-in event, the program can call the proper routine for the program to move forward.

3. Assembly and Troubleshooting

3.1 Assembly

Once the new design was completed, I proceeded to assemble the new design and troubleshoot as the next step in the project. In this section, I will discuss the details of the implementation of the new design as well as my troubleshooting process. It is my hope that the content of this section, with the implementation details and records of my troubleshooting process, would assist any plan for developments, should a student decide to take on a project for future work.

3.1.1 Monitoring the Stepper Motor

The Scan Controller has specific pins on a status port, depicted in Fig. 13, which connects to individual registers that monitor the scan controller's current operations. There are 25 pins on the status port, but for this project, the important pins that we will focus on are the STEP, SCANNING and DIRECTION pins. The STEP pin is an active-low pin that becomes low (0V) every time a step is taken by the stepper motor inside of the spectrometer. The Arduino is able to know precisely when to perform a voltage read (at steps = 400) by monitoring this STEP pin and incrementing steps every time the STEP pin pulses low. The SCANNING pin is an active-low pin that is low when the stepper motor is performing a scan at a prescribed scan rate in Å per seconds by the user. The DIRECTION pin is active-low and turns active when the stepper motor is stepping forward, inactive (or high at 5V) when the stepper motor is stepping backwards. The Arduino is set up to monitor all 3 lines and a table summary of this setup can be found in Table. 1. The Arduino is currently able to take one voltage reading every Å, however, the user is able to adjust the Arduino's program to capture multiple voltage readings every Å for more precise results.

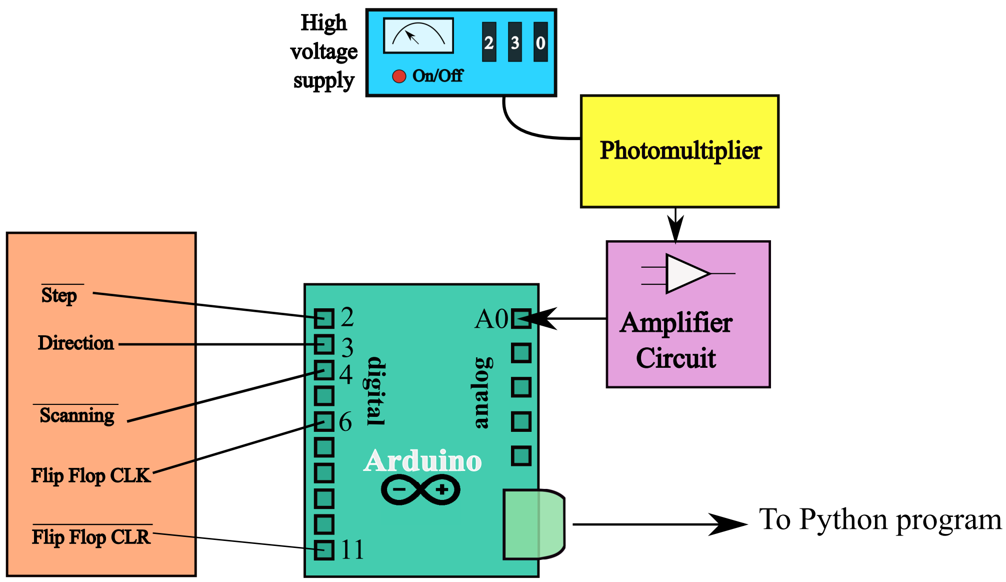

Figure 13: The general set-up of the new apparatus which shows how the Arduino will be wired to the scan controller and the amplifier. The Step, Scanning and Direction lines on the scan controller's status port must be connected to the Arduino's digital pins for the Arduino to monitor the scan controller. The photomultiplier outputs a photocurrent that requires processing by the amplifier circuit. After processing, this signal is put into pin A0 of the Arduino and waits to be measured.

