Spectral Telescope for Star Classification by Main Sequence (STSC-MS)

Spectrometry and Imaging Automation for Astronomy

The Orion Nebula: Photo taken by Addison Long

Group 3 Authors:
Addison Long (PSE), Aaron Moreno (CPE), Sebastian Rowe (EE),
Alejandro Olivo (CPE)

Reviewers:
Bahaa Saleh, Mr. Bill Hawkins (Microchip Technology),
Jim Larson (Comtech Systems), Brian DeFoe (Arrow Electronics)

Table of Contents

Table of Figures and Tables

Chapter 1 Executive Summary

STSC-MS is a four-system project designed to capture the spectrum of stars and classify them by the Morgan Keenan system, which allows us to figure out where on the main sequence the star being observed sits. We can get information on relative mass temperature and elemental makeup from this. Along with the information on the star we are observing we can also determine whether or not the star we are observing is a binary star system. We need to design a spectrometer, tracking system, guiding system, and imaging system to accomplish these feats. The spectrometer used a modified Czerny turner design optimized for scotopic conditions. Careful attention is being paid to the light budget, so we are only using reflective surfaces. Providing us with optimum signal retention while being achromatic. This is important because any refractive optics would modify our spectrum and contaminate our results. The guiding system is designed to resolve stars with short exposure times so that we can ensure tracking accuracy. Every half second we need to take a photo with the guide camera and ensure the target star stays centered. We are using a large aperture lens and a sensitive CMOS Raspberry Pi camera. To ensure perfect focus, both the guide camera and telescope were equipped with an autofocus system. Lastly, our tracking system, after calibration, created a map of the night sky and gave us go-to functionality so that we can easily find whatever object we wish and take its spectrum. We designed our own tracking system because it is crucially important to track accurately. If we could not keep the night sky perfectly still, any vibrations would have caused the spectrum to muddy since it is full of fine details. The software for this system was written and developed by our group and accessible through an on-device display. To ensure that our system can be used in all environments, we used an advanced image processing technique known as image stacking. This approach increased our dynamic range and signal-to-noise ratio at the cost of time. With this system, we formed an all-in-one system capable of providing amateur astronomers with research tools that were previously unaffordable. Even though our system has a lot of moving parts, everything was done piecemeal, and any amateur can pick and choose which piece is needed or adapt the system to their own needs. With this in mind, the design of the systems is simplified, and 3D printable parts are designed without tight tolerances with as few overhangs as possible, making the product as accessible as possible without requiring a high quality 3D printer. Given this product is marketed as a DIY system it is modifiable and can be adjusted to support the needs of the consumer.

Chapter 2 Project Description

2.1 Background & Motivation

Spectroscopy began with Sir Isaac Newton's observations of the sun's spectrum. He made the first measurements by allowing sunlight to pass through a small hole around one-third of an inch broad from the shutter of his window in an otherwise dark room. Then Newton placed a prism in front of the small hole, allowing sunlight to pass through the prism, revealing a spectrum of colors akin to a rainbow. In his 1705 publication "Opticks or, a Treatise of the reflexions, refractions, inflexions and colors of light" he wrote "The blew was refracted more by the lens than the red, so as to converge sooner by an inch and a half, and therefore is more refrangible". Newton understood his discovery was a significant advancement in the study of light. But still, I wonder if he understood the full grandiosity of his discovery that the color of light is related to the angle of refraction. Besides this crucial conclusion, Newton noted anomalies in the solar spectrum: certain colors were missing. Unbeknownst to him, he had just created the first spectrometer, a tool in optics and photonics that would go on to revolutionize our understanding of the cosmos.

Only a short time after Newton's discoveries, Joseph von Fraunhofer, inspired by the potential of these discoveries, sought new applications and is credited as the first to combine a prism with a telescope. This simple combination of instruments enabled him to study the spectra of various celestial bodies. Fascinated by his findings, Fraunhofer experimented further and created one of the first diffraction gratings, a simple grating constructed from metal wire. With this new invention, Fraunhofer measured the wavelengths of the spectral lines he observed.

Since then, the analysis of stellar spectra has become a cornerstone in astronomical research. In today's modern world, spectrometry empowers us to search for life on distant exoplanets using cutting-edge instruments like the James Webb Space Telescope. However, despite the many advancements in spectroscopy and the creation of new manufacturing techniques such as photolithography and precision machining, spectroscopy remains a costly endeavor, often exceeding five thousand dollars, placing it out of reach for many aspiring astronomers.

Tracking systems for astro observation have become a cornerstone for creating clear and accurate images. Manual tracking mounts which can be dated back to the earliest of astronomers have been around for many years however as humans have gotten smarter so has this technology. Jim G. Astheimer as a pioneer guide was added to commercial mounts back in the mid 1980’s and was implemented via feedback loops. This technology rapidly gained traction due to its ease of use and exceptional results. With developments of more powerful microcontrollers and more open source cameras guiding systems have begun to play a crucial role in tracking mounts.

The autoguiding systems found today often allow for near perfect tracking of astral bodies and therefore allow for considerably higher resolution images due to the longer exposure time. The main issue however lies in the cost and portability. With the cheapest viable guiding mounts ranging at around a thousand plus dollars this places it outside the range of amateur photographers. The other issue we bridge is the need for external equipment. Astro photography needs to remain as compact as possible due to the need to relocate or even transport this equipment to different sites. Due to the none all in one approach of these guiding systems at a low cost most of them require the use of an external computer to tap into the guiding system.

In our senior design project, we bridged this gap. We developed a spectrometer that is low-cost and adaptable to any amateur astronomer's telescope. To provide the lowest cost possible, we open-sourced the project by providing 3D printing schematics and software while using affordable precision optics so that our project may be replicable without buying a completed kit. With this approach, any willing astronomer can adapt the project to their needs. This should significantly reduce the cost of the spectrometer since the construction is handled on the consumer's end. This open-source initiative seeks to enable budding astronomers to explore the night sky and learn to interpret the spectra they observe while optimizing performance and price. It also fosters an interest in photonics and astronomy while gaining education on the analysis of spectrograms and the construction of a spectrometer.

2.2 Goals

The most basic goal of the system is to capture a spectrum and create a tracking mount which is able to stay locked onto a star. This goal shows the basic functionality of the system. This goal may seem too basic but the rest of the project scope would follow from this step. It is the most basic of basic goals but after this point in the project everything would follow given enough time we are attempting to do this project over spring and summer which is a short period of time to design and produce a system of this scope.

Project Star Spec should be able to classify stars by letter O, B, A, F, G, K, and M. These classes are well defined by spectroscopic analysis in the optical regime from about 450 nanometers to 700 nanometers. This commonly accepted method for classifying stars reveals information about elemental makeup, temperature, and a rough mass estimate. Having the capability to classify stars is the primary and most fundamental goal of Project Star Spec. Additionally the mount should be able to alter the right-ascension and declination in order to keep the star in frame and allow for larger exposure times for better image and spectral capturing.

Our advanced goal is to identify binary stars by spectroscopy. Very few binary star systems have orbits with eccentricity great enough that the stars are resolvable by observation independently. However, it is well-studied that the spectrum of these binary star systems shows a clear combination of spectra indicating the presence of a binary system. A notable binary star in the sky currently is Procyon. It is known as a spectroscopic binary star system because the two stars, Procyon A and B, are easily distinguishable due to their different classifications. While Procyon B is a white dwarf star, Procyon A is an F-type star. This difference makes it easy to determine that we are looking at a combination of spectrums. The main challenge of analyzing the spectrum for a spectroscopic binary star system is that our system must have enough resolution to differentiate the absorption lines of each star.[38]

Our advanced goal on the mount side is to allow tracking and guiding using image processing techniques to give more precise directions to the mount. Tracking and guiding is a stable in higher end astrophotography mounts which serves the purpose of reducing star trailing and allowing for longer exposure times to be possible with as little loss of single as can be achieved. The mount should be able to keep a star centered in frame accounting for any sag or motor error over time.

Despite our current design utilizing a wall outlet as our power source, and by extension power not being a significant restraint, we made our design as power efficient as possible for several reasons. The first being that we desire to practice principles of good design, and efficiency is a strong component of good design. Secondly, and more practically important, if we achieve a low enough level of power consumption we could advance the design to a battery based power source. This would allow for a more mobile design, enabling end users to use this product in places where power outlets are not readily available or where extension cords are impractical, not to mention making the entire design more compact due to the removal of the wall outlet cord.

The main motivation for choosing a tracking and guiding system which is at more entry level price is that after market analysis it is evident that guide systems greatly improve the ability to take clearer images and therefore more accurate spectrographs and allow for smart technology to be implemented due to the live image processing. This combined with the high entry point for systems such as this make this an interesting addition to Project Star Spec. It is critical that the cost remains accessible and low, without compromising on functionality. Acquiring images of distant stars is a delicate task and we must be able to keep the stability and accuracy requirements of this task intact, while reducing the cost.

Finally, our stretch goal is observing emission lines from bright nebulae such as the Orion Nebula. The main challenge of observing the nebula's emission spectra is signal-to-noise. Nebulas are very dim compared to stars, but unlike stars with a

primarily continuous spectrum, nebulas have individual emission lines. Astrophotographers have long since used narrow band optical filters to block as much surrounding light as possible. Should project star spec be able to resolve the emission lines of a bright nebula, it could be used to determine the correct narrow band filter for best viewing as well as provide descriptive information on the nebula's composition.

The mount's stretch goal is implementing a go-to system which would serve as a means of selecting any given star to lock onto without the need to manually locate it in the sky. This is a feature the high-end telescope mounts provide and serves as the ultimate tool in user experience. This allowed us to simplify the process of gathering spectrums, imaging, and analyzing astral bodies for the user to have the best possible quality of life features.

Objectives

Spectroscopic resolution is a critical factor and is typically classified into three distinct categories: low, intermediate, and high resolution. Low resolution is characterized by $R < 1000$, intermediate resolution falls within the range of $1000 < R < 10000$, while high resolution is defined as $R > 10000$. We aim to attain a resolution greater than 1000 ($R > 1000$). This resolution positions us right at the low end of the intermediate resolution spectrum. Intermediate resolution strikes an optimal balance, providing sufficiently detailed spectral data for effective star classification. By targeting this level of resolution, Project Star Spec ensures the acquisition of high-quality, usable stellar spectra without the prohibitive expense associated with high-resolution spectroscopy. To verify that we have reached our resolution objective we should be able to distinguish a difference of 1nm between wavelengths across the entirety of the visible spectrum.

The guiding optics should be able to focus to infinity and have minimal chromatic aberration so that the guiding system can accurately focus on a singular star. It must also have enough light gathering power to resolve stars with short half second exposures. With these objectives the guiding optics should be appropriately powerful. A good test to verify this objective is to resolve craters on the moon without observing chromatic aberration.

Tracking and guiding are determined by their accuracy in arcminutes as our goal is to produce high quality deep-sky spectrographs. We need our guiding system to operate at or under 3 arcminutes of accuracy. This is the main factor for the guiding system and we were able to reach this by implementing both an equatorial tracking approach which allows us to follow the earth's rotation and after calibration and then adjusting for any error created using a guiding system onboard the mount. With this approach we hope to be able to reach higher exposure times which would meet standards for tracking and guiding mounts.

To minimize noise and star trailing across longer exposure images, which provides better images and spectrographs of deeper space objects, we focused

on smoothness of motion and precision of motor control. The majority of this was determined in software as accurate instructions must be transmitted to the microcontroller. This could likely be done with a Raspberry Pi, or with a mid range MCU, but for our design we were interested in using an FPGA. We initially thought of using an FPGA for this control, alongside the Raspberry Pi, for two main reasons.

- First, FPGA's are wicked and Sebastian loves to use them.
- Second, and more importantly, we believe that the parallelization and low level control available to FPGA's allowed for the processing and response to the tracking data to be more timely and accurate than if this task were to be offloaded onto the Raspberry Pi.

After further consideration, we ended up deciding against using an FPGA, after further investigation into what the actual requirements were with our system, specifically the frequency of movement, we determined that while an FPGA would work, and would be cool, it is simply not necessary and offers less amenities and ready interface between the Raspberry Pi and the control mechanism. For this a mid-range MCU should be completely sufficient to provide the motors with the necessary control for our specific needs.

2.3 Engineering Specifications

Table of engineering requirement specifications

Component	Specification	Parameter
Diffraction Grating	Grating period	600 lines/mm
	Grating type	Holographic
Slit	Slit width	3mm
Concave Mirrors	Spectral range	450-700 nm
	Focal length	100mm
Spectral range	Spectral range	450-700 nm
Guide Camera	Pixel size	3.72um
	Frame Refresh Rate	1s
Spectrometer imaging camera	Pixel size	3.72um
	Frame Refresh Rate	1min>
Tracking mount	Position Accuracy	< 3 arcminute
	Weight	< 5lbs
	Motor Input Power	12VDC
	PCB Input Power	5VDC
	Plate-Solving Time	< 3 minutes
Raspberry Pi	Clock Speed	~3 GHz

Table 1

1.4 Existing Products

The following is a selection of a few existing products in the field that are used in the field of space photography. Though these are each different in nature and have different capabilities, these are all considered “entry level” space photography tools by those within the community.

2.4.1 HEQ5 PRO
Sky-Watcher HEQ5 Pro GoTo Equatorial Mount

HEQ5 PRO is a guiding equatorial mount and is an entry level autoguiding and tracking mount for astrophotography. This mount follows the principle that by moving as the earth rotates and then using software to make small adjustments you can increase the exposure time of the image allowing you to see deeper into the astros. This mount's selling point is its accuracy which is claimed to be at 1 arc minute and its ability to detect 42900+ objects via its smart guide system which is connected via a rs-232 to be able to take advantage of the autoguiding system.

The main areas we are looking at improving are reducing the price from its 1,335$ dollar price to something more entry level and accessible. The goal would also be to eliminate the need for an additional computer instead of offloading the guiding system to an on-board system with a bluetooth interphase. We hope to be able to keep similar arc minute precision while reducing the price and reducing the required hardware.

2.4.2 Star Analyzer 100/200
The Star Analyser SA-100, SA-200

The star analyzer 100/200 is a transmissive diffraction grating filter and is the simplest way to create a spectrogram. However, because it is a transmissive grating, you have a lot of signal loss. It comes in two varieties: 100 lines/mm and 200 lines/mm. The primary difference between these two varieties is the distance it must be mounted from the detector. Should you be limited by space, the 200 enables you to capture the same spectrum at a shorter distance at the trade-off of luminosity.

The benefit of transmissive gratings is the ease of use and price. This filter can be mounted to any camera and only costs $200. The performance may not be the best, but it is currently the most affordable way to do amateur stellar spectroscopy, assuming you have a camera and tracking mount.

2.4.3 SX-Spectrograph Reflective
SX-SPECTROGRAPH PRO

Despite still being considered an amateur spectrograph, it costs $4660. However, this product uses a concave reflective grating that cuts down on the optical design portion by combining one of the focusing mirrors with the diffraction grating. This approach also preserves signal since the spectrum does not have to transmit through an optic and is instead reflected, avoiding absorption. At 550 lines/mm this spectrometer has plenty of resolution and is

more than sufficient for producing sharp spectrographs. Along with a solid optical design, this product also integrates a guide camera to ensure the accuracy of the tracking to make the best quality spectrum.

2.4.4 Shelyak UVEX

The Shelyak UVEX is a professional spectrograph designed for research instruments and marks the entry level for professional spectrometers costing $6686. This is a reflective spectrometer just like the previous high end spectrometers. This spectrometer takes input through a multimode fiber and as a result requires the utmost tracking and guiding accuracy. Because fiber optic cables have core sizes on the order of a few microns the tracking accuracy must be around sub arc second. Along with the fiber design it features an adjustable diffraction grating wheel ranging from 150 lines/mm to 1800 lines/mm this enables the customer to decide what wavelength ranges are the most important to consider for high resolution. This product has much better specifications then we were able to manage but for thorough market analysis it is good practice to look at each price bracket to determine what area of the market we have to compete with.

Table of Technology Comparison Versus Our Product

Name	Grating	Aberration Correction	Camera Included	Imaging pass through	Price in USD
SX-Spectrograph	Concave reflective 550 lines/mm	Minimized with concave grating	Guide camera included in enclosure	No	$4660
Shelyak UVEX	Variable reflective 150/1800 lines/mm	Cylindrical lenses for astigmatism correction	No, Fiber based	No	$6612
Star analyzer	Transmissive 100/200 lines/mm	None and transmissive	No	No	$188
Our Project	Reflective 600 lines/mm	None	No	Yes	~$600

Table 2

2.5 Block Diagram

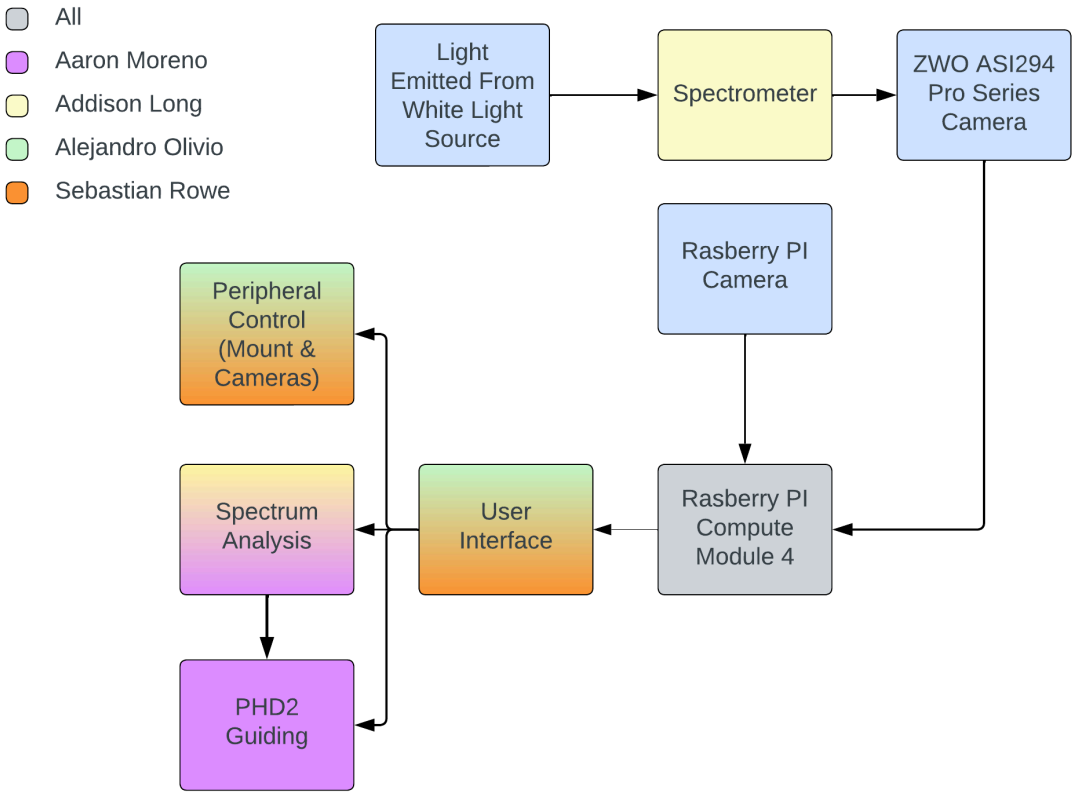


Figure 1

2.6 Operational Flow Diagram

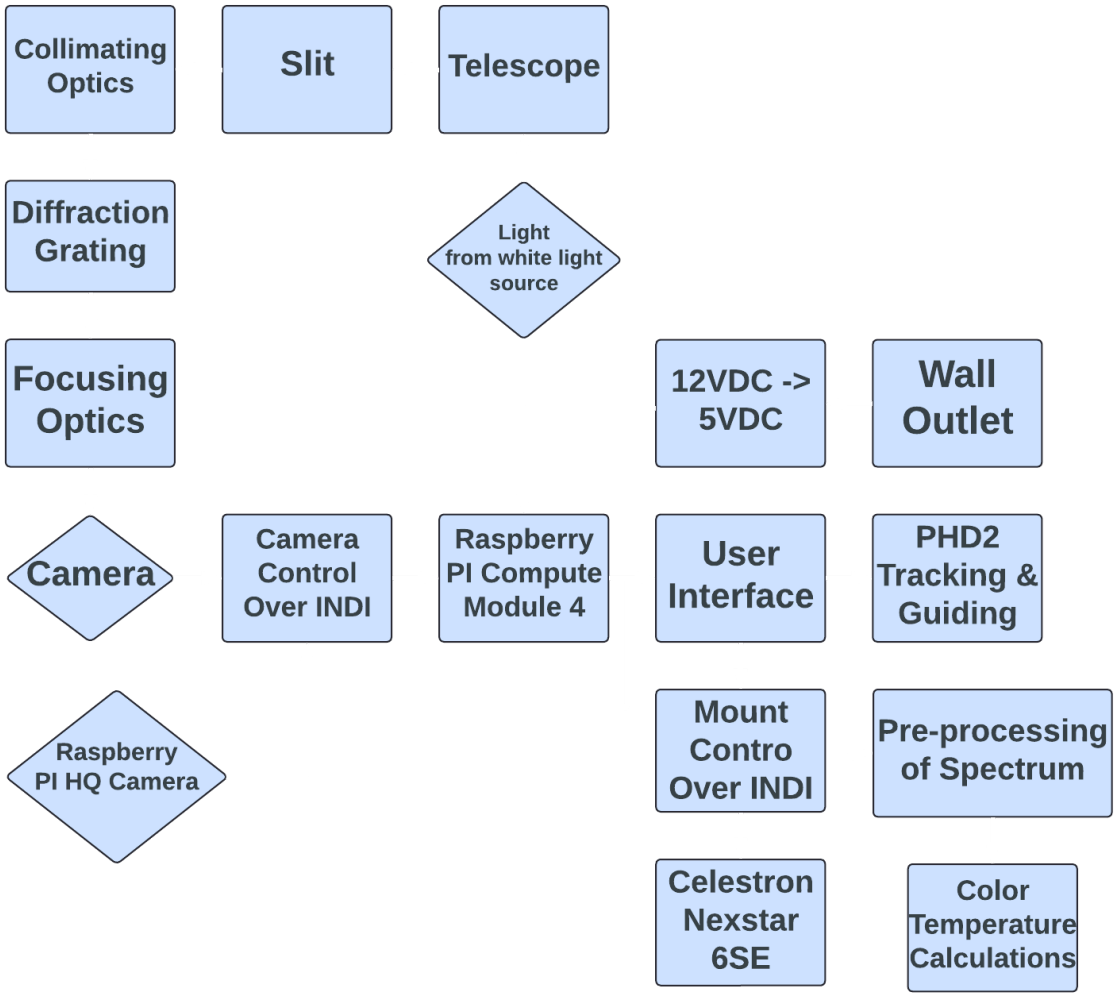


Figure 2

2.7 Image Processing Block Diagram

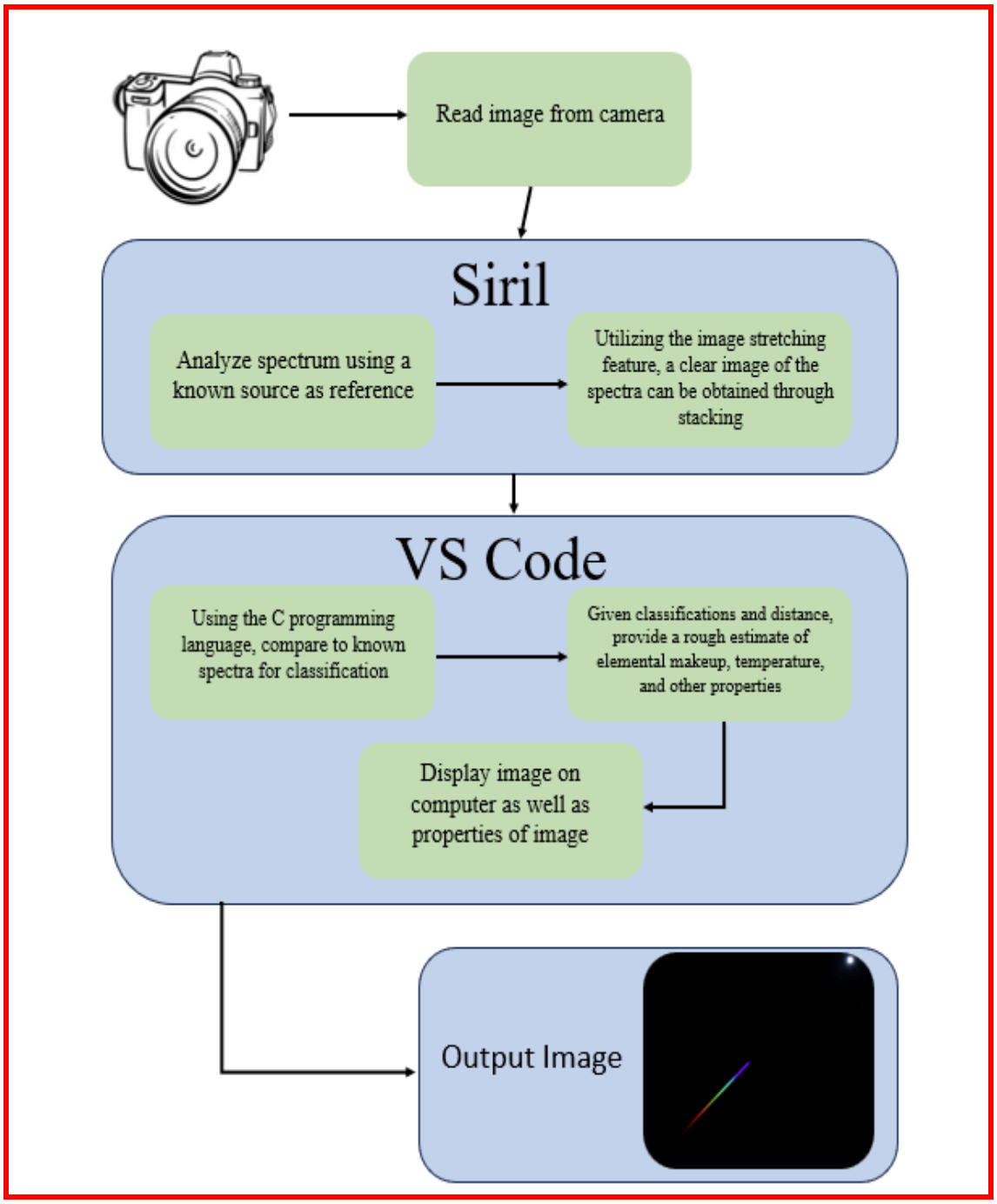


Figure 3

2.8 House Of Quality

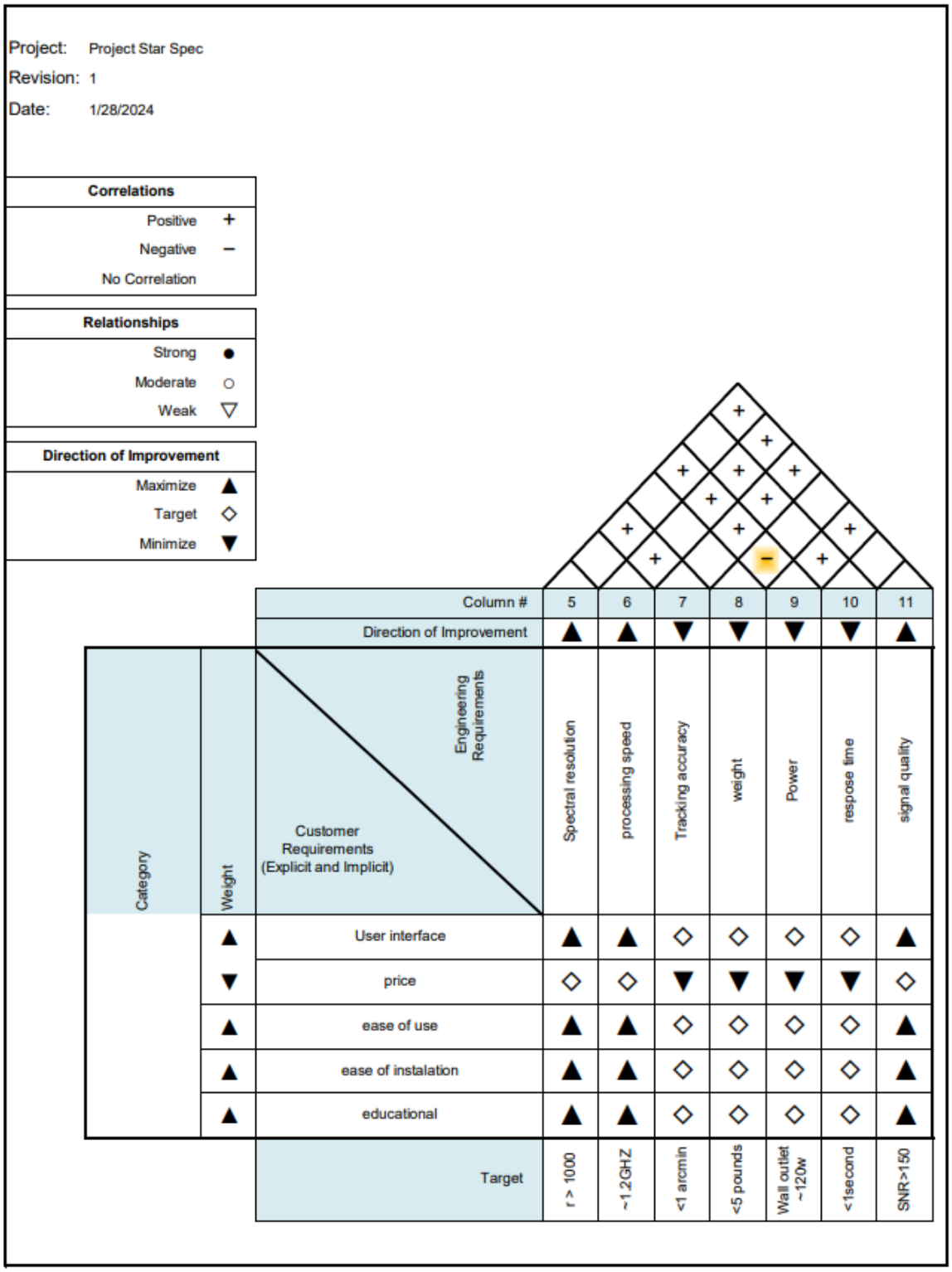


Figure 4

Chapter 3 Research and Investigation

Rough Diagram of Full Optical Design (not to scale)

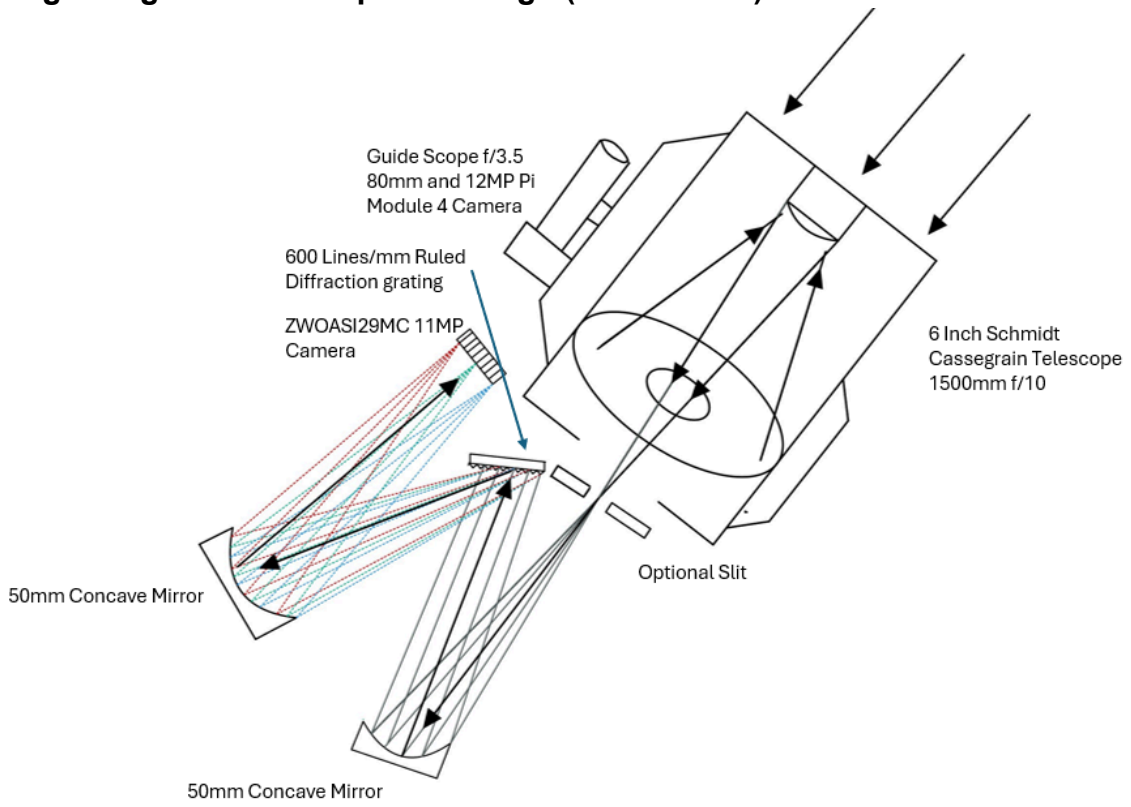


Figure 5

When devising the optical design to fulfill our requirements, we must consider these three core tenets: cost-efficiency, performance, and adaptability. Most importantly, we need to minimize expenses as much as possible to make stellar spectroscopy accessible to the amateur astronomer. The best way to accomplish this is by limiting the number of optical components used in the system.

Next, we need to ensure that we avoid refractive optical elements. The nature of our application—spectral analysis of starlight—demands signal integrity throughout our system. Refractive elements, by their very nature, introduce signal attenuation. This constraint is especially pertinent given the inherently weak signals involved in astronomical observations, where every photon counts.[37]

Moreover, the system must afford us the flexibility to adjust the spectral resolution according to specific observational needs. This adaptability is crucial for tailoring our apparatus to various scientific inquiries.

Given these considerations, the Czerny-Turner spectrograph is a great model for our optical design. This configuration starts with light entering through a slit. The light is then collimated and directed onto a diffraction grating that disperses the light incident onto a concave mirror, which subsequently focuses the dispersed light into our camera. This arrangement satisfies our criteria for minimal optical components and non-reliance on refractive elements while providing flexibility in tuning spectral resolution. Leveraging the Czerny-Turner design facilitates high-quality spectral analysis while adhering to our project's stringent cost and performance specifications.[36]

When determining the spectral resolution of our system we used the commonly accepted metric of spectral resolution. As seen in the following equation.

$$R = \frac{\lambda}{\Delta\lambda}$$

Where λ is our central wavelength and Δλ is equal to:

$$\Delta\lambda = \frac{RF(\delta\lambda)Ws}{n(Wp)}$$

RF stands for the resolution factor, a dimensionless number that quantifies the system's inherent resolution capability and is dependent on the slit width and pixel size of our system. We approximate to around RF = 1.5. **Ws** is the width of the slit, **n** refers to the number of illuminated pixels on the detector that the dispersed light covers, and Wp is the pixel size. The easiest quantities to tune for spectral resolution were the slit width and number of illuminated pixels[27].

3.1 Optical Design Part Selection

In the optical design of the Czerny-Turner Spectrograph, the selection of components is paramount to preserving signal quality while balancing cost and adaptability. A critical component in this regard is the diffraction grating, which significantly influences the system's overall performance. After careful consideration, we have opted for a holographic diffraction grating with 600 lines per millimeter, prioritizing a reflective over a transmissive approach for several reasons.

Despite their higher cost, reflective diffraction gratings offer superior signal quality compared to their transmissive counterparts. Transmissive gratings attenuate the signal and are rarely optically flat, leading to potential aberrations and a compromised output spectrum. The signal's integrity is crucial for our system, making the reflective option the clear choice despite its cost.

The line count per millimeter directly affects the spectral resolution, with a higher count dispersing light over a larger area, thus enhancing resolution. We balance cost and performance by selecting a 600 lines/mm grating, achieving ample

separation between the spectral range's low end (450 nm) and high end (700 nm). This choice provides a desirable angular dispersion while sitting at the low end of line counts for holographic gratings, saving money.

While ruled diffraction gratings offer efficiency and an advantageous blaze angle, their production costs are significantly higher due to complex manufacturing requirements. Conversely, holographic gratings present a cost-effective alternative, sacrificing only a marginal amount of signal efficiency (approximately 10%). This trade-off is acceptable for our application, allowing us to maintain budget constraints without severely impacting performance.[42]

3.2 Diffraction Grating Comparison

Size	Type	Price
12.7 mm x 12.7 mm	Ruled	$105
30 mm Diameter	Concave	$585
20 mm x 20 mm	Holographic	$26

Table 3

Given the decision to use reflective optics, selecting mirrors is the next crucial step, with considerations including spectral range, surface curvature, and numerical aperture.

Our mirrors must support the visible spectral range (450-700 nm), necessitating a dielectric coating that maximizes reflectivity within this range.

The trade-off between spherical aberration and alignment complexity influences the choice between parabolic and spherical mirrors. Parabolic mirrors, while eliminating spherical aberration, are challenging to align precisely, with minor errors causing significant coma. Therefore, we used spherically ground mirrors with a slow numerical aperture. This choice simplifies alignment and collimation at the expense of a larger assembly size but avoids the significant aberrations associated with misaligned parabolic mirrors.

3.3 Comparison of Mirrors

In our system the mirrors in our spectrometer are incredibly important. We have a very small light budget and any lost signal reduces the performance of our system significantly. The following section is a comparison of mirrors that operate in the visible range and have a focal length appropriate for the system. We are most interested in the coating on the front of the mirrors.

Mirror	Coating	Spectral range	Reflection efficiency
CM127-050-E02	Vis dielectric	450 nm-700 nm	99%
CM127-050-G01	Aluminum	450 nm-20 um	90%
CM127-050-P01	Silver	450 nm-20 um	95%

Table 4

The CM127-050-E02 is the best mirror for our system however it costs $20 more per mirror then the silver coating while also being more fragile. To increase the durability and reduce the cost of the system, we looked at the CM127-050-P01 where the protected silver coating only loses 5% efficiency over the Vis dielectric coating. This mirror is more than sufficient even with the slight reduction in reflection efficiency. After doing an extensive simulation we can use mirrors of a diameter 12.7mm the sensor of our camera is small enough that we can make our spectrometer as small as possible. Parabolic mirrors were an option for this project and would increase the performance of our system by eliminating spherical aberration but the trade off is the difficulty of assembly. Since our product is to be assembled by the consumer, tight tolerances are not allowed. For this reason we used spherical mirrors.[42]

3.4 Comparison of Telescopes

The choice of telescope does not significantly affect the operation of the spectrometer, but there are a few considerations important for best performance. First, the telescope must have a sufficiently long focal length and large f ratio to isolate individual stars. While a large f ratio is generally undesirable in this circumstance, it works to our benefit, enabling us to isolate individual stars. At a high f ratio, many background stars will be resolved, which will cause the spectrum captured to be a combination of our target star and surrounding stars. Just as having a high f ratio is essential, having a long focal length is also important. Too short a focal length, and the star we wish to measure the spectrum of will be crowded by other stars. The following figure shows the difference between two images of the star Alnitak from a lens with an f ratio of f/1.4 and a focal length of 50mm and a lens with an f ratio of f/10 and a focal length of 1500 mm. As you can see, using a higher f ratio and longer focal length in this circumstance is advantageous.