Status Pin	Arduino Pin	Function
Scan	4	Active LOW Monitors whether a scan is being performed
Step	2	Active LOW Pulses LOW when a step is taken by the stepper motor Is used to trigger interrupts to keep track of steps taken
Direction	3	Active HIGH Sets HIGH when spectrometer is moving forward (incrementing) in wavelength Sets LOW when spectrometer is moving backwards (decrementing) in wavelength

Table 1: The Scan and Step pins are active LOW whereas the Direction pin is active HIGH. The Step pin is pulsed to LOW every time a step is taken by the stepper motor. The Scan pin is set to LOW throughout the duration of a scan. The Direction pin is HIGH when the spectrometer is scanning towards larger wavelengths, LOW when the spectrometer is scanning towards lower wavelengths.

In order to monitor the current wavelength, digital pin 2 is currently connected to the Step pin on the status port. When Step pulses low, the Arduino will enter an interrupt on the falling edge of Step to increment to amount of steps that the spectrometer has taken. Once 400 steps have been recorded, the Arduino performs a voltage read and sends the result to the user's laptop via the serial line. At this point, the Arduino also refreshes its step counter to begin another cycle of counting up to 400 steps, because this spectrometer's stepper motor has a resolution of 400 steps per Å. The Direction pin is attached to pin 3 on the Arduino and Scan is attached to pin 4. Step has to be attached to pin 2 because pin 2 and pin 3 are the only two pins that are capable of triggering external interrupts, which means that the Arduino enters an interrupt routine upon the cue of an external signal. However, pin 2 and pin 3 are also able to function as a normal pin. For this project, Step is attached to pin 2 for external interrupts while pin 3 remains a normal input pin for the signal from Direction.

3.1.2 Photomultiplier to Arduino

When photons are detected at the exit slit, a device called the photomultiplier, as referenced in Fig. 2, produces a photocurrent with an amplitude that is proportional the amount of photons that was detected. Previously, the photomultiplier had an amplifier circuit with 2 op-amps that were used to convert the outgoing photocurrent into a voltage signal and to amplify that signal for better resolution. This voltage signal would then be fed into a NIDAQ card inside of the original Mac computer that used to control the spectrometer.

For my new design, the photomultiplier's signal is first processed in an amplifier circuit, as referenced in Fig. 11, before entering a clipper circuit to ensure that the signal does not exceed 5V. After the clipper circuit, the signal then enters a circuit called a sample and hold op-amp. Because the Arduino reads incoming voltages as the scan occurs, the voltage level could potentially change while the Arduino is doing a measurement. This sudden change could result in unwanted fluctuations in the analog-to-digital conversion results and create unwanted spikes in the data. In order to avoid changes in the input signal, a sample and hold op-amp is implemented prior to the Arduino stage. A sample and hold op-amp has two modes of operation: sample mode and hold mode. When it is in sampling mode, the circuit acts like an op-amp voltage follower where its output is exactly the same as its input. The mode of the sample and hold is controlled by the Q* pin from the Flip Flop in Fig. 11. Pin Q* outputs a low signal Once hold mode is triggered by a low (0V) digital signal on pin 3 by the Flip Flop, as pictured in Fig. 11, the circuit holds the output at the voltage level that was at the input at the start of hold mode. The circuit will only enter sampling mode again once the hold pin is released back to high (5V). By holding the input signal at a fixed level, the Arduino will be able to avoid fluctuations during measurements.

3.2 Troubleshooting

Throughout the duration of the project, I had encountered numerous difficulties that arose due to either the setup of the spectrometer, incompatibilities between different electrical components or simply software troubles. In the following paragraphs, I outlined the most significant problems that I troubleshooted throughout this project, with hopes that this section could provide guidance for those who would like to implement further developments to the apparatus. In addition, this section is meant to serve as a reference to the major problems that I encountered during my work on this project, along with solutions that I attempted on these problems.

3.2.1 Op-Amp Troubles

One of the challenges that comes with using op-amp circuits is the variety of unexpected problems that op-amps can bring. Despite their usefulness and robustness, even JFET op-amps, a widely used low-noise op-amp, can have a lot of problems, including input offset voltages, common mode rejection ratio, offset voltage drift, etc.... Sometimes, the problem could simply as simple as grounding issues (typically coming from oscillations within the power supply to the op amps that eventually become amplified according to the gain of the op-amp) or the op-amp has been burned out. At other times, the problem is much more subtle and required further investigation, which could be very costly in terms of time.