3.5 Telescope Comparison for the star Alnitak

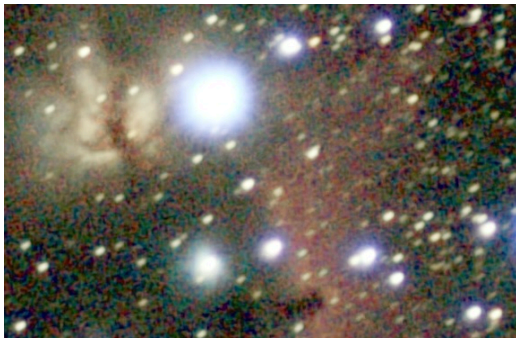



Figure 6 1500mm f/10 50mm f/1.8

In addition to focal length and f-ratio, the type of telescope through which data is collected significantly impacts the clarity of the resulting spectrum. If you chose a refracting telescope, choose at least an extra-low dispersion (ED) glass doublet apochromatic. It is essential to minimize chromatic aberration to avoid significantly distorting the data. The Schmidt-Cassegrain model is preferable for those opting for a reflector telescope to bypass chromatic aberration issues. This design offers minimal primary mirror obstruction, attributed to the secondary mirror's support by the front correcting plate, and eliminates diffraction spikes commonly associated with Newtonian reflectors. Furthermore, the Schmidt-Cassegrain telescope benefits from a folded light path, which shortens the tube's length and considerably reduces the weight of the optical tube assembly.[40]

3.6 Telescope Comparison

Telescope	F/#	Focal Length	Style	Glass Type	Aperture	Price
Celestron 6se	f/10	1500 mm	SCT	Mirror	6 inch	$1299
Askar 120 Apo	f/7	840 mm	Triplet Apo	ED Glass	4.72 inch	$1499
Sky-watcher Quattro	f/4	1000 mm	Newtonian	Mirror	10 inch	$1180

Table 5

Guiding Optics

While a high f ratio and long focal length are essential for the telescope taking the measurements, it is the opposite for the guiding telescope. The guiding telescope should have a shorter focal length and low f ratio. In this circumstance, we are primarily concerned with resolving as many stars as possible with as little exposure time as possible. This way, we can ensure that we guide accurately. Using optics with a wide field of view gives us more information to determine where we are looking in the night sky. The main principle behind guiding is to

focus on one or more stars and lock them onto a single pixel or set of pixels, then guide the tracking mount to keep the stars in the exact location. We want a low f-number to pull in as much light as possible and a low focal length to reach both goals of significant signal and wide fov. The best option for our guide telescope is an 80mm f/3.5 lens. Not only is this lens available to us, but it will give a sufficiently wide field of view and gather enough signal to resolve stars with short exposure times of less than one second, allowing us to update the position frequently and ensure tracking accuracy.

3.7 Camera Comparison

A significant restriction on this project is creating a spectrometer that is accessible to as many people as possible, so when we design the system, we want to ensure adaptability. In ideal circumstances, the best option would be a monochromatic mirrorless camera, but most entry-level astronomers only have access to a DSLR. Two main concerns related to DSLR cameras are vibration and the ir cut filter over the camera sensor. Every SLR camera has a mirror that flips up to act as the shutter for the camera sensor. This motion of the shutter causes a vibration through the camera, and sometimes this vibration is enough to cause the stars to appear stretched, leading to a smeared spectrum, making it more difficult to resolve spectral lines. Fortunately, from my experience taking thousands of images with our camera, the Canon Rebel T7, I only lose around one-third to vibration-related artifacts. For most images, this trailing is only noticeable in the dim background stars, but it would certainly affect the spectrum since we are trying to resolve fine details. The figure below shows two frames, one frame where the vibration impacted the image and another where the vibration caused no issue. As long as we take multiple frames and cull any that show star trailing, we should be able to preserve a mostly intact spectrum.

Distortion caused by mirror in DSLR compared to non distorted image

Figure 7

Camera Comparison

Camera	Pixels	Sensor Diagonal	Color/Mono	Pixel Size	Price
Canon Rebel T7	24.1MP	26.8mm	Color	3.72um	$400
ZWO ASI183MM	20.2MP	15.9mm	Mono	2.4um	$1000
QHY183M	20.5 MP	15.9mm	Mono	2.4um	$1000

Table 6

In the table above, I tabulated our camera options. We decided to pick the Canon Rebel T7 because it provides us with the best sensor at the best price. However, there is still one significant main concern.

The Canon Rebel T7 has an infrared cutoff filter. To ensure that daytime photos do not have a red tint, modern cameras contain a filter that ramps up from around 500 nm to 700 nm, and at around 700 nm, nearly 75% of the signal is lost. I plan to regain this lost signal by using an advanced data processing technique known as photo stacking to combine the signal from multiple short exposures, thereby decreasing the noise floor and increasing the signal. This works because we are working with low-intensity light, so we have to deal with quantum noise inherent in light, which will be proportional to root n, where n is the number of collected photons. As we collect more photons through multiple exposures, the amount of noise decreases by root n. Hopefully, this will retain enough signal to resolve the absorption line up to 700 nm. The Canon Rebel T7 also gives us the advantage of a massive sensor. While it is an APS-C format sensor, it has 24 megapixels, significantly more than most dedicated astro cameras of the same price range. If we overcome the vibrations and infrared

filter, it will provide us with a higher spectral resolution due to the amount of pixels we can illuminate with our sensor.[49]

3.8 Control Architecture Research & Part Selection

Computer Vision plays a key role in our autonomous guiding and tracking system for autonomous spectral analyses. Computer Vision is the idea of interfacing with some kind of camera over some integrated circuit in order to detect and compute information about the image being seen by the camera. The main considerations for Computer Vision image processing speed as well as image detection accuracy and mapping. Due to the limitation presented by standard microcontroller based approaches. Raspberry PI's have entered the scene as a reliable and efficient solution to embedded Computer Vision based systems. Explained Below are the Pro's and Con's of using Raspberry PI based architecture within the Tracking and Guiding System. The Rasberry Pi of choice for the Computer Vision application will be the Raspberry Pi Compute Module 4S.

Raspberry PI Compute Module 4S

Figure 8

Pros:

Parallel Processing Capabilities:

The Raspberry Pi Compute Module 4S is structured around the Broadcom BCM2711 processor which offers 4 cores and 4 threads as stated in [31]. This means that much like a modern computer the Raspberry PI can run multiple applications simultaneously allowing for image processing, vector calculations, and model training and analysis to all happen off one chip set. The project requires many different processes to be run concurrently as the tracking and guiding system work as a central control point to manage and dictate timing on all other processes.

The approach to opt-in for a multiprocessing approach rather than a multiple independent microcontroller based approach stems from the size of the data that

would have to be transferred between stages. By using the Raspberry Pi to handle the multiprocess we can avoid having to transfer large images and other relevant information back and forth in a hardware feedback based communication system which would increase the overall propagation delay of the calculations and could lead to loss or errors due to the amount of data needed to be transferred per image taken. Due to these factors the parallel Processing capabilities provided by the Raspberry Pi work as the perfect central control and data management system for our project.

Linux Environment:

The Raspberry Pi Compute Module 4S runs off the Raspbian operating system as stated in [31] which is a debian based operating system. Access to a linux based operating system such as Raspbian opens up the doors for what is capable. Linux allows for the use of the Software Development Kit (SDK) provided by cannon for their EOS Rebel T7 DSLR camera which is our camera of choice for the spectrogram image. Linux's extensive community and developer kits make a task such as interfacing with a Canon camera simple and efficient; this is something that if done via a microcontroller would take large software and hardware approaches to attempt to interphase with the camera.

Linux also allows programming on any available programming language which will play a key role in the all in one approach which we reach. With the ability to rely on Python we are able to leverage complex libraries in both machine learning and image processing. Scikit-Learn library allows leveraging powerful machine learning models that we can use for classification by leveraging extensive open source databases that have been precompiled. OpenCv's Python wrapper allows us to simplify algorithm design for analyzing and mapping our guiding and tracking image by leveraging the over 2,500 available image processing algorithms.

Linux file structure will also allow for easy storage of large files for temporary processing. The ability to store images easily and manage the systems storage through multiple processes at once will allow for us to offload some of the more memory intensive processes as we only have 1 GB of SDRAM available to the system as that is the limitation of the Raspberry Pi Compute Module 4S.

Built-In Camera Support:

The Raspberry Pi Compute Module 4S comes with a built-in Camera-Serial Interface CSI-2. The CSI-2 has been the primary imaging solution for embedded systems since its release in 2005 its high bandwidth allows for high resolution imaging that protocols such as I2C or UART based cameras lack the functionality for. Being able to leverage a dedicated interphase port for the camera makes creating a high performance tracking system possible. High-Resolution images with proper exposure control and the bandwidth to rapidly take these images and perform calculations on them would be quite next to impossible if solely relying

on microcontroller technology by leveraging the Raspberry Pi Compute Module 4S this now becomes possible.

The Raspberry Pi Computer Module 4s also comes with extensive support, documentation, and libraries that will make development more smooth and allow for faster high resolution images with limited loss. With its built-in micropython code testing and design should be more straightforward as we once again will be able to lean upon the linux environment of the Raspberry PI Compute Module 4S and its ability to easily interphase across powerful peripherals for faster development.

Expandable Storage:
The Raspberry Pi Computer Module 4S also comes equipped with a 200-pin SODIMM edge connector as stated in [31] suitable for expanding the storage capabilities of the Raspberry Pi; this is a key feature needed for our design. The goal of stacking requires ample amounts of available storage to hold high resolution images and then overlay them on top of each other to create a clearer, more accurate image. Stacking alone will require around 10 GB of storage space which becomes difficult to accomplish through a standard microcontroller based approach as it lacks easy and fast extra storage options.

Built-In Graphics Processing Unit (GPU):
Broadcom BCM2711 once again shines through above standard microcontrollers as it comes pact with a built-in GPU which allows for faster input/output on graphical processing based tasks. The Broadcom VideoCore VI which is the graphics processing unit of the BCM2711 features a 0.5 GHz graphics processing unit with 64 shaders and 2 GB of max memory as stated in [31]. This will free up the CPU's bandwidth to allow for imaging and tracking to be handled by this unit while the machine learning model and complex vector calculations done after the image processing to be handled by the CPU. This increases our processing power more than any single microcontroller based architecture could do.

GPU's excel at processing image's compared to CPU's this will be key with reducing the tracking error as speed will play a major point in our overall accuracy. Delays in calculating the vector will lead to reductions in overall image quality and this means that we need to be able to quickly process images and generate complex computations to derive the best directions for the motor to keep our telescope mount centered. GPU's feature higher memory bandwidth which will allow for the images to pass through and be processed faster. The VRAM will also allow us to leverage additional resources aside from the built in RAM of the Broadcom BCM2711 to further speed up the computations needed in the tracking and guiding system.

The linux based architecture will also allow us to leverage Python's GPU image processing libraries which rely on a GPU for image processing optimizations. The

parallelization options for images that become available to us increase drastically by having access to a built in graphics processing unit like the Broadcom VideoCore VI.

Cons:

Size:
The form factor of the Raspberry Pi Compute Module 4S is one of the biggest fall backs. The 67.6mm x 31.0mm is substantially larger than microcontroller's which are normally ranging from the 5x5mm to 20x20mm in size as stated in [31]. Increasing the size of the central processing unit of the design will mean that the overall size of the PCB will increase. This is a primary concern as the limited mounting points of the mount will make fitting a large enclosure for the circuitry a challenge.

Cost:
The Raspberry Pi Compute Module 4S is considerably more expensive than the standard microcontroller. The price restriction we set to create an affordable entry level guiding and tracking system makes it so that the price increase from the Compute Module 4S will make us have to limit ourselves on additional features.

Embedded System Comparison Table:

Name	Processor Speed	Cores	GPU	Price
Raspberry Pi Compute Module 4	1.5 GHZ	4	YES	30.00
Raspberry Pi Compute Module 3	1.2 GHZ	4	YES	39.99
ESP32-D0WD-V3	240 MHZ	2	N/A	1.50
TMS570LC4357-SEP	300 MHZ	2	N/A	24.99

Table 7

3.9 Embedded Tracking Camera Research & Part Selection

The camera used for the tracking and guiding system is a primary focal point of the research and part selection. This will be the peripheral on which we rely the

most on to meet the required accuracy specifications; without clear and low-loss images we run into a degradation of our positional accuracy due to image detection outliers affecting the calculations caused by the distortion of the base image. The camera of choice is the Raspberry Pi High Quality Camera. This is a 12.3 Megapixel camera with all the base requirements and configurability that were needed in order to adapt it to a tracking and guiding system. The camera was also made with a stacked imaging architecture which means the circuitry lies right below the pixel layer allowing for faster image processing and noise reduction due to the distance traveled by the signal as stated in [33]. These shortened pathways will play a major role in squeezing out as much speed as we can for our image processing.

Raspberry PI High Quality Camera

Figure 9

Features

Output Types:

The High Quality Camera from Raspberry PI allows us to leverage their RAW 12 Uncompressed imaging format which will allow us to create clearer images with little to no loss as stated in [33]. This is done by just sacrificing some extra memory which by leveraging the GPU's bandwidth should allow us to process high definition images quickly and efficiently. The RAW format allows for a larger access to the metadata of the image directly from the camera before any software correction allowing for further post processing in creating the most clear and lossless image. The main selling point of RAW 12 imaging format is its lack of artifacting due to compression techniques often found in the JPEG image format.

Sensor Type:
The High Quality Camera from Raspberry PI comes fitted with the Sony IMX477R camera sensor as stated in [33]. This is a stacked,back-illuminated sensor which is known for high quality images within a cheap and small form factor. The back-illuminated design is ideal for low light environments like tracking a star. The sensor allows for more light to hit the photodiodes making it capture brighter images in dimmer environments. This will play a major role in the tracking and guiding system as image quality will determine how close we can get to generating the perfect position and will lead us to increasing our arc seconds of accuracy.

Customizable Lens:
Lenses selection served as the deciding factor when choosing the Raspberry Pi High Quality Camera. Our application required a very narrow depth of field in order to visualize astral bodies which lie so far away and allow for clear detection of them. The High Quality Camera allows for both CS and C mount lens selection which will allow us to retrofit a 50mm lens allowing for us to narrow down the FOV to (Insert FOV).

Embedded Camera Comparison Table:

Name	Sensor Resolution (Pixels)	Pixel Size	Focus Type	Maximum Exposure Time (Seconds)	Lens Type
HQ Camera	2592 x 1944	1.55µm x 1.55µm	Adjustable	670.74	Adjustable
Camera Module 3	4608 x 2592	1.4µm x 1.4µm	Motorized	112	Fixed
GS Camera	1456 x 1088	3.45µm x 3.45µm	Adjustable	15.5	Adjustable

Table 8

3.10 Tracking and Guiding Research

Tracking and guiding design will face a couple issues which we have overcome. These images must be processed in real time in order to limit the loss of tracking accuracy. This comes as a challenge as it will require quick and efficient algorithms in order to properly translate the positions to the driving motors on both the x and y axis. The process can be broken down into 3 different steps. First comes making the images share a uniform coordinate system that factors in

for shifts in scaling and slight variation in focusing. This process is called Image alignment. Next, we have feature detection and registration in order to properly align the images and make clear how much our stellar object has moved relative to the earth. We need to detect and classify key points within the image that will work as anchor points. Finally, once we have defined the key features within an image we can use optical flow techniques in order to properly calculate the vector between the two images therefore providing where our telescope should move to.

Image Alignment:

The key process of preparing two different images for processing and comparing is called image alignment which is a subcategory of image registration. This process adjusts rotation, translation, and scale in order to make one image align with one another. Image alignment in our case will primarily work as a way of improving our object recognition by ensuring that features are matched correctly between the different frames. The chosen approach for solving this issue will be using Python's OpenCV library. The OpenCV library will allow us to translate, rotate and scale our image in order to create a homography transformation. A homography transformation is a perspective transformation used in computer vision. This perspective transformation works by shifting different matrix values of key features so that they can align across different frames [26].

OpenCV employs the ORB (Oriented FAST and Rotated BRIEF) alignment algorithm as stated in [26]. This algorithm is a feature detection and description algorithm which will aid in detecting the key features which to base the homography transformations on. ORB consists of two major processes with some optimizations in order to overcome its shortcomings. The first is the FAST keypoint detection algorithm; its purpose is to find keypoints within the image. The keypoints are then run through Harris corner measure in order to find the top N points among them. The Harris corner measure states simply that corners can be found anywhere there are drastic changes in features by moving in any direction. This is used to generate the edges, corners, and flat regions within the image; these will work to identify key features within the image and serve as reference points. ORB further improves the Harris corner measure by accounting for rotation which the original principle does not do.

Visual Representation of Harris Detector Algorithm [48]

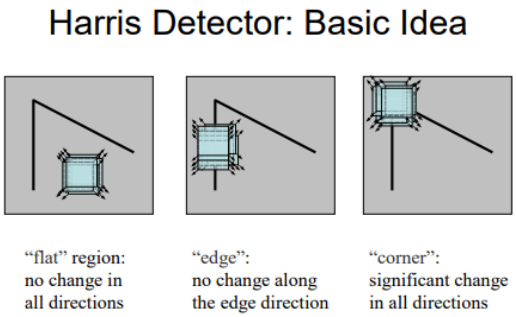


Figure 10

Feature Selection and Detection:

Feature detection is a complex problem which will leverage Python in order to solve. Creating bounding boxes around our star of choice is our primary goal and in order to do so and accurately be able to feed this information into optical flow techniques we must be able to quickly and accurately create bounding boxes around the important stars within our frame. This issue will be resolved with the aid of the Photoutils Python library which is built with astral photography in mind and brings an extensive set of tools to locate stars within an image. Photoutils is an open source Python library which means it will work perfectly within the design constraints. Photoutils contains the DOAStarFinder class which works based off the DOAFIND algorithm as stated by [26]; this will be our primary way of creating bounding boxes around our star of choice. We define our threshold intensity values and the DOAFIND algorithm will then create a bounding box around any object whose intensity values are above the predescribed amplitude threshold and follow the basic shape of a 2D gaussian kernel. The DOAStarFinder class will also give us an estimate of the object's roundness and sharpness which could prove useful for our optical flow techniques.

The photoutils library allowing us to create a bounding box around the stars will work to determine our feature selection to use for both references during the Optical Flow calculations and will define how large the remaining area will be when fed into the Optical Flow portion of the problem. The main considerations here once again rest upon computation speeds as all image processing the smaller the overall image the faster computations can be derived. This means we leverage the live feed of the camera in order to center ourselves on our star initially and then narrow down the area of inspection of the algorithm.

Optical Flow:

Optical flow is a computer vision technique used in order to ascertain the motion of an object between two consecutive frames or images relative to the user's point of view [26]. The main purpose of this technique is to allow us to accurately track an object across different images taken moments apart although it is often

used in video applications we use it in reference to consecutive images. The main principles that will allow this technique to work are the idea that the brightness of an object is the same across different frames. We use the brightness of an object almost as a unique identifier for each star across different frames. The next important concept which Optical flow techniques and algorithms work off is the principle that consecutive images will only have small changes in them.

The first limitation of Optical flow techniques is that it is computationally expensive. Due to the comparison of two already large files we used an optimization approach in order to not add unneeded strain to our Raspberry Pi. The recommended approach is to reduce the size of the overall image due to the fact that we only should have about 15.04 arcseconds per second of change in the image it is safe to assume we can reduce our overall image size to only include a smaller frame centered around the center of our camera which is where our stars can have their initial alignment determined. This will work to reduce the computational strain of optical flow techniques by reducing the amount of objects within the frame. The next limitation which must be acknowledged and addressed is the change in lighting due to the stable environment of the stars; we did not see significant changes in lighting across frames unless a phenomenon were to occur. These phenomena were easy to resolve as we were able to perform checks to eliminate dark and bad pixels in images.