One of the most common op-amp troubles is an input offset voltage. Op-amps are originally designed to have two balanced positive and negative inputs. However, manufactured op-amps never quite reach this ideal situation, therefore, most op-amps typically have slightly unbalanced inputs. This asymmetry from the inputs can cause an offset voltage to occur at the inputs. If the op-amp is used in a circuit with a small gain, for example with a 20 gain, this input offset voltage might be insignificant in the end. However, if the op-amp is used in a circuit with a large gain, say 100, the input offset voltage would be amplified by 100 times and severely alter the expected output voltage of the circuit. In order to correct this problem due to asymmetrical inputs, you can utilize a trimmer, which is an adjustable resistor, on pins 1 and 5 of the op-amp. A trimmer has three pins: two of them are called "legs" and the remaining pin, the "wiper", is connected to the negative power supply of the op-amp. You would need to ground both of your inputs and then connect the two "legs" of the trimmer to pins 1 and 5 on the op-amp. At this point, for an op-amp with symmetrical legs, the output signal of the op-amp should give 0V because both inputs are grounded. For an op-amp with asymmetrical legs, however, the output signal at this point should be slightly more than 0V due to the input offset voltage. Once everything is connected, turn the knob on the trimmer until the output signal shows exactly 0V. When that occurs, the input offset voltage has been eliminated.

Another common op-amp problem is noise. This category of trouble is more vague than input offset voltage because noise can originate from just about anything. Wall outlets and electrical lines inside of buildings typically give off 60Hz noise that can be easily picked up by any circuit. Power supplies are also notorious for producing noise. It is extremely difficult to completely get rid of noise in a circuit, but when the correct noise sources are detected and dealt with, the majority of the noise can be significantly reduced. In order to diagnose the correct sources of noise in the circuit, begin with isolating components within the circuit. Typical sources of noise include other op-amps and power supplies, so it is suggested that the troubleshooting process be started with isolating the op-amps in the circuit away from one another. Usually, the power supply is the main culprit for noise. If that is the problem, put $1\mu F$ capacitors between the positive power output (+15/+12V line) and ground as well as between the negative power output (-15/-12V line) and ground. The capacitors will provide a bypass route for any oscillations, which are the precursor to noise in op-amps, to go directly to ground instead of into the op-amps' power lines.

Sometimes, your problem can be very straightforward and your op-amp could simply be a bad one. Op-amps stored in CNS 205's drawers are communal op-amps that might have suffered previous damage from other users before you got them. If you have tried every single fix and your circuit is still misbehaving or producing wildly unexpected output signals, it would be my recommendation that you replace your op-amps as the next step.

3.2.2 Python Program

Python programs are famous for their user-friendliness: the syntax is very relaxed, it is not a typed language and there are over hundreds of built-in libraries that users can easily use without knowing the implementation. This is one of the reasons why Python was chosen the language of choice when it came to implementing the graphical user interface for this project. However, because Python hides most of its implementation from the user, subtle errors, which would typically be detected in more low-level programming languages, can become exacerbated in Python. During this project, the most significant problem with Python that I encountered was with the PATH of the Python executable.

Python programs are interpreted because Python is an interpretive language and not a compiled language. Therefore, once a Python program is opened from whatever chosen platform, whether this is the bash shell (terminal) or an Intergrated Developing Environment (IDEs such as PyCharm by JetBrains), an interpreter that is located at a built in path is called. Depending on the PATH that is configured in your environment, any number of Python interpreters can be called. A machine can have more than one Python interpreter, and choosing the correct Python interpreter is crucial to getting the Python interface running. The primary reason for this is because the Python interface for this experiment requires specific packages, such as wxPython, pySerial and numpy, to be installed at the same location as the Python interpreter so that the interpreter can expand these packages during the program's execution.

Throughout the first half of the project, I was unable to correctly open the window for the interface because my PATH for the Python interpreter was to the native Python that is installed with most Macbooks. This is **not** the Python interpreter that you should be using because of the limitations that Apple has placed on this Python version. In order to overcome this problem, I consulted with professor Toby Dragon from the Department of Computer Science and per his instruction, I installed Homebrew, a package manager for Python, in order to have my own version of a different Python interpreter on my computer. Homebrew, as a package manager, will install a more complete Python interpreter in `/usr/local/Cellar` and every single package that is later installed would be directed towards this very same directory. By making sure that everything is in one specific directory, Homebrew ensured that once the Python interpreter in `usr/local/Cellar` is called, it is able to find all of the necessary packages to expand for the program. Once I completed this setup process, I was finally able to open the Python interface.

4. Testing & Data Analysis

To verify that my design is capable of reproducing the spectrometer's original functionality, I used the completed apparatus to observe 2 types of spectral sources, a blackbody source (continuous light) and an elemental source (a helium Geissler tube), and compared the captured spectra with theoretical data in order to confirm that the apparatus is able to produce the correct spectra for the chosen sources.

$$L_\lambda = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1} \tag{18}$$

A blackbody source of electromagnetic radiation is a source that produces wavelengths of varying intensity over a continuous range. In a blackbody spectrum, the shape of the spectrum is determined by the temperature of the blackbody radiation source according to Eq. 18. For a very high-temperature source, the peak of the blackbody spectrum tends to occur in the lower wavelengths, typically below 5000Å. As the temperature of the source decreases, the peak of the blackbody spectrum migrates towards longer wavelengths. Therefore, determining the peak intensity of a blackbody spectrum is one way of estimating the temperature of the spectral source that produced the blackbody spectrum. An elemental source, such as a sodium vapor lamp or a helium Geissler tube, produces a discrete spectrum instead of a continuous one. This discrete spectrum consists only of high peaks in intensity at specific wavelengths while the remainder of the spectrum has almost zero intensity. Most spectrometers should be capable of resolving both continuous and discrete spectra and thus, in order to verify the complete functionality of this apparatus, both types of spectra were used in testing.

4.1 Continuous Spectrum

For a test with a blackbody spectrum with an unknown temperature, the captured spectrum showed a peak in intensity at 6000Å as shown in Fig. 14.