The OpenCV library will once again be our main way of implementing optical flow algorithms. Its implementation is based upon the Lucas-Kande algorithm for optical flow. This is the best choice given our hardware as Lucas-Kande deals with small changes between frames and has a focus on further optimizing performance which in a real time system will be crucial for reducing our arcsecond of accuracy error. The Lucas-Kanade Algorithm works of the previously discussed assumptions that optical flow must follow as stated in 26. It then computes a taylor series expansion that linearizes the change in intensity and incorporates a temporal derivative which specifies the change in intensity overtime. The following is the simplified result of that equation. The Ix represents the change in intensity in the X-axis, and the Iy represents the change of intensity in the Y-axis.

$$I_x u + I_y v + I_t = 0$$

The v vector is the result which is the motion vector and is derived from taking b which is the constant value of It and a matrix A which is the gradients of Ix and Iy for each pixel in our desired frame. The final result is a minimized vector which contains the optimal motion vector between our point and every point within the neighborhood.

$$\mathbf{v} = (A^T A)^{-1} A^T \mathbf{b}$$

3.11 PHD2 Research

PHD2 is a leading open source guiding and tracking utility application. This application works as a means of integrating many different guiding and tracking algorithms with the aim of building your own tracking and guiding system. This works as a bridge and test reference that we can use against our own system design or even as an alternative if integration is deemed possible. The main advantage of integrating a system such as PHD2 is that it goes beyond typical tracking and guiding, providing predictive capabilities on top of its base library of tracking and guiding algorithms.

The PHD2 software features a diverse set of algorithms of interest to us and gives a good indication into possible algorithms we can implement. Amongst this list are the following algorithms which are the most interesting to implement for us.

Hysteresis algorithm is a historical based guiding algorithm that uses past data as a way of computing future data with the aim of reducing over correction. This is done by altering two main weights one is the historical weight and the next is the aggressiveness weight. The historical weight which is accepted as a percentage is the main indicator of how much the telescope should move based on its past data. This serves as a way of smoothing out movement and allows for cleaner transitions that attempt to move linearly towards the desired target rather than snap to a point. The historical weight parameter will play a major role in this algorithm's implementation as smoothness of movement plays a major role in reducing star trails and additionally will aid in reducing the vibration of the motors from sudden jumps which would alter the image. The next major parameter which is fed into the Hysteresis algorithm is the aggressiveness parameter this parameter allows us to define what is considered significant movement as overcorrection can lead to unnecessary movements in the system which can both reduce image quality and allow us to move the motors in smoother linear motions rather than rapid jagged motion.

Resist switch is an algorithm of interest that focuses on the declination of the guiding system. Algorithms tend to focus mostly on the right declination of a system however the Resist switch algorithm works to primarily correct the declination of the system. The declination of the system is crucial for reaching the 1 arcsecond accuracy system requirement as it will work to offset natural vibrations and mount sag against the gears. This would be a key algorithm to implement as the means of increasing our accuracy and will be a major point of interest. The algorithm works much like the Hysteresis algorithm; this means it is a history based approach that factors in previous directions when assessing the next direction. The next major implementation of the algorithm is its ability to

change direction of a single frame which is a major point as it helps deal with outside variables such as cable snag, gear misalignment, and high wind conditions that can create sag in the systems declination over a single frame.
The Z-filter algorithm is also a potential solution to our basic goal of just tracking and keeping us within the desired 1 arcsecond accuracy range. This is due to its focus on analyzing low frequency components in the aim to fix the motor's periodic error. This algorithm also allows us to compute tracking at lower exposure times which should decrease our overall error as the ability to calculate the correct position more frequently should substantially reduce gaps in tracking error. There are two main factors when working with the Z-filter algorithm the first is the Exposure Factor which serves as a means of simulating larger exposures within the algorithm allowing us to estimate what we are seeing and predict what to do. The next is the Minimum Movement parameter which denotes what the minimum amount of change should be before we direct the motors to adjust in our case this parameter would be set to half an arcsecond of change.

The last algorithm of interest is the most computational heavy approach but provided the smoothest directions for tracking and guiding. The PPEC algorithm is a machine learning approach which allows us to predict the accuracy of tracking in real time. The ability to proactively generate guide instructions to reduce error reduces propagations delays often associated with other algorithms. There is an increase to the initial setup time as the model must first learn the inherent error established in the mount and then generate weights which can predict the movement error. This algorithm works at reducing all 3 types of error typically associated with tracking and guiding systems. These 3 error types can be described as Short-term for high-frequency errors such as those caused by gear roughness or seeing. Medium-term which are residual periodic typically occurring at intervals less than or equal to the worm period. The last kind of error is Longer-term for steady drift and for lower frequency (longer time interval) harmonics that can be caused by the interactions of multiple gears in the drive train. This will be the main algorithm to detect star movement that we chose to implement as it will be the most likely to reduce all the required drift generated by the system.

3.12 Motors

3.12.1 Motor Type Overview

In order to track and capture the celestial bodies as we wanted in the image, we needed to expose the sensor with exceedingly accurate positioning in order to stitch our data together into a single cohesive image. An image that is not blurred or distorted by inaccurate positioning. In order to get accurate positioning, we have to move the camera and in order to move the camera we most certainly will need motors.

Before looking at specific models of motors, we first decided to narrow down what type of motor we would end up wanting to use. Here the different types of motors are compared and contrasted. (Hint: We go with the stepper motor.)[22]

The following indent was written with the assistance of ChatGPT.

DC Motors:

Pros:

1. Simplicity: DC motors have a straightforward design, making them easy to understand and maintain.
2. Variable Speed Control: They offer precise control over speed and direction, making them suitable for applications requiring flexibility.
3. High Starting Torque: DC motors provide high torque at low speeds, making them ideal for applications requiring rapid acceleration.

Cons:

1. Brush Wear: Traditional DC motors use brushes and commutators, leading to wear and occasional maintenance requirements.
2. Limited Lifespan: The brushes in DC motors can wear out over time, necessitating replacement and potentially affecting reliability.
3. Sparking: Brushed DC motors can produce sparks due to brush-commutator contact, posing a safety hazard in certain environments.

Asynchronous/*Induction Motors (AC):*

Pros:

1. Robustness: Induction motors have a robust design, making them suitable for harsh industrial environments.
2. Low Cost: They are generally more cost-effective than other motor types, making them a preferred choice for many applications.
3. High Reliability: With fewer moving parts, induction motors offer high reliability and require minimal maintenance.

Cons:

1. Limited Speed Control: Induction motors typically operate at a fixed speed determined by the frequency of the AC power supply, limiting their suitability for variable-speed applications.
2. Lower Efficiency at Low Speeds: Efficiency can decrease at low speeds, impacting performance in applications requiring precise control.

3. Lower Power Density: Compared to some other motor types, induction motors may have lower power density, limiting their use in compact applications.

Synchronous Motors:

Pros:

1. Precise Speed Control: Synchronous motors offer precise speed control and operate at a constant speed synchronized with the frequency of the AC power supply.
2. High Efficiency: They can achieve high levels of efficiency, especially at rated speed and load conditions.
3. Power Factor Correction: Synchronous motors can improve power factor, reducing energy consumption and improving overall system efficiency.

Cons:

1. Higher Cost: Synchronous motors may be more expensive than induction motors due to their more complex construction and control requirements.
2. Complexity: They require additional control systems for synchronization, increasing complexity and potential points of failure.
3. Limited Torque: Synchronous motors may have lower starting torque compared to induction motors, limiting their suitability for certain applications.

Servo Motors:

Pros:

1. Precise Position Control: Servo motors offer precise control over position, velocity, and acceleration, making them ideal for applications requiring accuracy.
2. High Torque-to-Inertia Ratio: They provide high torque relative to their size and weight, allowing for compact designs with high performance.
3. Feedback Control: Servo motors incorporate feedback mechanisms for closed-loop control, ensuring accurate motion and positioning.

Cons:

1. Cost: Servo motors can be more expensive than other motor types due to their precision and control capabilities.
2. Complexity: They require complex control systems, including feedback mechanisms and controllers, increasing system complexity and cost.

3. Limited Power Output: Servo motors may have lower power output compared to other motor types, limiting their use in high-power applications.

Stepper Motors:

Pros:

1. Precise Positioning: Stepper motors provide precise control over position and rotation, making them ideal for applications requiring accurate motion.
2. Open-Loop Control: They do not require feedback mechanisms for position control, simplifying control systems and reducing cost.
3. High Holding Torque: Stepper motors can hold a specific position without the need for continuous power, enhancing stability in stationary applications.

Cons:

1. Limited Speed: Stepper motors may have lower maximum speeds compared to other motor types, limiting their suitability for high-speed applications.
2. Resonance Issues: They can experience resonance issues at certain speeds, leading to vibration and noise, particularly in microstepping applications.
3. Higher Power Consumption: Stepper motors may consume more power than other motor types, especially in holding or stationary positions, potentially affecting energy efficiency.

Brushless DC Motors (BLDC):

Pros:

1. High Efficiency: BLDC motors offer high efficiency and power density, making them ideal for applications requiring high performance and energy efficiency.
2. Low Maintenance: They do not use brushes, reducing maintenance requirements and enhancing reliability.
3. Quiet Operation: BLDC motors produce less noise and vibration compared to brushed DC motors, making them suitable for noise-sensitive applications.

Cons:

1. Complex Control: BLDC motors require electronic commutation and control systems, increasing complexity and cost.
2. Cost: They may be more expensive than traditional brushed DC motors due to their advanced construction and control requirements.

3. Limited High-Speed Operation: BLDC motors may have limitations at very high speeds, potentially affecting performance in certain applications.

Motor Type	Pros	Cons
DC Motors	Simplicity	Brush Wear
	Variable Speed Control	Limited Lifespan
	High Starting Torque	Sparking
AC Induction Motors	Robustness	Limited Speed Control
	Low Cost	Lower Efficiency at Low Speeds
	High Reliability	Lower Power Density
Synchronous Motors	Precise Speed Control	Higher Cost
	High Efficiency	Complexity
	Power Factor Correlation	Limited Torque
Servo Motors	Precise Position Control	Cost
	High Torque-to Inertia Ratio	Complexity
	Feedback Control	Limited Power Output
Stepper Motors	Precise Positioning	Limited Speed
	Open-Loop Control	Resonance Issues
	High Holding Torque	High Power Consumption
Brushless DC Motors	High Efficiency	Complex Control

	Low Maintenance	Cost
	Quiet Operation	Limited High Speed Operation

Table 9

Taking this information into account and applying it to our use case, we believed best to use a stepper motor as they provide a very high degree of accuracy and stability. Though it does not move very fast, it does not need to move very fast as the motions we make with the motor are very very small. Furthermore, our entire rig was not that heavy and therefore we got away with not having a huge stepper motor to get enough torque to move our rig.

For our application in particular, we did not need an exceeding amount of torque, but what is good news is that though that torque will not be a pressing need, the fact that we already have to purchase a gear rig to control and manage the camera rig means that we would be able to leverage this to give us even more torque and precise control. By using a smaller ratio gear to interface with the larger ratio gear rig, we can immediately gain the accuracy and torque advantage, thus making it easier to get a clear image.

Also, many stepper motors have a good control interface and are not be exceedingly expensive. Stepper Motors are not particularly power efficient, and will have a good amount of power draw even while idle to maintain exact position. If we were building a mobile rig we would probably take this more into consideration, but since we are not, and are plugging this into a wall, power draw is not a concern for this project, but is absolutely a way this product could be improved should we continue iterating on our design after this course.

For our specific tracking needs we estimated that we needed to achieve approximately 15 Arcseconds of accuracy, which is equivalent to .004167 degrees. Most stepper motors do not have the proper number of steps in order to achieve that 15 arcseconds of accuracy. For this we needed to use a technique called microstepping.

3.12.2 Stepper Motors

Stepper motors are electromechanical devices that convert electrical pulses into precise mechanical motion. Unlike conventional DC motors, which rotate continuously when voltage is applied, stepper motors move in discrete steps, making them ideal for applications requiring accurate positioning control, such as 3D printers, CNC machines, and robotic systems.

At the heart of a stepper motor are coils of wire wound around a central shaft, known as the rotor. These coils are arranged in phases, often with two or more

sets. The rotor is surrounded by a stationary magnetic field generated by permanent magnets or electromagnets in the stator, the stationary part of the motor.

4 COIL STEPPER MOTOR DIAGRAM

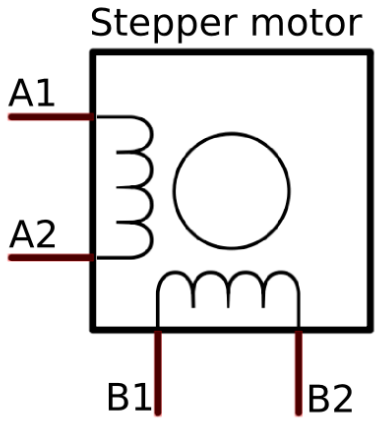


Figure 11

The operation of a stepper motor relies on the interaction between the magnetic fields of the rotor and the stator. When an electrical current flows through one of the coils, it creates a temporary magnetic field that interacts with the permanent magnetic field of the stator. Depending on the polarity of the current and the arrangement of the coils, this interaction produces a torque that causes the rotor to move to a specific position.

Stepper motors can operate in different modes, including full-step, half-step, and microstepping. In full-step mode, each pulse of current causes the rotor to move to the next step, resulting in relatively coarse positioning. In half-step mode, the current pulses are divided into smaller increments, allowing for finer resolution. Microstepping takes this further by applying varying current levels to the coils, enabling even smoother motion and greater precision.

The control of stepper motors is typically achieved through a stepper motor driver, which generates the necessary electrical pulses to move the motor shaft. These pulses are usually generated by a microcontroller or a specialized motion control system, which determines the speed and direction of rotation based on the desired application.

One of the key advantages of stepper motors is their ability to maintain precise positioning without the need for feedback sensors, such as encoders, which are

required in many other types of motors. However, this also means that stepper motors can lose synchronization if subjected to excessive load or acceleration, leading to missed steps and potential loss of position accuracy.

In conclusion, stepper motors are devices that convert electrical pulses into precise mechanical motion through the interaction of several distinct types of magnetic fields. Their ability to move in discrete steps makes them ideal for applications requiring hyper-accurate positioning control, and their versatility and reliability have made them indispensable in our project.

3.12.3 Microstepping

The following indent was written with the assistance of ChatGPT.

> Microstepping is an advanced technique employed in the control of stepper motors, offering finer control over their movement compared to traditional methods. Stepper motors typically operate by rotating a shaft in discrete steps, each step corresponding to a specific angle of rotation. In standard operation, these steps are relatively coarse, limiting the motor's ability to smoothly transition between positions.
>
> Microstepping addresses this limitation by subdividing each step into smaller increments, known as microsteps. Instead of simply energizing the motor coils fully for each step, microstepping involves applying intermediate levels of current to the coils, allowing for more granular control over the motor's position.
>
> The process of microstepping involves modulating the current in the motor windings using a technique known as pulse-width modulation (PWM). By varying the duty cycle of the PWM signal, the average current flowing through the coils can be controlled, resulting in smoother and more precise motion.
>
> Microstepping offers several advantages over traditional full-step or half-step methods. First and foremost, it enables smoother movement with reduced vibration and noise, as the motor transitions between microsteps more gradually. Additionally, microstepping allows for higher resolution positioning, improving the accuracy and precision of the motor's movements.
>
> While microstepping enhances the performance of stepper motors, it also introduces some challenges. Implementing microstepping requires more sophisticated control electronics and algorithms compared to simpler stepping methods. Additionally, the torque produced by the motor

decreases as the microstep size decreases, which can affect the motor's ability to maintain position under load.

Ideally, due to our low torque needs, this shortcoming of reduced torque while microstepping is negligible.

3.12.4 Stepper Motor Selection

Two types of motors would have needed to be chosen for this project, one motor for the focusing of the tracking lens, and two (of the same) motors for controlling the actual guiding and positioning of the camera.

Focusing Motor Selection

MIKROE-1530

Figure 12

Specifications:

The MIKROE-1530[25] motor is designed to operate at a rated voltage of 5VDC and is equipped with four phases. It features a speed variation ratio of 1/64 and a stride angle of 5.625°/64. Operating at a frequency of 100Hz, the motor maintains a DC resistance of 50Ω±7% at 25°C. It boasts an idle in-traction frequency exceeding 600 Hz and an idle out-traction frequency surpassing 1000Hz. The motor's in-traction torque measures greater than 34.3mN.m at 120Hz, with self-positioning torque also exceeding 34.3mN.m. Friction torque falls within the range of 600-1200 gf.cm, while the pull-in torque is 300 gf.cm. Insulated resistance is guaranteed to be over 10MΩ at 500V, with insulation capable of withstanding 600VAC/1mA/1s. Classified under insulation grade A, the motor experiences a temperature rise of less than 40K at 120Hz. Furthermore, it emits noise lower than 35dB at 120Hz under no load condition at a distance of 10cm. This motor model, known as MIKROE-1530, offers precise performance and reliability for various applications.

The MIKROE-1530 is be a strong contender because of its frequent restocking, ready availability, and attractive pricing ($8.00/each at time of writing), but

unfortunately it's actual function would very much be poor for our use case. Its Stride Angle is unacceptable as it is only 64 steps per rotation at 5.625 degrees per step before microstepping.

However, we have not only one camera but two cameras for our project. One high resolution camera for actual image capture and another, smaller, camera for the tracking and guiding and processing and moving of that larger high resolution capture camera. This smaller camera does not have an internal motor for an auto-focus and instead requires our hand to control the focus manually on the lens. For this we set up a gear mechanism connecting from a smaller stepper motor to the larger lens.

For this reason, this MIKROE-1530 would be and is an optimal candidate for an application that would require less precision, such as this adjustment of the guiding camera focus for our project, this part would be a perfect, affordable option. For this reason, we elected to use the MIKROE-1350 for the guiding camera focus mechanism.

	MIKROE-1350
Step Angle	5.625
Steps Per Revolution	64
NEMA	N/A
Holding Torque	>34.3 mNm @ 120Hz
Design	4 Wire Design

Table 10

Guiding Motor Selection:

SF2421-10B11

Figure 13

Specifications:
The SF-2421-10B11[34] stepper motor, manufactured by Sanyo Denki under the SANMOTION brand, is designed for precise motion control applications. It

belongs to the stepper motor product category and operates as a 2-phase motor with a NEMA 17 frame size. The motor features a bipolar connection for enhanced control.

With a minimum holding torque of 0.29 N-m and a step angle of 1.8 degrees, the SF-2421-10B11 offers reliable performance. It operates effectively under a voltage rating of 24 VDC and a current rating of 1 A. The rotor inertia is measured at 0.031E-4 kg-m-sq, ensuring smooth and stable operation during use.

Measuring 81 mm x 42 mm x 42 mm in dimensions, with a length of 81 mm, this motor is compact and suitable for applications with limited space. It belongs to the F2 series and is widely used in various industries for its reliability and precision. The motor utilizes 26 AWG wire gauge for its connections and weighs approximately 8.113011 ounces, making it easy to handle and install.

The SF-2421-10B11 is much more attractive than the previous motor. It is available, currently priced at $28.32, and semi-regularly restocked. It has a much more robust step angle at 1.8 degrees vs the MICROE-1350's 5.625. It also can handle a much higher degree of microstepping, which returns a significantly higher degree of accuracy. The microstepping allows for 256 step divisions giving 1.8/256 = .007 degrees of accuracy. This accuracy in arc-seconds would be 25.2 arc-seconds which is close but approximately 10 arc-seconds too many for our application as our application (controlling the pitch and yaw of the main camera) necessitates 15 arc seconds of accuracy.