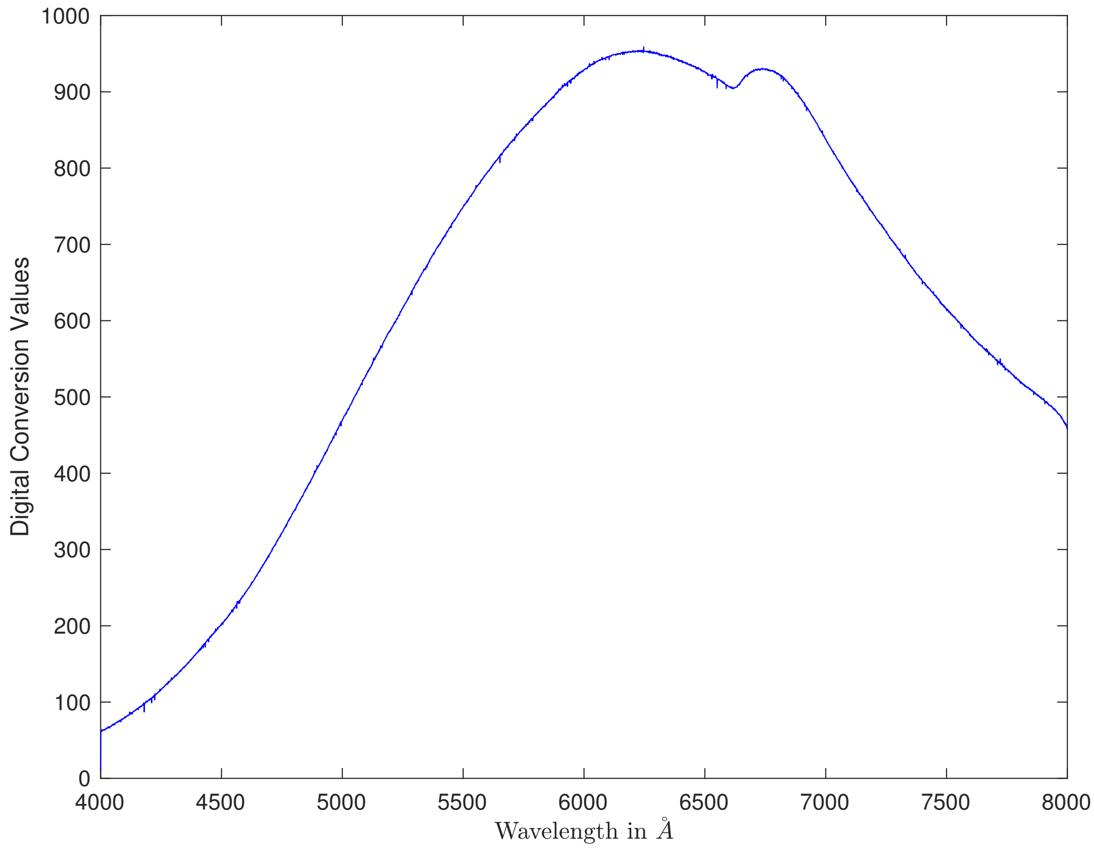

Figure 14: The captured spectrum of a blackbody source with unknown temperature showed a peak in intensity at 6000Å, which is 600nm. The high voltage supply to the photomultiplier was set to be 230V. The blackbody's power supply was set to have a current of 6.2A and a voltage of 5.8V. The intensity is measured in terms of the analog-to-digital conversions by the Arduino on the processed voltage signal leaving the circuit in Fig. 11. This signal, in turn, has an amplitude that is determined by the photocurrent's amplitude at the photomultiplier. The photocurrent's amplitude is related to the amount of photons that were detected at the exit slit.

In order to estimate the temperature of the spectrum, a theoretical blackbody spectral intensity plot was generated using the blackbody radiation intensity function, Eq. 18 [7], to

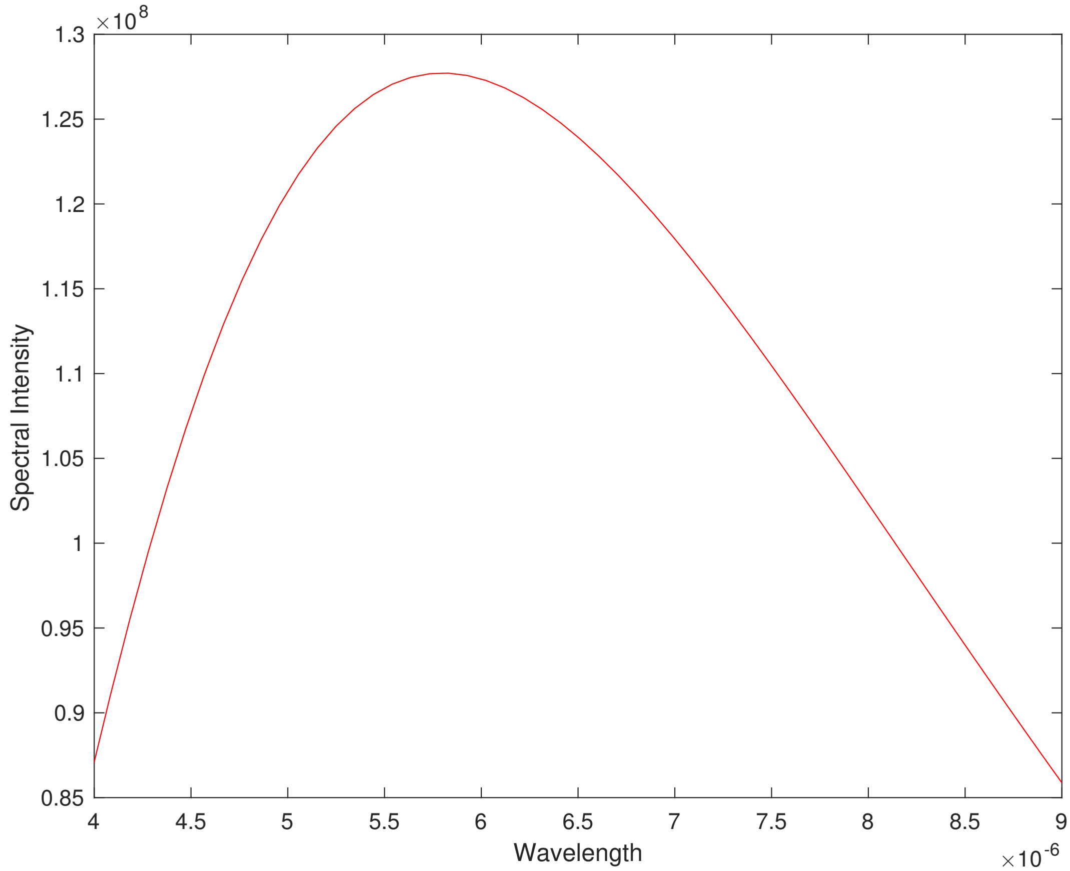

Figure 15: The theoretical spectrum shows a theoretical peak of intensity at around 600nm which corresponds to 6000Å. From Eq. 18, the temperature of this blackbody is extrapolated to be roughly 500K.

obtain Fig. 15.

When Eq. 18 was plotted with T = 500K, the theoretical spectral intensity plot showed a peak at $5.5 \pm 0.2 \times 10^{-6}$m, which is essentially 5500 ± 200Å. This peak overlapped with the peak in the experimentally measured spectrum of the blackbody source with unknown temperature. Due to this overlapping, I can conclude that the blackbody source must have a temperature of 500 ± 100K. This result also indicated that the apparatus was able to successfully capture a continuous spectrum.

4.2 Emission Spectrum

The apparatus' functionality was also tested with an emission spectral source. For emission spectrum testing, a Helium Geissler tube was used. The primary goal of this test is to ascertain whether the apparatus was able to capture discrete spectral lines by comparing the captured spectrum with previously measured spectral data that is published by NIST [4]. In the data that was provided by NIST in Table 2, the bright emission lines of the Helium spectrum are located at the wavelengths depicted in Table 2.

Wavelength(Å)	Relative Intensity	Measured Wavelengths
4471	200	4473
5015	100	5025
5875	500	5885
6678	200	6687

Table 2: [4]The relative intensities are ratios between the intensity at a specific wavelength and the intensity of the wavelength with the lowest intensity of the Helium spectrum. From the above distribution of intensities, the wavelength at 5875Å is shown to have the highest intensity in the spectrum and would be remarkably more visible than the remaining wavelengths in the table. The wavelengths that were captured by the spectrometer are depicted on the third column. The maximum deviation from NIST's wavelengths of the captured wavelengths is 10Å, which is an indication that the spectrometer is still very well calibrated despite not being used in over 15 years.

A scan using a Helium Geissler discharge tube, scanning from 4000Å to 8000Å showed peak intensities at 4473 ± 3Å, 5025 ± 10Å, 5800 ± 10Å and 6687 ± 9Å. The recorded peak intensities occurred at approximately the same wavelengths as the peak intensities which were measured by NIST in Table 2. This showed that the apparatus was also capable of capturing emission spectra. A plot of the captured Helium emission spectrum can be found in Fig. 16.

From both Fig. 16 and Fig. 14, I was able to conclude that the spectrometer's current apparatus is fully functional and ready to capture further spectra.