STEPPERONLINE 17HM19-1684S

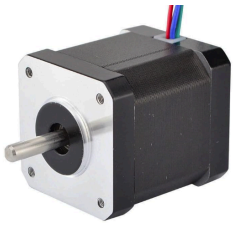


Figure 14

This stepper motor, called the 17HM19-1684S [6], operates as a bipolar stepper with a step angle of 0.9 degrees, ensuring precise movement control. It boasts a holding torque of 44 Ncm (62.3 oz.in) for stability and strength during operation, while each phase draws a rated current of 1.68 A with a phase resistance of 1.65 ohms. Its inductance is specified at 4.1 mH with a tolerance of +/- 20%, measured at 1 kHz. Physically, the motor is compact, with a frame size of 42 x 42 mm and a body length of 47 mm, featuring a 5 mm shaft diameter and a shaft length of 20 mm, including a D-cut length of 15 mm for convenient coupling. It is

equipped with four leads, each measuring 500 mm in length, and despite its robust performance, it maintains a lightweight design, weighing only 370 g.This is a high precision bipolar Nema 17 stepper motor with 0.9 deg. step angle (400 steps/revolution). Each phase draws current 1.68A, allowing for a holding torque of 44 Ncm(62.3oz.in)

The 17HM19-1684S stepper motor is a bipolar stepper motor featuring a step angle of 0.9 degrees, providing precise control over movement. With a holding torque of 44 Ncm (62.3oz.in), it offers sufficient force to maintain position under load. Rated at 1.68A per phase, it operates effectively within its specified electrical parameters. The motor's physical dimensions include a frame size of 42 x 42mm and a body length of 47mm, making it compact and suitable for various applications. Its 5mm diameter shaft extends 20mm, with a D-cut length of 15mm, facilitating easy coupling with external components. The motor is equipped with four leads, each 500mm in length, allowing for flexible wiring configurations. Weighing 370g, it strikes a balance between performance and portability, making it a versatile choice for robotics, automation, and other motion control systems.

For our application, the .9 degree stepping angle of 400 steps per revolution is incredibly valuable and is very useful for our application. As many of the other motors we researched have a stepping angle of 1.8 degrees therefore making them 200 steps per revolution. Without microstepping at all this makes the STEPPERONLINE 17HM19-1684S have a swing of 3240 Arcseconds, which is way high, but is very very achievable to achieve the goal of 15 Arcseconds of accuracy through proper microstepping.

	17HM19-1684S	SF2421-10B11
Step Angle	0.9 Degrees	1.8 Degrees
Steps per Revolution	400 steps/rev	200 steps/rev
NEMA	NEMA 17	NEMA 17
Holding Torque	44Ncm	29Ncm
Design	4 Wire Design	4 Wire Design
Current Draw	1.68A	1A

Table 11

Therefore we elected to choose the STEPPERONLINE 17HM19-1684S as our main driving motors for guiding our rig for spectral photography with extreme precision. All that would be left to do is to pick a stepper motor driver that can handle the microstepping.

3.12.5 Stepper Motor Drivers

In order to drive a motor, you most certainly need a motor driver.

Stepper motor drivers play a crucial role in controlling the movement and performance of stepper motors. They act as the intermediary between the control system (such as a microcontroller or computer) and the stepper motor itself, converting digital signals into the appropriate electrical currents needed to drive the motor.

Here are the key functions and features of stepper motor drivers:

1. Pulse Generation: Stepper motor drivers generate the precise sequence of electrical pulses required to move the motor shaft in discrete steps. These pulses determine the speed, direction, and extent of rotation.

2. Current Regulation: Stepper motors require specific levels of current to generate the necessary torque for movement. The driver regulates the current flowing through the motor coils to ensure optimal performance and prevent overheating or damage.

3. Microstepping: Our stepper motor drivers must support microstepping, a technique that divides each motor step into smaller increments. This results in smoother motion, reduced vibration, and improved positional accuracy. Microstepping is especially beneficial in applications requiring fine control and high resolution.

4. Direction Control: Stepper motor drivers determine the direction of rotation by controlling the sequence in which the motor coils are energized. By altering the order of coil activation, the motor can rotate clockwise or counterclockwise as needed.

5. Acceleration and Deceleration: Some stepper motor drivers offer acceleration and deceleration profiles, allowing for gradual changes in speed to prevent sudden jerks or overshooting. This feature is essential for smooth and precise motion control, particularly in applications with rapid changes in velocity.

6. Protection Mechanisms: Stepper motor drivers often incorporate protective features to safeguard the motor and the driver circuitry from overcurrent, overvoltage, and overtemperature conditions. These protections help prevent damage and ensure the longevity of the motor system.

7. Communication Interfaces: Many stepper motor drivers support various communication interfaces, such as UART, SPI, or I2C, enabling seamless

integration with different control systems and devices. This flexibility facilitates remote control, monitoring, and automation of stepper motor operations.

Zooming in, microstepping and by extension, accuracy is without a doubt the most critical component for selection of our stepper motor driver. Stepper Motor Drivers control the microstepping by doing the following:

Stepper motor drivers achieve microstepping by precisely controlling the current flow through the motor coils in a manner that generates intermediate positions between full steps. Here's a simplified explanation of how stepper motor drivers control microstepping:

1. Current Regulation: The stepper motor driver regulates the current flowing through each coil of the stepper motor. By adjusting the current levels dynamically, the driver can control the strength of the magnetic field generated by each coil.

2. Phase Shifting: Microstepping involves shifting the phase relationship between the currents in different motor coils. Instead of energizing coils sequentially as in full-step operation, microstepping introduces intermediate phases where the currents in adjacent coils overlap or are slightly offset. This creates a smoother transition between steps and enables finer positioning.

3. PWM (Pulse Width Modulation): Many stepper motor drivers use PWM techniques to control the average current flowing through the coils. By rapidly switching the current on and off at a high frequency and adjusting the duty cycle, the driver can achieve varying levels of current and thus control the motor's torque and microstepping resolution.

4. Sinusoidal or Trapezoidal Control: Some advanced stepper motor drivers employ sinusoidal or trapezoidal current control waveforms instead of simple square waves. These waveforms mimic the smooth curves of sinusoidal motion, resulting in even smoother microstepping and reduced vibration.

5. Microstep Resolution Configuration: Stepper motor drivers typically offer settings to adjust the microstepping resolution, allowing users to choose the number of microsteps per full step. This configuration is often done through dip switches, jumpers, or software commands, enabling customization according to the specific requirements of the application.

6. Feedback and Closed-Loop Control (Optional): In some high-precision applications, stepper motor drivers may incorporate feedback mechanisms such as encoders to monitor the motor's actual position and adjust the microstepping accordingly. This closed-loop control enhances accuracy and compensates for any deviations or disturbances in the system.

By employing these techniques, stepper motor drivers achieve microstepping, enabling smooth, precise, and quiet motion control in various applications requiring high-resolution positioning. As we are selecting two different motors, we need to have two different motor drivers, because it is critical to get a motor driver that is tailored specifically to the motor of choice.

3.12.6 Stepper Motor Driver Selection:

Focusing Stepper Motor Driver Selection

As we are choosing the MIKROE-1350 for our motor driver selection, it was critical to get a motor driver that is compatible with the MIKROE-1350.

The MIKROE-1350 is a stepper motor that is already in possession of several of our team members. And alongside that MIKROE-1350 stepper motor, is a stepper motor driver that is designed to fit with the MIKROE-1350 and drive it exactly as needed.

Wifehelper ULN2003

Figure 15
Wifehelper describes their board as follows:[5]

- ULN2003 driver board uses high power chip for driving stepper motor Increased output expansion interface
- A, B, C, D four-phase LED indicates the status of the stepper motor work.
- It has a standard interface for plug and play
- A, B, C, D light-emitting diode instructions four-phase stepper motor work with the state of the standard stepper motor interface, can be used directly plug[ed into the motor and MCU]

From this driver it can divide the stepper motor into a total of 2048 steps, which is a resolution that is low, but as the application of this is a focusing motor and does not require nearly as much accuracy as something like tracking would, this resolution is perfectly acceptable. It does not need as much resolution as the margin for error when focusing on a star is high, plus when attached to the lens,

the motor is geared down significantly giving it an even greater advantage and resolution, not that it would be needed.

The Focal length keeps things in focus and the fact that it requires less and less precision to keep an object adequately in focus the further and further away an object is from the lens. When an object is further away, the difference in distances between various points on the object and the lens is smaller relative to the lens's focal length. This results in a broader depth of field, where objects appear relatively sharp over a wider range of distances. As a result, the tolerance for precise focus is greater because slight variations in focus are less noticeable due to the extended depth of field. Conversely, when an object is closer, the depth of field becomes narrower, requiring more precise focus to ensure sharpness across the smaller range of distances.

For this reason, paired with the fact this this board is also already in possession of several of our team members and that it is the de-facto standard of the MIKROE-1350 and that it is affordable, we have chosen to use the Wifehelper ULN2003 as our primary stepper motor driver for our focusing motor.

3.12.7 Guiding Stepper Motor Driver Selection

STEPPERONLINE DM332T Digital Stepper Driver

Figure 16

From STEPPERONLINE, they describe the DM332T as follows:

> The DM332T is a digital stepper drive with simple design and easy setup. By implementing advanced stepper control technology, this stepper drive is able to power 2-phase and 4 phase stepper motors smoothly with optimal torque and low motor heating & noise. Its operating voltage is 10-30VDC and it can output up to 2.2A current. All the micro step and output current are done via DIP switches. Therefore, the DM332T are ideal choices for applications requiring simple step & direction control of NEMA 11, 14, 16, 17 stepper motors.

This industrial-grade stepper driver offers exceptional performance and flexibility for various applications. With 8 selectable micro-step resolutions ranging from 400 to 12800 via DIP switches, it provides precise control over motor movement. Additionally, it features 8 selectable output current settings of 1A to 3.2A via DIP switches, ensuring compatibility with different stepper motor types. With a pulse

input frequency of up to 60 KHz, it enables swift and accurate motor control. Designed to suit Nema 17 and 23 stepper motors, it caters to a wide range of industrial needs, promising reliability and efficiency in operation.

Tofelf TB6600

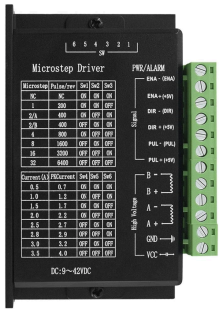


Figure 17

From Tofelf they describe the TB6600 as follows.

> The TB6600 upgraded driver is a professional 2/4 phase hybrid stepper motor driver, compatible with various brands. Suitable for 4/6/8 wire 2-phase hybrid stepping motors with current of 4.0A and below. Suitable for all kinds of small and medium-sized automation equipment and instruments, such as: engraving machine, marking machine, cutting machine, laser phototypesetting, plotter, numerical control machine tool, holding and placing device, etc.

The signal input for this device is single-ended and operates via pulse/direction. It features optocoupler isolated signal input, providing strong anti-interference capability. Subdivision options range from 1/2/4/8/16/32 microstepping, enabling it to drive 4/6/8 wire 2 phase/4 phase hybrid stepper motors with currents up to 4.0A and compatible with step angles of 1.8° and 0.9°. Output current ranges from 0.5A to 4.0A with a maximum output of 4.0A, while the input voltage ranges from 9-42VDC, allowing direct connection of external signals from 3.3-24V without the need for a series resistor. The device is designed with safety features including overheat, overcurrent, undervoltage lockout, and input voltage protection against reversal. Additionally, it automatically halves the current at standstill. With high compatibility, it suits all brands of 4/6/8 wire 2/4 phase hybrid stepper motors and is ideal for various small and medium-sized automation equipment and instruments such as engraving machines, marking machines, cutting machines, laser illumination, plotters, and CNC machine tools.

	DM332T	TB6600
Microstepping	400-12800	1-32
Compatibility	0.9° - 1.8°	0.9° - 1.8°
Output Current	1A - 3.2A	0.5A - 4A
Input Voltage	10 VDC - 30 VDC	9 VDC - 42VDC
NEMA 17 Compatible?	Yes	Yes

Table 12

From both of these we found the DM332T to be the more attractive option because of its electrical compatibility with the STEPPERONLINE 17HM19-1684S and its ability to microstepping with a resolution of up to 12800 microstepping. This, combined with the .9 degree step angle of the STEPPERONLINE 17HM19-1684S should be more than enough accuracy for our application.

Therefore we chose the STEPPERONLINE DM332T to be our stepper motor driver for the guiding motors of our project.

3.12.8 A Note on the "Stretch Board"

It was part of the original plan of this project to enable our system to be able to handle a custom mount, and to build that custom mount. This is why the Main PCB was enriched with an extra power line and an FPC connector with connectivity with the Raspberry Pi Compute Module 4. As the Main PCB was deemed "significant" enough to satisfy the Senior Design Requirements, we shifted our focus away from the Stretch Board (for Motor Control of a Custom Mount) and completely toward the Main PCB.

We still feel it necessary to still include the discussion of the motors, motor drivers, and the stretch board as a whole as it explains several components of the design of the Main PCB.

3.13 Microcontrollers

3.13.1 An Overview of Microcontrollers:

The following written with the assistance of ChatGPT

Microcontrollers (MCUs) are compact integrated circuits (ICs) that consist of a processor core, memory, and various peripherals, all embedded on a single chip. They are designed to execute specific tasks within electronic devices, ranging from simple consumer gadgets to complex industrial machinery. Here's an overview of what MCUs are and what they do:

1. Embedded Processing: MCUs are designed to provide processing power for embedded systems. Unlike general-purpose computers, which are versatile but larger and more power-hungry, MCUs are optimized for specific tasks and are usually smaller, consume less power, and are more cost-effective.

2. Processor Core: At the heart of every MCU is a processor core, which executes instructions and controls the operations of the device. Depending on the application, MCUs can feature various types of processor cores, including simple 8-bit cores for basic applications and more powerful 32-bit or even 64-bit cores for advanced tasks.

3. Memory: MCUs typically have two types of memory: program memory (ROM or Flash) and data memory (RAM). Program memory stores the firmware or software that controls the MCU's operation, while data memory holds variables and other temporary data used during program execution.

4. Peripherals: MCUs include a variety of built-in peripherals to interface with the external world and perform specific functions. These peripherals may include analog-to-digital converters (ADCs) for reading analog sensor inputs, digital I/O pins for interfacing with other digital devices, timers and PWM controllers for generating precise timing signals, communication interfaces such as UART, SPI, I2C, Ethernet, USB, and wireless protocols like Bluetooth and Wi-Fi.

5. Low-Power Operation: Many MCUs are designed for low-power operation, making them suitable for battery-powered or energy-efficient applications. They often incorporate features like sleep modes, where non-essential components are turned off to conserve power, and low-power operating modes that reduce the overall power consumption of the device.

6. Real-Time Operation: MCUs are often used in real-time systems, where timely responses to external events are critical. They are capable of executing tasks with predictable timing characteristics, ensuring that critical operations are completed within specified time constraints.

7. Application Areas: MCUs find applications in a wide range of industries and devices, including consumer electronics (such as smartphones, smartwatches, and home appliances), automotive systems (such as engine control units and infotainment systems), industrial automation (such as robotics and process control), medical devices (such as pacemakers and insulin pumps), and IoT (Internet of Things) devices (such as smart thermostats and connected sensors).

Overall, MCUs play a vital role in powering the ever-expanding world of embedded systems, providing the processing power and versatility required to bring intelligence and functionality to a diverse range of devices and applications.

3.13.2 Microcontrollers and Motor Control:

The following written with the assistance of ChatGPT

Motor control refers to the manipulation of electrical energy to regulate the speed, torque, and direction of electric motors. This is essential in a wide range of applications, including robotics, automotive systems, industrial machinery, and home appliances. MCUs are integral to motor control systems, providing the intelligence and computational power needed to accurately control motor behavior.

There are several techniques for controlling motors, each with its advantages and applications. Some common methods include:

1. DC Motor Control: Direct current (DC) motors are widely used due to their simplicity and controllability. MCUs can regulate the speed and direction of DC motors by adjusting the voltage or pulse width modulation (PWM) signals applied to them.

2. Stepper Motor Control: Stepper motors are commonly used in applications requiring precise positioning, such as 3D printers, CNC machines, and automated systems. MCUs generate step pulses to control the rotation of stepper motors, allowing for accurate movement in discrete steps. This is the type of control that we used.

3. Brushless DC (BLDC) Motor Control: BLDC motors offer high efficiency and reliability, making them suitable for applications like electric vehicles, drones, and HVAC systems. MCUs synchronize the operation of the motor's three-phase windings using sensor feedback or advanced algorithms to achieve smooth and efficient rotation.

4. Servo Motor Control: Servo motors are widely used in robotics, remote-controlled vehicles, and camera stabilization systems. MCUs generate control signals to adjust the position of the servo motor's shaft, providing precise angular positioning and speed control.

The integration of MCUs into motor control systems offers several benefits:

1. Flexibility: MCUs can be programmed to support various motor types and control algorithms, allowing for flexibility in system design and adaptation to different application requirements.

2. Precision: MCUs enable precise control of motor speed, torque, and position, leading to improved performance and accuracy in motion control applications.

3. Efficiency: By optimizing motor control algorithms and energy usage, MCUs contribute to increased system efficiency and reduced power consumption, particularly in battery-powered devices.

4. Integration: MCUs often integrate multiple peripherals, such as analog-to-digital converters (ADCs), pulse-width modulation (PWM) controllers, and communication interfaces, streamlining the design and implementation of motor control systems.

5. Cost-effectiveness: The compact size and integration capabilities of MCUs result in cost-effective solutions for motor control applications, minimizing the need for additional components and reducing overall system complexity.

Microcontrollers (MCUs) are indispensable components in modern motor control systems, offering the computational power, flexibility, and integration needed to achieve precise and efficient control of electric motors. From simple DC motor speed regulation to complex servo positioning and BLDC motor synchronization, MCUs enable a wide range of motor control techniques across various industries and applications. As technology continues to advance, MCUs play an increasingly vital role in driving innovation and enhancing the performance of motorized systems.

3.13.3 Microcontroller Selection

Microchip ATMega328P

Figure 18

Microchip describes the ATMEGA328P as follows:[12]

> The high-performance Microchip picoPower® 8-bit AVR® RISC-based microcontroller combines 32 KB ISP Flash memory with read-while-write capabilities, 1024B EEPROM, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented Two-Wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.

By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching one MIPS per MHz, balancing power consumption and processing speed.

The ATmega328P is a high-performance, low-power AVR® 8-bit microcontroller featuring an advanced RISC architecture tailored for efficient processing. With 131 powerful instructions, many of which execute in a single clock cycle, it ensures swift and responsive operation. Its 32 x 8 general-purpose working registers facilitate versatile computing tasks, operating fully statically for stability. With up to 16 MIPS throughput at 16 MHz and an on-chip 2-cycle multiplier, it offers robust computational capabilities. The microcontroller boasts high-endurance non-volatile memory segments, including 32 KB of in-system self-programmable flash program memory, 1 KB EEPROM, and 2 KB internal SRAM, providing ample storage for program and data. It supports true read-while-write operation and includes programming lock for software security, ensuring data integrity and protection. Peripheral features encompass two 8-bit Timer/Counters, one 16-bit Timer/Counter, real-time counter, six PWM channels, 8-channel 10-bit ADC, and various communication interfaces such as USART, SPI, and I2C, enabling seamless connectivity. Special microcontroller features include power-on reset, programmable brown-out detection, interrupt capabilities, and six sleep modes for optimized power consumption. The ATmega328P offers 23 programmable I/O lines and is available in 32-lead TQFP and 32-pad QFN/MLF packages, catering to diverse application needs. Operating voltage ranges from 2.7V to 5.5V, with an automotive temperature range of -40°C to +125°C, ensuring reliability in harsh environments. It exhibits low power consumption, with active mode drawing 1.5 mA at 3V and 4 MHz, and power-down mode consuming only 1 µA at 3V, contributing to energy efficiency and prolonged battery life.

Microchip ATMega328PB

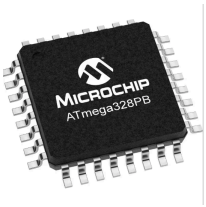


Figure 19

Microchip describes the ATMEGA328PB as follows:[13]

The high-performance, low-power Microchip 8-bit AVR® RISC-based microcontroller combines 64 KB ISP Flash memory with read-while-write capabilities, 2 KB EEPROM, 4 KB SRAM, 27 general purpose I/O lines, 32 general purpose working registers, two flexible timer/counters with compare modes and PWM, one UART with HW LIN, an 11-channel 10-bit A/D converter with two differential programmable gain input stages, a 10-bit D/A converter, a programmable watchdog timer with an internal individual

oscillator, SPI serial port, an on-chip debug system, and four software selectable power saving modes. The device operates between 2.7-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching one MIPS per MHz, balancing power consumption and processing speed.

The ATMEGA328PB[14] has a set of 131 instructions enabling most operations to be executed in a single clock cycle, ensuring efficient processing. With 32 x 8 general-purpose working registers, it facilitates versatile computing tasks. Operating fully statically, it maintains stability while offering up to 20 MIPS throughput at 20 MHz. Equipped with on-chip 2-cycle multiplier and 32 KB of in-system self-programmable Flash program memory, alongside 1 KB EEPROM and 2 KB internal SRAM, it offers ample storage and programming capabilities. The device supports true read-while-write operation and features programming lock for software security. Additionally, it incorporates capacitive touch buttons, sliders, and wheels, with 24 self-cap channels and 144 mutual cap channels for enhanced user interaction. Its timer/counters, including two 8-bit and three 16-bit ones, coupled with a real-time counter and ten PWM channels, enable precise timing and control. The microcontroller also integrates multiple communication interfaces, including two USARTs, two SPI serial interfaces, and two byte-oriented two-wire serial interfaces, offering versatile connectivity options. With a programmable watchdog timer, interrupt capabilities, and various sleep modes, it optimizes power consumption while maintaining responsiveness. The device operates within a wide voltage range of 1.8-5.5V and a temperature range of -40°C to 105°C, offering versatility in different environments. Its low power consumption in various modes, such as 0.24 mA in active mode, 0.2 µA in power-down mode, and 1.3 µA in power-save mode, ensures energy efficiency. Additionally, it features a clock failure detection mechanism and individual serial number for unique identification, further enhancing its reliability and security. Available in 32-pin TQFP and QFN/MLF packages, it caters to diverse application needs.

It should be noted that these two chips are remarkably similar. In fact the ATMEGA328PB is the in production updated version of the ATMEGA328P. They share the same architecture just in a different form factor, and are both, totally sufficient for our needs.

	ATMEGA328P	ATMEGA328PB
Flash Memory	32 KB ISP with read-while-write	64 KB ISP with read-while-write
EEPROM	1 KB	2 KB
SRAM	2 KB	4 KB
General Purpose I/O Lines	23	27

A/D Converter	6-channel 10-bit (8 in TQFP/QFN packages)	11-channel 10-bit
Communication Interfaces	USART, SPI, I2C	Two USARTs, two SPI, two I2C
Operating Voltage	1.8-5.5V	1.8-5.5V
Temperature Range	-40°C to +125°C	-40°C to +105°C

Table 13

3.13.4 A Note on the Use of the Development Boards

Neither the Arduino Uno R3[8] nor the Raspberry Pi[32] were present in the final design of the STSC-MS.

For the final and more developed iterations of our project, we utilized a PCB which had our central brain and intelligence allowing for proper interfacing with our peripherals as well as power distribution. Prior to the printing of this PCB as well as in case of iterations of the PCB being needed due to PCB failure, it was useful to the Computer Engineering side of the project to have some degree of development capability in the meantime in case of failure.

For this reason, we have elected to use the ATMEGA328P MCU as it is affordable, meets all of our requirements and as an added bonus is also already present and used in the Arduino Uno R3. This allowed for low level software development to begin immediately.

Arduino describes their UNO R3 development board as follows

> Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Again, in our final design, neither the Arduino nor the Raspberry Pi were present. Their early use in the design stage allowed for more time lenient prototyping and the advantage of an existing infrastructure. Once it was determined more clearly exactly which infrastructure is needed and not needed, these boards were “cannibalized” and ported to our PCB design with full integration with all the other peripherals required as well as the power distribution.

3.14 Software Languages

With the variety of programming languages in the world, there are multiple options as to which languages we could use for this project. The languages that best suited our expertise and objectives were C and Python.

3.14.1 The C Programming Language

The C language is a very commonly used language in the computer science industry. It is almost impossible to learn how to code without crossing paths with C at least once. C is best known for its simplicity, efficiency, and adaptability and is great for system programming as well as application development.

Attributes of C include its procedural programming model, low-level design, portability, structured programming, static typing, memory management, standard library, use of pointers, and its extensibility.[45]

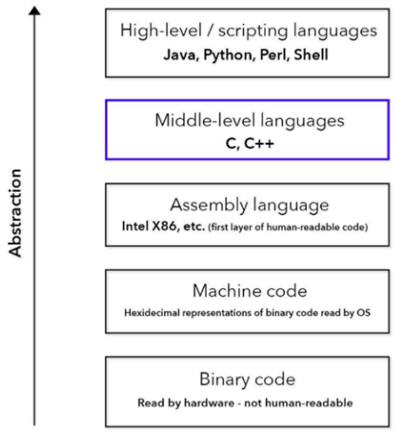
Procedural Programming:

C is a language that follows a structured program. This simply means there is organization and purpose behind every command and line in C code.

Low-Level Design:

C is labeled as a low-level design because of how close it is to the hardware of a computer system. C's low-level design gives the language the ability to directly manipulate addresses, which is great for system-level programming. Below is a diagram that represents how close programming languages are to the hardware of a system. Typically, the "easier" the language, the further away it is from the hardware. Notice how Python is in the highest level, only proving that it is one of the easiest languages to use.

Programming Language Levels



[47]

Figure 20

Portability:

The portability of C is one of the many reasons it is so commonly used. C is portable across various systems and requires very little adjustments. The availability of C compilers for a wide range of platforms and the language's platform-independence allows C to be portable.

Structured Programming:

Structured programming involves loops, conditional statements, and functions. These properties create much more organized and maintainable code while reducing the cost of memory and complexity.

Static Typing:

C requires that variable types be explicitly declared when code is being compiled. C is able to detect errors early in the development phase because of this.

Memory Management:

Programmers have manual control over memory which is an attribute caused by the previously mentioned low-level design.

Standard Library:

C's standard library contains a variety of functions with common tasks such as input/output operations, string manipulation, memory allocation, and mathematical computation.

Pointers:

Pointers are what allow programmers to manipulate memory addresses. As useful as pointers are, they can be complicated, so it is important to pay close attention to how you are manipulating memory.

Extensibility:

C can be extended through external libraries and modules that permit programmers to use pre-built functions which makes development much more efficient.[45]

3.14.2 The Python Programming Language

The other programming language we used in our design was Python. Python has become one of the most commonly used programming languages for anyone that is picking up programming for the first time. Many programmers find Python to be one of the easier languages to learn and work with because of its higher-level script language. Python being high-level just means it is further away

from the actual hardware of the system and therefore contains an easier syntax. Python's simplicity and extensive standard library have also made it a very desirable programming language.

Python contains many advantages such as its simplicity and reading and learning it. Python's simplicity in syntax creates greater focus on actually developing your program instead of worrying about syntax structure. Its simplicity also reduces maintenance cost. Python is a language that is widely adaptable and used by scientists, mathematicians, as well as engineers. Python contains a vast library that contains memory management capabilities. The language contains a private heap that contains all Python objects and data structures, as well as a built-in memory manager. Python utilizes asynchronous coding which prevents the use of complex research contentions, deadlocks, or other complexity. Python even includes libraries that integrate with other languages like C, Java, and C++. These libraries are called Cython and Jython and enable cross-platform development.

Python is a general-purpose, high-level programming language. It is best used for data analysis, data visualization, task automation, as well as web development. A few reasons people use Python are for its readability and maintainability, support of multiple programming paradigms, extensive library, high compatibility, simplified complex software development, open-source frameworks and tools, and test-driven development.[3]

Readable and Maintainable Code:

Python is a language that has no complex structures, unlike C++ or Java. Python also does not use delimiters, which is just a sequence of one or more characters that are used to specify the boundary of separate, independent regions in plain text or other data streams. Examples of this would be a comma, tab, or colon. Python also requires fewer lines of source code which is great when you want to update or maintain your code.

Support of Programming Paradigms:

Python supports object-oriented, procedural, and functional programming paradigms. This means that Python supports these methods of programming that are found in other common languages like Java, C, etc. Python supports code reusability, extensibility, abstraction, polymorphism, encapsulation, and inheritance. The functional paradigm is used to help programmers write more bug-resistant code.

Extensive Library:

Python contains a wide variety of packages and modules that enhance the functionality of the language. The extensive library capabilities were used in our project to help us not only analyze images, but also gather data from them.

High Compatibility:

One of Python's best features is its compatibility with different operating systems. Because it is a cross-platform and portable language, we can run code on one platform and use the same code on another platform without having to change the code at all.

Simplified Complex Software Development:

Python's syntax is very simple; it is almost like writing sentences in English. This is extremely useful when utilizing Python's abilities to develop scientific and numerical applications. These applications are great when you require data analysis or visualization.

Open-Source Framework and Tools:

Python being open-sourced helps programmers reduce the cost of their development. A variety of open-source frameworks, libraries, and other development tools are available that help programmers decrease development time.[3]

Test-Driven Development:

Python allows programmers to write and test their code almost simultaneously. This helpful feature allows for the formation of test cases before source code is even written.

Organizations that use Python include, but are not limited to: Google, Yahoo (Maps), YouTube, Microsoft, Spotify, Dropbox, Mozilla, Cisco, Facebook, and Quora.[30]

3.15 Machine Learning Programs

Machine learning is a very useful tool when handling a large amount of data. Because we are analyzing multiple images, we utilized machine learning's ability to gather and interpret data. More specifically, we looked at PyTorch and TensorFlow.

Both PyTorch and TensorFlow originate from tech giants within the Python deep learning world, assisting computers in solving real-world challenges with human-like capabilities. The primary difference lies in their approach to code execution, with PyTorch showing a tighter incorporation with the Python language.

TensorFlow contains robust visualization features, production-ready deployment pathways, and compatibility with mobile platforms. In contrast, PyTorch receives praise for its simplicity, dynamic computational graphs, and efficient memory utilization.

The debate over whether TensorFlow or PyTorch reigns supreme comes down to individual objectives. TensorFlow shines in constructing AI-driven products, while PyTorch appeals more to research-centric developers. PyTorch advances project initiation, yet TensorFlow offers superior capabilities for expansive projects and intricate workflows. Ultimately, the choice hinges on the specific goals and requirements of each endeavor.[19]

3.15.1 PyTorch

PyTorch is an open-source deep learning program best known for its ease of use. PyTorch's ease of use is mainly due to its compatibility with the Python programming language. Python is very commonly used by machine learning developers and data scientists.[46]

PyTorch is a framework used to build deep learning models. Deep learning models are a type of machine learning that is used for image recognition and language processing. PyTorch is distinguished for its support for GPUs as well as its use of reverse-mode auto differentiation. This feature allows computation graphs to be modified almost effortlessly.

PyTorch is the combination of GPU-accelerated backend libraries from Torch and a Python frontend that focuses on readable code, rapid prototyping, and support for the widest variety of deep learning models. PyTorch allows programmers to use a familiar program while being able to output graphs.

A significantly useful feature is PyTorch's reverse-mode automatic differentiation. The innovation that is the reverse-mode differentiation is a tool that records operations that have been completed. Once completed, the tool is able to replay the operations backwards to compute gradients. This feature makes the program easier to debug and well-adapted to certain applications such as dynamic neural networks. PyTorch is even useful for prototyping because of the multiple possibilities of iterations.

One of the main reasons PyTorch is so popular is because it is written in Python. Python is well-known for its ease of use. Programmers love the language because of its imperative and define-by-run execution where operations are executed as they are being called. A few assets of PyTorch are its ability to rapidly prototype, its ease of use, as well as its flexibility.[46]

3.15.2 Key Benefits of PyTorch

-Utilizing Python as its base language, PyTorch seamlessly integrates with popular Python libraries such as NumPy, SciPy, and Cython, facilitating scientific computing and enhanced performance through Python-to-C compilation

-With robust support from leading cloud platforms, PyTorch is well-established in cloud environments

-TorchScript, its scripting language, offers ease of use and flexibility, particularly in eager mode. This mode executes operations immediately, transitioning seamlessly to graph model for optimized performance in C++ runtime environments

-Supporting CPU, GPU, and parallel processing, PyTorch facilitates distributed training across multiple cores and GPUs on various machines

-Dynamic computational graphs allow runtime network behavior modifications, providing unmatched flexibility compared to static definitions in other frameworks

-The PyTorch Hub offers a repository of pre-trained models, readily accessible with minimal code

-Custom components can be effortlessly developed as subclasses of standard Python classes, with seamless integration of parameters with external toolkits like TensorBoard.

-Extensible APIs empower users to augment core functionality with ease

-Supporting both eager and graph modes, PyTorch caters to both experimentation and high-performance execution

-A vast array of tools and libraries spanning computer vision to reinforcement learning enhances PyTorch's versatility

-Its C++ front-end interface, reminiscent of Python syntax, facilitates the creation of high-performance C++ applications[46]

How PyTorch Operates:

PyTorch and TensorFlow share a fundamental structure centered around tensors and graphs.

Tensors:

Serving as the backbone of PyTorch, tensors are similar to multidimensional arrays, facilitating storage and manipulation of model inputs, outputs, and parameters. Unlike NumPy's ndarrays, tensors are GPU-compatible, accelerating computational tasks.[46]

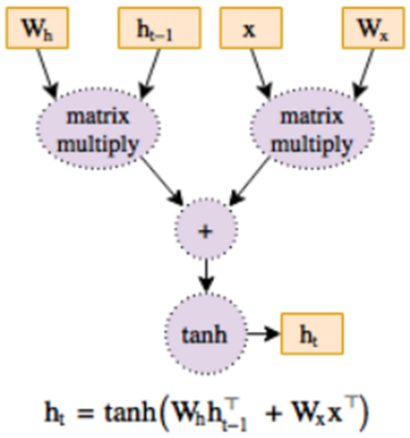
Graphs:

Central to deep learning, neural networks employ a sequence of nested functions to process input data. The objective is to optimize parameters—comprising weights and biases stored as tensors in PyTorch—by computing gradients with respect to a loss metric. During forward propagation, the network yields confidence scores for subsequent layers until the output layer, where error

computation occurs. Backpropagation, part of gradient descent, iteratively adjusts weights based on computed errors, refining the model.

Graphs, as data structures, consist of interconnected nodes (vertices) and edges. Every modern deep learning framework relies on graph representations of neural networks. PyTorch maintains a Directed Acyclic Graph (DAG) of tensors and operations, with Function objects capturing executed operations. Input tensors serve as leaves, while output tensors serve as roots.

Directed Acyclic Graph used in PyTorch



[46]

Figure 21

Unlike static computation graphs in frameworks like TensorFlow, PyTorch adopts dynamic computation graphs. These graphs are constructed and updated at runtime, allowing seamless integration of forward pass computations with backpropagation data structures. PyTorch stands out as the first define-by-run deep learning framework, matching the performance of static graph counterparts like TensorFlow, making it suitable for various applications, from convolutional to recurrent neural networks.

PyTorch Applications are renowned for its convenience and adaptability, PyTorch finds application across diverse domains, including reinforcement learning, image classification, and natural language processing.[46]

3.15.3 TensorFlow

TensorFlow is another open-source machine learning framework that is designed to facilitate deep learning, neural networks, and general numerical computations, TensorFlow operates seamlessly across CPUs, GPUs, and GPU clusters. Its standout feature lies in its expansive open-source community comprising developers, data scientists, and engineers, all actively contributing to its repository. TensorFlow can be regarded as one of the most widely adopted AI engines in present use.

TensorFlow caters to a wide spectrum of tasks encompassing numerical computation, large-scale machine learning, deep learning, and various statistical and predictive analytics endeavors. This technology streamlines the implementation of machine learning models for developers, facilitating data acquisition, scalable predictions, and iterative refinement processes.[19]

TensorFlow excels in training and executing deep neural networks for diverse applications, ranging from handwritten digit recognition to image classification, word embeddings, and natural language processing (NLP). Its software libraries contain code snippets adaptable to any application, enhancing its ability to learn and perform these tasks.

TensorFlow's versatility extends to its compatibility with different hardware platforms. It seamlessly operates on both conventional CPUs and high-performance GPUs. Additionally, owing to its development by Google, TensorFlow integrates seamlessly with the company's tensor processing units (TPUs), purpose-built to accelerate TensorFlow workloads.

While Python serves as the primary front-end API for building applications within the framework, TensorFlow offers wrappers in various languages, including C++ and Java. This language flexibility empowers users to swiftly train and deploy machine learning models irrespective of their preferred programming language or platform.

TensorFlow joins various machine learning and deep learning models and algorithms, rendering them accessible through a unified interface.

Programmers leverage TensorFlow to construct dataflow graphs wherein computational nodes symbolize mathematical operations. Interconnections between these nodes represent multidimensional arrays, known as tensors.

Although Python serves as the front-end API for TensorFlow, the actual mathematical computations occur outside Python's domain. Behind the scenes, high-performance C++ binaries execute these operations, while Python orchestrates communication between components using high-level programming abstractions.

TensorFlow applications exhibit remarkable versatility, capable of running on a multitude of platforms, including iOS and Android devices, local machines, cloud clusters, and various hardware configurations like CPUs, GPUs, or Google's custom TPUs available on Google Cloud.

Offering a comprehensive suite of APIs, TensorFlow caters to diverse user needs. While Google advocates for the high-level APIs to streamline data pipeline development and application programming, the low-level APIs (termed TensorFlow Core) prove invaluable for debugging and experimentation purposes.

TensorFlow serves as a streamlined platform for developing and deploying advanced analytics applications, catering to a diverse user base including data scientists, statisticians, and predictive modelers.

Adopted by businesses of all sizes, TensorFlow plays a pivotal role in automating processes and pioneering new systems, with its prowess evident in large-scale parallel processing tasks like neural networks. Moreover, it finds application in experimental domains such as self-driving vehicle research and testing.[19]

Unsurprisingly, Google, the brain behind TensorFlow, employs it extensively for internal operations. From enhancing search engine capabilities to powering applications for automated email responses, image classification, and optical character recognition, TensorFlow remains integral to Google's technological ecosystem.

An advantage of TensorFlow lies in its abstraction capabilities, allowing developers to focus on application logic while the framework handles intricate details. Additionally, TensorFlow simplifies debugging and introspection processes, enhancing developer convenience.

TensorBoard, part of TensorFlow's suite, offers an interactive web-based dashboard facilitating graph inspection and profiling. Furthermore, TensorFlow's eager execution mode enables transparent evaluation and modification of individual graph operations, fostering enhanced flexibility and control.

For seamless incorporation, Databricks Runtime for Machine Learning incorporates TensorFlow and TensorBoard, eliminating the need for separate package installations, thereby enhancing accessibility and ease of use.[19]

Due to conditions out of our control, we unfortunately could not utilize the tools found in machine learning programs. Astrophotography relies on clear night skies and because of the increase in cloudy and rainy nights faced in Orlando, we could not capture enough data to train the programs.

In summation, both PyTorch and TensorFlow would have been very important programs for our group to utilize not only for their ease of use, but also because of the countless features each program possesses. Deep learning programs are great programs that can analyze data from the images we obtain and would have allowed us to gather information that we required.

3.16 Classification Algorithms

To gather the information we desire from the images viewed through the telescope, we require a classification algorithm. A classification algorithm simply gathers input features and values, analyzes these values, merges the information all together, and labels the values for its output. Classification algorithms were

very important to us because they were necessary for classifying astral objects and determining their temperatures, weight, and other features.