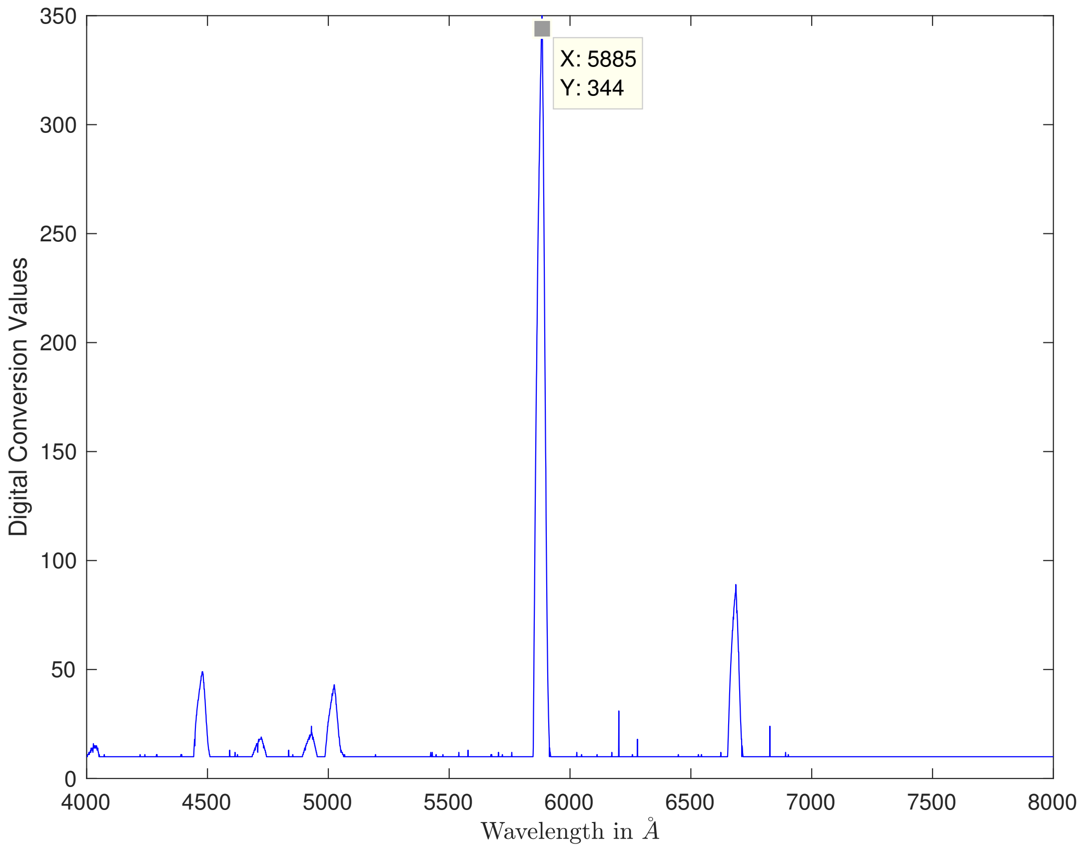

Figure 16: The captured spectrum of a Helium Geissler tube showed approximate intensity peaks at wavelengths that correspond with previously identified peak-intensity wavelengths by NIST in Table 2.

5. Conclusion

The project concluded with a fully functional diffraction grating spectrometer that is able to capture both discrete, emission and continuous, blackbody spectra using an Arduino and a signal processing circuitry. The signal processing circuit is responsible for processing the photocurrent from the photomultiplier, converting it into a voltage signal and triggering the sample and hold op-amp to hold the processed voltage signal steady for the Arduino to do voltage reads. The Arduino, in turn, monitors the steps of the stepper motor, which is located inside of the spectrometer, determines when to perform voltage reads, measures the voltage signal from the signal processing circuit and sends the result to the user's laptop. The Python program receives the incoming data from the Arduino and plots this data in real-time. Every scan is initialized with parameters that the user puts into the scan controller via the scan controller's keyboard. The user is also required to put in the same parameters into the Python program's start-up window for the program to initialize its graph panel and interface window.

The spectrometer, once completely decrepit and abandoned to collect dust, is now ready to capture the spectrum of any source as well as the interest of any student who wants to explore the mysterious world of spectroscopy.

5.1 Possible Improvements

Despite the apparatus' complete functionality at the present, there exists a lot of room for improvement. In the following paragraphs, I will discuss two major aspects of the apparatus that could significantly benefit from further improvements.

Currently, the signal processing circuitry, the status port in Fig. 9 and the Arduino are wired together in a very loose manner using standalone wires. This type of connection between multiple external components creates opportunities for wires to loosen, break or

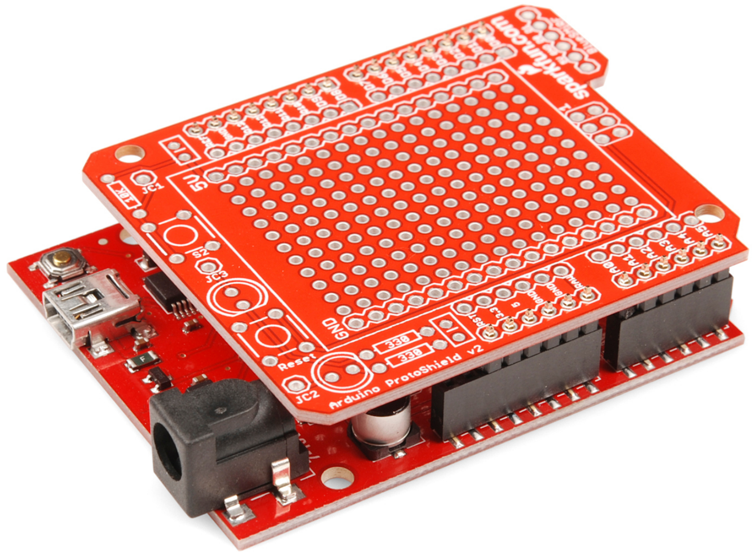

Figure 17: [8]A shield that could hold up to 5 electronic components and be placed right on top of an Arduino. By using a shield to hold electronic components, the need for dangling and loose wires, along with potential harm to the electronic circuitry, is eliminated almost completely.

being pulled out on accident over time. In order to prevent these problematic opportunities from arising, I suggest that the signal processing circuitry be mounted and wired onto an Arduino shield. This placement of the circuitry would completely eliminate the need for a circuit on a breadboard because the electronic components of the circuit would be soldered together and placed compactly next to one another on top of a shield.

Another area that could benefit from further development is the Python GUI interface. Currently, the Arduino and the Python program are in an agreement that voltage readings at the output of the photomultiplier would be taken at every Å. However, for experiments that require higher resolutions than 1 Å, it would be necessary for the programming on both the Arduino and the Python program to change in order accommodate for the higher resolution measurement. The Python program currently allocates an array of N elements for the incoming voltage measurements, with N being the difference between the starting and stopping wavelengths plus one. However, this N amount only represents data that is taken every 1 Å. For a higher rate of data collection, for instance 3 or 5 measurements/Å, an array of size larger than N would need to be allocated for data collection.

References Cited

[1] Arduino. https://reference.wolfram.com/language/ref/device/Arduino.html. Accessed Online on November 28, 2018.

[2] Photomultiplier. https://www.rp-photonics.com/photomultipliers.html. Accessed December 2nd, 2018.

[3] Spectrometers. https://www.rp-photonics.com/spectrometers.html. Accessed November 30, 2018.

[4] Strong Lines of Helium. https://physics.nist.gov/PhysRefData/Handbook/Tables/heliumtable2_a.htm. Accessed April 10th, 2019.

[5] *Diffraction Grating*. CK12 Science, June 2013. Last Edited: June 08, 2016.

[6] Spectroscopy and the Birth of Astrophysics. https://history.aip.org/exhibits/cosmology/tools/tools-spectroscopy.html, 2018. Accessed November 15, 2018.

[7] Glenn Elert. The Hyperphysics Textbook. https://physics.info/planck/. Accessed April 9th, 2019.

[8] Jimblom. Arduino Shields. https://learn.sparkfun.com/tutorials/arduino-shields/all.