Classifying any image requires a neural network which is a mathematical model composed of layers, and within those layers are objects called neurons. Before classifying images, the system would first need some previously entered images to pull from. This way the classification algorithm knows what features to look for in astral objects and can successfully classify them.[17]

Inspired by the function of a brain, a neural network model consists of an input layer, at least one hidden layer, and an output layer. The input layer would take in the pixels of an image, for example, and each matrix of pixels would be given a value. Each neuron contains a value known as the weight, and this process continues until the entire initial image is analyzed. Each input neuron is then connected to a neuron in the following hidden layer. Some neural networks can contain hundreds or thousands of hidden layers and can contain a variety of neurons. The most common neural network is known as a “feed-forward” model which means the outputs from a neuron only connect to subsequent layers.[17]

The value of a new neuron is determined from the sum of weights and neuron values from the previous layer. Every neuron’s value is multiplied by a weight that connects the old neuron to the new neuron. Not all neurons have the same weight, therefore each new neuron should yield a new value. If a new weight is below a given threshold value, then the neuron passes no data into the following layer. Once the neuron values are multiplied by this weight, the network adds these values together and this results in a single number value that becomes the new neuron’s new value. This process is repeated for every layer until it reaches the output neuron. The output neuron is a singular neuron that is composed of all the weights and neuron values from the final hidden layer. The equation for a hidden layer is as follows:

$$a_j = f\left(\sum_{i=1}^{N} w_{ij} * x_i + b_j\right)$$

where

a_j represents the value of a neuron (also known as an activation function),

j represents a neuron,

$f(\theta)$ represents the activation function in the hidden layer which transforms the input signal so that it varies between 0 and 1,

w_{ij} represents the weight of an input,

x_i represents a set of inputs,

b_j represents a bias term that simply determines the input level at which a neuron is activated.[17]

The output layer of the neural network is represented by the following equation:

$$y_k = g(\sum_j w_{jk} * a_j + b_k)$$

$$y_k = g(\sum_j w_{jk} * f(\sum_i w_{ij} * x_i + b_j) + b_k)$$

where

y_k represents the value of the output of the neural network,

$g(\theta)$ represents the activation function in the output layer,

w_{jk} represents the weight of an input,

b_k represents a bias term that determines the input level at which a neuron is activated.[17]

Once each neuron in the final hidden layer is processed to create the output neuron, the neural network has created a new image that is essentially a copy of the input image. Below is a representation of how a neural network looks:

Representation of a Neural Network

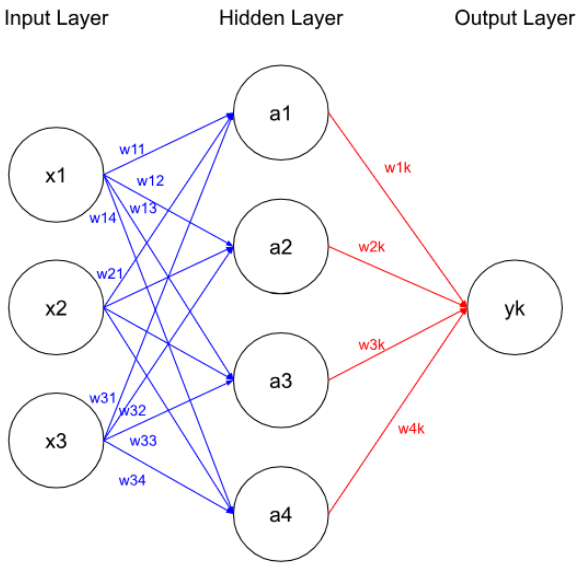


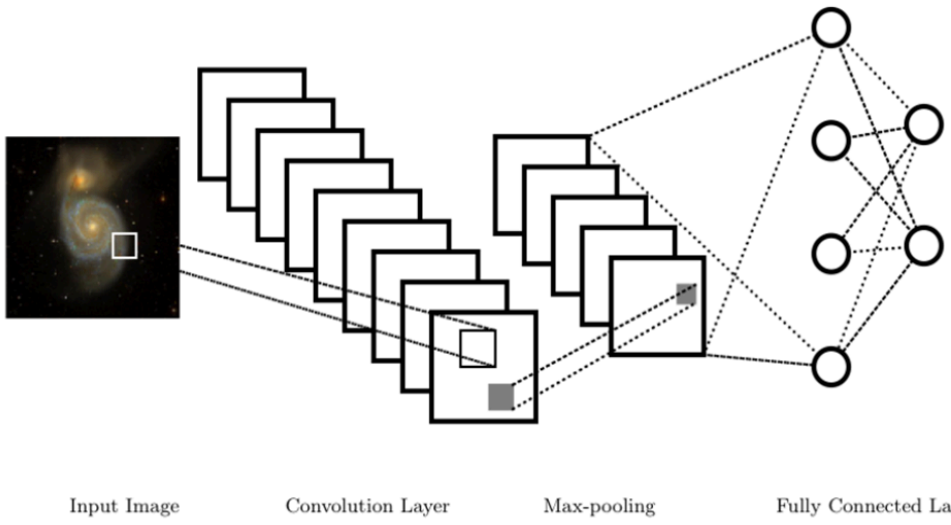
Figure 22

The process of setting up and using a neural network can be coded using the Python programming language in Siril. Using Siril would have allowed us to create the neural network by training a network, loading visualization functions, loading training samples, and validating the sample. The code to create such a model is extensive but very effective with the use of frameworks such as TensorFlow, PyTorch, or Keras. Each of these neural network frameworks are accessible in Siril.

TensorFlow is the most commonly used framework of the three. It is more developed and contains tools that aid in inspecting a network. PyTorch has a better learning curve in comparison to TensorFlow. It also creates dynamic networks whereas TensorFlow creates static networks. This means that PyTorch can adjust the network quickly, making it easier to implement changes in the input. PyTorch is also an object-oriented framework which leads to less lines of code and easier understanding of the code. Keras, on the other hand, is an Application Programming Interface (API) written over TensorFlow. This means that it is a framework that allows you to use TensorFlow easier.[17]

A specific type of neural network that is commonly used with images is the Convolutional Neural Network (CNN). CNNs reduce complexity by requiring that neurons only respond to inputs of an image from a smaller portion of the image. In other words, the CNN breaks down an image into smaller and smaller pieces and analyzes said pieces instead of the image in its entirety. Doing so decreases the computational complexity of the network. The building blocks for a CNN include a convolutional layer, non-linear activation function, a pooling layer, and a fully connected layer. Below is a diagram that depicts how a CNN works:

Representation of a Convolutional Neural Network



[17]

Figure 23

The way a CNN works is a kernel is used to convolute an image input. The size of the kernel is determined by 2 factors called depth and stride. Depth is the number of kernels and stride is the number of pixels a kernel shifts at each step of the convolution. If we view the image as a matrix *N* x *M*, once the convolution with the kernel is complete, we get a “data cube” of feature maps with dimensions *N* x *M* x *K*.[17]

Once all the pixels have gone through the convolution, we now can focus on the non-linear activation function. This function is applied to every individual pixel and creates feature maps. The pooling layer uses these feature maps and generalizes the values within a given region (for example, a 3x3 window of pixels). Typically, the highest pixel value within that region is preserved during the down sampling.

Pooling is an important feature in CNNs because it reduces the size of the resulting network and makes the network easier to analyze since the images would contain less distortions. The last layer of the CNN is the fully connected layer. The fully connected layer creates a set of labels and links the output of the network to those labels. In this layer, the outputs connect to the neurons of the classification layer.[17]

The use of these neural networks is important because they are the most efficient way to analyze an image. When using the telescope, we used a camera to capture the images of astral objects. These images not only need to be saved but analyzed to determine numerous features like size, temperature, distance, etc. They also allow a computer program to do the work that would fall to error if a human were to classify astral objects with their eyes alone.

3.17 Stellarium

Stellarium is an open source software program that allows you to see images of the sky from your laptop. Think of it as Google Earth, but for space. Like a planetarium, Stellarium allows you to view a realistic sky in 3D, much like how one might view it with the naked eye, a telescope, or binoculars. It is compatible with all Linux, macOS, and Windows systems. Stellarium was used as a reference as to what we wanted our user-interface to look like and how we wanted it to function.[39]

Stellarium contains many useful features that involve the sky, its interface, visualization, as well as customizability:
(information pulled directly from stellarium.org)
Sky:

- -Default catalog of over 600,000 stars
- -Extra catalog with more than 1 million deep-sky objects
- -Asterisms and illustrations of the constellations
- -Constellations for 40+ different cultures
- -Calendars of 35+ different cultures
- -Images of nebulae (full Messier catalog)
- -Realistic Milky Way
- -Realistic atmosphere, sunrise, and sunset
- -Planets and their satellites
- -All-sky surveys (DSS, HiPS)

Interface:

- -Powerful zoom
- -Time control
- -Multilingual interface
- -Scripting interface
- -Fisheye projection for planetarium domes
- -Spherical mirror projection for your own low-cost dome
- -Graphical interface and extensive keyboard control
- -HTTP interface (web-based control, remote control API)
- -Telescope control

Visualization:

- -Several coordinate grids
- -Precession circles
- -Star twinkling
- -Shooting stars
- -Tails of comets
- -Eclipse simulation
- -Supernovae and novae simulation
- -Exoplanet locations
- -Ocular view simulation
- -3D sceneries
- -Skinnable landscapes with spherical panorama projection

Customizability:

-Plugin system adding artificial satellites, ocular simulation, telescope control and more

-Ability to add new solar system objects from online resources

-Can add your own deep sky objects, landscapes, constellation images, scripts[39]

Stellarium's Graphical User Interface

[39]

Figure 24

In this image, we can see Stellarium's graphical user interface. The buttons along the left-hand side and the bottom of the image include some of the features used to change what the user sees.

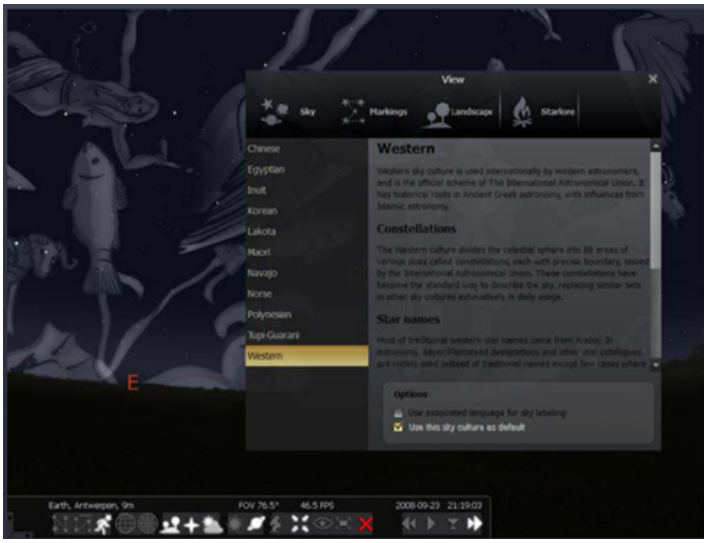
Stellarium's Location Window

[39]

Figure 25

This image displays Stellarium's location window. In this window, the user has to set their observing position by simply searching the city they would like to be in. This window also allows the user to determine location just by inputting longitude, latitude, and altitude.

Stellarium's Starlore Tab

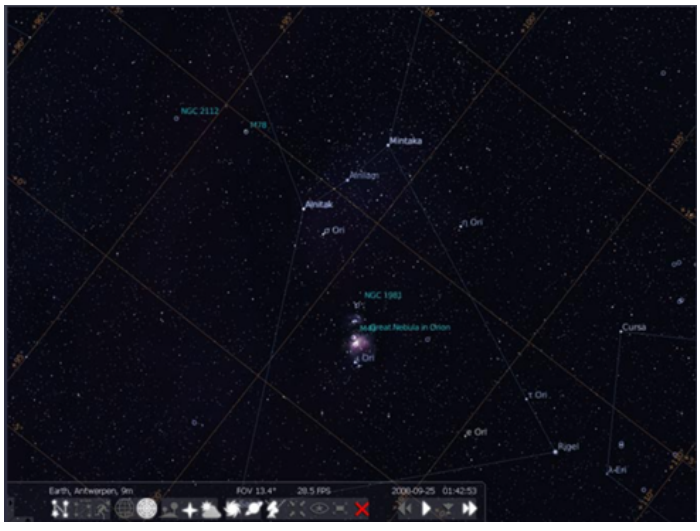


[39]

Figure 26

This window is the "Starlore" tab. Stellarium included this feature to provide information and history to see how different cultures looked at the stars.

Constellations in Stellarium

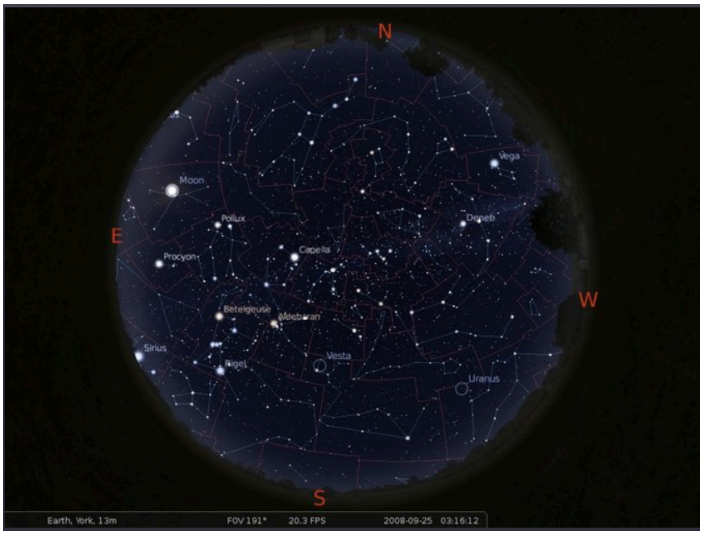


[39]

Figure 27

This image is an example of how Stellarium shows constellations. This image is of the great nebula in Orion. Stellarium allows you to view nebula lines as well as constellation lines.

Constellations in Stellarium



[39]

Figure 28

This image is a full sky view of constellations and their boundaries within the Milky Way.

Stellarium is a great guide for some of the features we may include in our software. Although it contains many useful features, we do not plan on fully using the tools Stellarium contains. The plan is to utilize the important tools found in Stellarium but strictly keep it as a reference.

3.18 Auto-Focusing

Achieving precise focus is critical in astrophotography to capture sharp and detailed images of celestial objects. Auto-focusing systems help streamline this process by automating the adjustment of focus based on feedback from imaging sensors or dedicated focusing routines. Common types of auto-focusing systems used in astrophotography include:

Motorized Focusers:

Motorized focusers attach to the telescope's focuser mechanism and allow for remote adjustment of focus using a motorized controller. They are compatible with various telescopes and provide precise control over focus.

Software-Based Autofocus:

Some astrophotography softwares, including Siril, offer built-in autofocus routines that analyze image sharpness and adjust focus accordingly. These software-based solutions streamline the focusing process and can be integrated into automated imaging sequences.

Electronic Focusers:

Electronic focusers interface directly with imaging cameras or computers, allowing for electronic control of focus adjustments. They offer convenience and precision, particularly when integrated with astrophotography software for automated focusing.

Auto-focusing systems help astrophotographers achieve optimal focus quickly and accurately, reducing the time spent on manual adjustments and maximizing imaging efficiency.

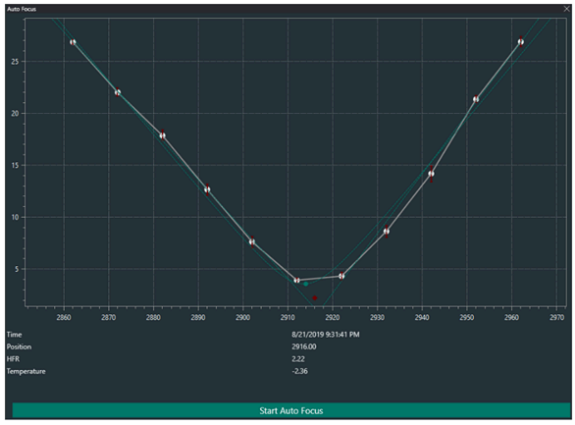
With poor focusing, images may result in softer and a lower SNR (signal-to-noise-ratio). It is most common for astrophotographers to focus with precision tools like the Bahtinov mask before imaging. Unfortunately, most optical assemblies suffer from focus shift throughout the night either because of temperature or other reasons. On top of this, the use of monochrome sensors mixed with narrowband (or broadband) filters leads to a need for refocusing between filters. Thus, the necessity for auto-focusing.[16]

3.18.1 N.I.N.A.

Night-time Imaging N Astronomy (N.I.N.A.) is one example of an open source software that astrophotographers use for autofocusing. N.I.N.A. automates most of the tasks required to capture deep sky images. It is unique in that it has various methods to autofocus on star fields, bright objects (like planets), or on terrestrial objects.[16]

N.I.N.A. uses two methods to measure how far from focus an image is. The first method is called Star HFR. Using this method, N.I.N.A. moves out of focus, takes an exposure, detects the stars in the image, computes the Half Flux Radius (HFR) of the stars and the average HFR across the frame. After moving the focuser by a certain amount, N.I.N.A repeats the process until a focus curve is available and the minimum, which is the best point of focus, can be found by different types of fitting. An example of this curve can be seen below:

Fitting of Star HFR Curve using N.I.N.A.



[16]

Figure 29

Each focus point represents the HFR at its respective focus position. Meanwhile the red bars on each point represent the potential error on each focus point. In order for the Star HFR to work, the stars must be in view. If stars maintain short exposures or are far from focus, the software will have some issues.

The second method N.I.N.A. uses to measure how far an image is from the focus is called contrast detection. Like how a cell phone performs its autofocus; this method simply analyzes the contrast within the image. With N.I.N.A., contrast is measured with different techniques. A curve is formed from the results of the image and resembles a Gaussian Curve, which is a frequency distribution symmetrical about the line of zero error and in which the frequency and magnitude of quantities are related by an expression. In this curve, the maximum is the point where the contrast is highest and results in the best focus. An example of these results can be seen below:

Fitting of Contrast Detection Curve using N.I.N.A.

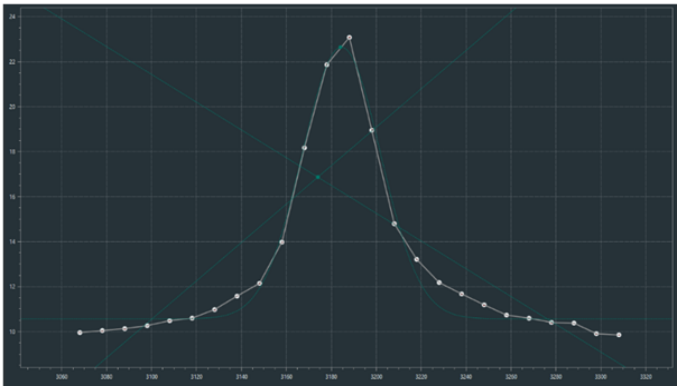

[16]

Figure 30

Using contrast detection does not rely on the detection of stars. Therefore, exposure times and computing time both decrease in comparison to Star HFR. Gathering points near the best focus would be ideal because if the maximum is too narrow, then it is possible to miss it.

3.18.2 N.I.N.A. Auto-Focus Algorithm

N.I.N.A. takes an exposure at a focuser position and computes the Star HFR. This is used as a threshold for comparison. The final focus point is compared to this threshold to make sure the better focus was found. The focuser is then moved outwards (higher focuser position value). Next, the focuser begins moving inwards (lower focuser position value) and measures the contrast or HFR at each step. The focuser continues to move inward until a minimum (for HFR) or maximum (for contrast) point is found.

The focuser also moves a small amount to the sides of the minimum or maximum. If there are not enough focus points, to the right of either the minimum or maximum, the focuser is moved back outwards to the rightmost point, and proceed until it has enough points. Next, a fitting is performed on the points in order to determine the point of best focus, and the focuser moves to it. For Star HFR, a final exposure is taken to compare to the previously mentioned threshold. If the HFR is worse than the threshold by 15% or more, the routine is seen as a failure and the focuser returns to its initial focus position. Either this is the outcome, or an additional auto-focus run is attempted.

Astrophotography continues to captivate enthusiasts worldwide, thanks to advancements in technology and dedicated software tools like Siril. Combined with a diverse array of telescopes and auto-focusing systems, astrophotographers have access to powerful tools for capturing stunning images of the cosmos. As technology continues to evolve, the future of astrophotography holds exciting possibilities for both amateurs and professionals alike.[16]

To produce an effective auto-focus, parameters need to be set for the autofocus algorithm. The three important parameters N.I.N.A. focuses on are the initial offset steps, step size, and exposure time.[16]

Initial Offset Steps:

A good default value for the Star HFR would be 4. For contrast measurement, a good default would be 6.

Step Size:

One method in finding the best step size would be to start with the focuser slightly out of focus from the best focus. You would then move the focuser outwards by about ten steps. If there is an obvious difference and the star diameter has increased between 20-30%, then this would most likely be the right step size.

Another method to find the best step size would be to start from a good focus position and progressively move the focuser inward or outward until N.I.N.A. cannot determine stars HFR within the image. The total increment would represent the maximum initial offset.[16]

3.18.3 Exposure Time

For exposure time, it is best to take an exposure near the best focus on an area with many stars. For the Star HFR, the best exposure time would have to be an exposure time that almost saturates the brightest stars but keeps the majority under a saturation threshold. With this method, stars should be easily visible. The next step would be to enable the HFR measurement so the stars can be detected. If the stars are detected, exposure time should be fine.

To make sure the stars are properly detected, you should move the focuser out of focus. Once this is done, take a frame with the same exposure time as before. Within this frame, check to see if some of the out of focus shapes are still bright. If some of the brightest stars are correctly detected, this is how you know you have the right exposure time.[16]

Chapter 4 Standards and Design Constraints

4.1 Production Standards

ANSI/ASQ Z1.4-2003 is an essential standard for establishing quality control measures and sampling procedures. This standard, adopted from MIL-STD-105E—developed by Harold F. Dodge—underscores a robust framework for inspecting samples from extensive batches. The objective is to assess the overall quality of a lot, including finished products, in-process items, operational efficiency, quality control, and maintenance protocols.
Thorlabs' implementation of ANSI/ASQ Z1.4-2003 has yielded numerous advantages, lower sampling costs,time efficiency, and elevated accuracy levels; adherence to such a standard fosters a rigorous, data-intensive approach, boosting confidence in product integrity while minimizing resource expenditure. By aligning with this standard, Thorlabs ensures the consistent delivery of high-quality optical components and positions itself as a competitively priced, reliable supplier in the marketplace, making it our top option for purchasing optical components.

The role of cleanrooms is paramount for optical component manufacturing, particularly for optical coatings for products such as dielectric mirrors. Thorlabs has certified **ISO Class 10,000 and Class 1,000 cleanrooms.** These classifications indicate the stringent control over environmental conditions, specifically the number of airborne particles per cubic meter. While typical ambient air contains approximately 1,000,000 particles per cubic meter, Thorlabs' cleanrooms drastically reduce this figure to 10,000 and 1,000 particles per cubic meter.
The necessity for such controlled environments stems from the delicate nature of optical coatings, which can be as thin as a few nanometers. Contamination by particles, even as small as 0.1 microns, can significantly impair the integrity and performance of these coatings. To mitigate such risks and elevate the quality of the final products, Thorlabs implements additional quality control procedures that surpass conventional cleanroom standards. These include ultrasonic cleaning to remove microscopic contaminants and hand inspecting each optical component. Through these practices, Thorlabs ensures the production of high-quality optical components.[24]

4.2 Communication Standards

The Universal Serial Bus (USB) standards are established by USB-IF which is a non-profit corporation which developed Universal Serial Bus specifications. The standards are explored and highlighted in the **IEC 62680-2-3** document which defines the pinout, transfer rate, and electrical specification, protocols and much more which must be adhered to. These standards serve as the foundation for USB based communication and help establish seamless communication between peripherals and define expected values making USB communication reliable and consistent.

These core standards established in the IEC 62680-2-3 work as the foundation for our ability to reliably and efficiently control our EOS Rebel T7. Making sure to follow the designated pinouts and keeping the transfer rate in mind were able to develop a system which leverages this USB design standards to achieve imaging via the accompanying SDK. The transfer rate helps us define accurate processing times by calculating a rough estimate of how long each image should take to move from the camera to the Raspberry PI as well as helps us understand what input delay to expect between sending the signal to take the image and the camera beginning to develop the image. The USB 2.0 system works to make the entire idea possible as this is the only way we can communicate with the camera which drives our main vision of being able to classify stars.

The Camera-Serial Interface CSI-2 standards are another set of communication standards which played a major role in establishing communication between the guiding and tracking camera and our Raspberry PI. The standards for the CSI-2 Interface are created and defined by the MIPI alliance and have been adopted by IEEE board as of 2021-06-16 as stated in their **IEEE 2977-2021** document. These standards serve as the base to define the electrical specifications,register definition and pinouts which the Camera-Serial Interface used.

The CSI-2 document focuses on defining the APPI standards which is the interface between the upper-layer and the A-PHY. It also defines the data link layer which is the reference to follow for how the scheduling prioritization and packet forwarding should operate. The physical layer is the last thing spoken about in the scope of this document. The Physical layer standards describe how the data should be encoded and decoded and how packets are extracted.

Table 1 below describes the expected transfer rate based on the architecture defined by the standards as a benchmark for performance. Gears define the implementation profile and technical specifications that the CSI-2 profiles should follow. The Raspberry PI Compute Module 4 has all 5 Gears available which allows us to leverage up to 15 Gbps of transfer speeds.

The technical specifications for the Coaxial cable that transfers the information between the interface are also defined. In this section expected impedances, voltage levels, and loss across frequencies is defined. These points served as the baseline for choosing a high quality connector that falls within these standards allowing us to reduce the loss described as the quality of the cable degrades.

The hardware architecture is also defined in the file explaining the control flow in the Master/Slave architecture. The Master is defined as the Source Port which in our case is triggered via the Raspberry PI module and the Sink is the Raspberry PI High Quality Camera. Packet fields are also established and work as a major insight to the data transfer and used for debugging purposes given we encounter

issues. The first 28 bits are the A-Packet header which define the SCI, TX-Delay, and the Message Counter and A-Header CRC fields. The A-Packet Tail is a 32 bit field which has the payload and will be where we receive our image data from. These fields and their descriptions are defined in the Table 16 Below.

4.3 Memory Standards

SD Card will be the storage option of choice due to its high storage capacity and high transfer speeds. The standards are defined by SD Card association; they state both the transfer protocol speeds and limitations and the physical layer attributes. The simplified documents of both are Physical Layer Simplified Specification and UHS-II Simplified Addendum define the standards and implementation notes for defining the hardware architecture.

The ***Physical Layer Simplified Specification Ver9.10 20231201*** defines the Capacity of Memory for each particular SD Card Type. The Raspberry PI Compute Module 4 supports the Extended Capacity SD Memory Card (SDXC) which will be the primary focus of the standards discussed. This is defined as allowing for more than 32GB and including up to 2TB of storage which will be more than suitable for our application. We relyed on their defined SD Express Memory Card standards which denote the system operating voltage VDD1 as sitting between 2.7V and 3.6V, and VDD2 sitting between 1.70V and 1.95V. The physical layer standards also define the SD Card types that we used for the Read/Write card which are defined as blank cards used for mass data storage, end user video audio or digital image recordings.

The standards also dictate the SD Bus Protocol; they define the protocol sequencing which focuses is broken down into 3 separate sections. Command which is the token that starts the operation. Response which is what is expected after each command verifies the command was received. Lastly, the Data protocol which specifies that data is transmitted via the data lines as a result of a successful command. The SD Bus Protocol standards also define how data packets should be established. There are two formats to be followed: Usual Data (8-bit width) and Wide Width Data (SD Memory Register). The breakdown for the data packets are defined in the image below.

The document also indicates the physical pinouts and size requirements of SD Cards. The SD Card standard defines a 9 pin architecture which is 24mm x 32 mm x 2.1 mm or 24 mm x 32 mm x 1.4 mm. The pinouts are defined by the table below. We chose to use the SD Mode pinout as it allows for faster transfer rates as well as asynchronous data transfer due to its multiple data lines.

Transfer speeds play a crucial role in our system design. In order to create an accurate and responsive system the need to uphold the strict transfer rate standards and attempt to generate as little loss in our design is essential. The Raspberry Pi Compute Module 4 is compatible with the Ultra High Speed Phase I (UHS-I) transfer card protocol. The maximum transfer speed we can achieve on

UHS-I is 104 MB/sec. This will work sufficiently to transfer to store and access our images. The expected image size is roughly 5 MB per image meaning we can expect an approximate transfer time of 48ms between each read and write. This transfer speed will be suitable for our less time intensive applications and work as the source for storing required long term images and software. This standard also defines how often we can take images and will play a crucial point in establishing optimal timings of images.

The bus mode configurations are defined by the standards document and give further insight into the different operating modes expected from the SD cards (UHS-I) architecture. The system design will be using the SDR104 bus speed mode as it allows for the highest transfer rate of 104MB/sec while operating at 1.8v and only requiring a clock of 208MHz which our Raspberry Pi Compute Module 4 is more than capable of providing.

The standards as defined by SD Card Association also rigorously define the required function and features which are supported on the hardware. They define everything from setting transfer Cache best practices, Boot Functionalities, Security protocols, etc. This portion of the standards will give us better insight into how we can optimize our system and get the most out of the rich feature set which is provided by the SD card architecture. Since, this will be our primary way of storing large files required by the project and allowing us to meet our image processing needs, leveraging the standards will play a crucial role in creating a deterministic result within our system.

The final major topic which needs to be addressed via the standards placed by the SD Card Association are those which define the SD Memory Card Hardware interface. Our embedded system design will require us to meet these standards rigorously in order to create a stable data transfer system. This section defines the bootup time of when the SD card would be ready for its first command which it places at 1ms. This time will play a key role as attempting to communicate via our hardware implementation before this time will result in data loss and therefore a loss of accuracy or even failure to complete the desired process. The Standards also define the base operating voltages of different pins such as the required base voltage of the power pin before the 1ms time has elapsed for the initial boot. This pin must be kept below 0.5V for more than 1ms to allow for a proper power cycle to occur. The reset of the SD Card if needed is also defined. We are required to keep that min below 0.5V for more than 1ms again in order to have a proper power off cycle. The power ramp up required for this system is defined as needing to be monotonic, with a ramp up time no longer than 35ms for the 2.7-3.6V power supply. It also defines a need to wait for the VDD to reach a stable voltage before accomplishing any communication with the device in order to prevent data loss. The last power requirement states that the VDD must be stable for at least 74 cycles before issuing the first command.

4.4 Optical Standards

The U.S. standard **MIL-PRF-13830B** assesses optical component surface quality by establishing a methodology to evaluate and quantify surface imperfections. This standard facilitates maintaining uniformity across optical component manufacturers. It introduces two metrics for evaluating surface quality: the scratch number and the dig number, which range from 10 to 80 in increments of 10, offering a scale for grading the severity of surface imperfections.

The scratch number is determined by comparing the visual brightness of surface imperfections to a calibrated reference under specific lighting conditions—either a standardized 40W incandescent or a 15W cool white fluorescent lamp. This comparison provides a quantifiable metric of scratches. On the other hand, the dig number evaluates the size of the largest surface flaw, such as a pit, bubble, or inclusion, measuring it in hundredths of a millimeter. For instance, a dig number of 80 corresponds to a flaw size of 0.8mm.

These metrics collectively inform the overall quality assessment of an optical component, dictating whether it meets the requisite standards for acceptable quality. For scratches, the standard stipulates that the maximum allowable scratch size should not exceed one-quarter of the optic's diameter. Each optic must meet a scratch specification defined by the manufacturer, calculated by multiplying the number of scratches by the ratio of each scratch's length to the optic's diameter.

For digs, the standard sets a maximum specification that the sum of an optic's dig numbers must not surpass, typically double the specified maximum. Additionally, it mandates that only one maximum-sized dig is permissible per 20 mm of optic diameter, and all digs of at least 0.1 microns must be spaced at least 1 mm apart. Generally, scratch-dig specifications fall within the range of 40-20, ensuring a balance between manufacturing practicalities and the need for high-quality optical surfaces. This standard underscores the industry's commitment to producing high-quality optical components by defining clear, measurable criteria for surface imperfections.[43]

ISO 10110-8 addresses surface roughness on optical components, a parameter integral to controlling light scattering. This characteristic is particularly vital in laser systems, where excessive light scatter can pose safety risks and significantly impair the performance of high-power optical systems. In our context, minimizing surface roughness is essential for maximizing signal retention and reducing losses in our optical system.

Surface roughness refers to small, high-frequency spatial variations on an optic's surface, with features sometimes as small as a few angstroms. ISO 10110-8 introduces specific polishing grade designations, categorizing the polishing quality based on the number of micro defects per 10mm sampling length. For

instance, a P1 designation indicates 80-400 micro defects per length, whereas a P4 designation corresponds to fewer than three micro defects.

Beyond polishing grades, the standard further dissects the surface characteristics of an optic into three additional metrics: figure, waviness, and roughness. These terms are distinguished by their scale of measurement:

- Figure pertains to deviations measured in tenths of millimeters to centimeters, representing the macroscopic curvature and flatness of the optic's surface.
- Waviness encompasses mid-scale irregularities measured in micrometers and millimeters.
- Roughness, the most minute scale, covers imperfections from angstroms to tens of micrometers.

The methodologies employed for measuring these features are tailored to their specific scales:

- A conventional interferometer is utilized for assessing the figure,
- A white light interferometer for waviness,
- An atomic force microscope (AFM) for the fine details of roughness.

This approach to quantifying surface characteristics under ISO 10110-8 ensures a comprehensive understanding of an optic's surface quality, enabling the production of components that meet stringent performance criteria.[44]

Chapter 5 Comparison of ChatGPT with other Similar Platforms

The following was written with the assistance of ChatGPT:

In recent educational history, the use of AI-driven programs has become more and more common. Programs like ChatGPT offer unique opportunities for collaboration, problem-solving, and ideation. But, even with the number of benefits a program like ChatGPT can provide, it has its faults as well. However, it is crucial to assess their limitations and advantages in comparison to similar tools. In this analysis, we discuss ChatGPT and compare it with other platforms that are very similar, discussing their pros, cons, and real-world implications.

ChatGPT, run by OpenAI's GPT technology, is a conversational AI platform capable of generating human-like text responses based on the user's input prompts. ChatGPT can adapt to various tasks and answer countless questions, from solving math problems, writing JavaScript code, to creative writing. But like any tool, it has its pros and cons.

The pros of ChatGPT include versatility, accessibility, time efficiency, and continuous improvement. For versatility, ChatGPT can be employed across a spectrum of tasks, including brainstorming, code generation, and problem-solving. For accessibility, its user-friendly interface enables seamless interaction, making it accessible to individuals with varying levels of technical expertise. For time efficiency, ChatGPT speeds up idea generation and problem-solving by providing instantaneous responses. For continuous improvement, as part of OpenAI's ecosystem, ChatGPT benefits from ongoing updates and enhancements, ensuring its relevance and efficacy.

The cons of ChatGPT include bias and inaccuracy, lack of contextual understanding, and limited creativity. For bias and inaccuracy, ChatGPT is susceptible to biases present in its training data, leading to occasional inaccuracies or inappropriate responses. In terms of contextual understanding, while proficient in generating coherent text, ChatGPT struggles with nuanced contextual understanding, resulting in occasional misinterpretations. When it comes to creativity, ChatGPT's creativity remains constrained by its reliance on pre-existing patterns in the data it was trained on.

IBM Watson, another AI platform, contains a number of useful features including natural language understanding, machine learning, and data analysis.

The pros of IBM Watson include enterprise integration, advanced analytics, and domain expertise. For enterprise integration, IBM Watson provides robust integration capabilities, making it suitable for large-scale enterprise applications. For advanced analytics, its analytics features enable deep insights into data, facilitating informed decision-making. For domain expertise, with its vast

repository of knowledge, IBM Watson excels in specialized domains, offering tailored solutions to complex problems.

The cons of IBM Watson include complexity, cost, and customization challenges. For complexity, implementing IBM Watson solutions may require significant expertise and resources, posing a barrier to entry for smaller projects or teams. For cost, licensing fees associated with IBM Watson can be prohibitive, particularly for budget-constrained endeavors. And for customization, tailoring IBM Watson to specific use cases may involve complex customization processes, potentially extending project timelines.

The pros of Microsoft Azure's AI Services include scalability, integration with ecosystems, and pre-built models. In terms of scalability, Azure's cloud infrastructure offers scalability and flexibility, accommodating projects of varying sizes and complexities. For integration with an ecosystem, as part of the Azure ecosystem, its AI services seamlessly integrate with other Microsoft tools and services, streamlining workflows. For pre-built models, Azure provides a range of pre-built models and APIs, simplifying the development process for common AI tasks.

The cons of Microsoft Azure's AI Services include a learning curve, dependency on Azure, as well as cost management. For the learning curve, while Azure's offerings are comprehensive, navigating its ecosystem may require a learning curve, particularly for users unfamiliar with cloud technologies. For dependency on Azure, relying on Azure for AI services may lead to vendor lock-in, limiting flexibility and portability across different platforms. And for cost management, users need to be vigilant in managing costs associated with Azure services, as usage-based pricing models can result in unexpected expenses.

In addition to AI-driven platforms like ChatGPT, IBM Watson, and Microsoft Azure's AI services, Senior Design projects often leverage other platforms such as GitHub and Coursera. Let's examine how these platforms contribute to the learning experience and project outcomes.

GitHub is a collaborative platform for software development, facilitating version control, project management, and code collaboration.

The pros of GitHub include version control, code sharing and collaboration, and integration ecosystem. For version control, GitHub's version control system enables teams to track changes, collaborate on code, and revert to previous versions if needed, promoting project stability and transparency. In terms of code sharing and collaboration, by hosting code repositories on GitHub, teams can collaborate on projects in real-time, share code snippets, and review each other's contributions. For the integration ecosystem, GitHub integrates with a wide range of tools and services, including CI/CD pipelines, issue tracking systems, and project management tools, enhancing workflow efficiency.

The cons of GitHub include a learning curve, complexity in collaboration, as well as security risks. For students new to version control systems like Git and GitHub, the learning curve can be steep, requiring time and effort to grasp concepts and workflows effectively. For collaboration complexity, while GitHub streamlines collaboration, coordinating contributions from multiple team members and resolving conflicts can be challenging, especially in larger projects. And for security risks, GitHub repositories may be susceptible to security vulnerabilities, such as code injection or unauthorized access, necessitating robust security measures and best practices.

Coursera offers online courses and educational resources on a wide range of topics, including computer science, data science, and engineering.

The pros of Coursera include access to expert instruction, flexibility and convenience, and certification and credentialing. For access to expert instruction, Coursera courses are taught by industry experts and academic instructors, providing students with high-quality instruction and learning materials. For flexibility and convenience, with on-demand courses and flexible schedules, Coursera accommodates students' diverse learning needs and preferences, allowing them to learn at their own pace. For certification and credentialing, completing Coursera courses can lead to verified certificates and credentials, which students can showcase on their resumes and portfolios, enhancing their professional profiles.

The cons of Coursera include cost considerations, self-discipline challenges, and limited interaction. In terms of cost, while Coursera offers many free courses, access to premium features and certifications may require subscription fees or one-time payments, which can be prohibitive for some students. For self-discipline challenges, online learning platforms like Coursera require self-discipline and motivation to stay engaged and complete course requirements, and students may struggle with procrastination or lack of accountability. And for limited interaction, despite efforts to foster community and collaboration through discussion forums and peer assessments, online courses on Coursera may lack the interpersonal interaction and feedback found in traditional classroom settings.

In Senior Design projects, the choice of AI platform can significantly impact the project's outcomes and overall experience. Here are a few examples where these websites may influence a student’s learning experience:

Example 1: Ideation and Brainstorming

Using ChatGPT, teams can quickly generate ideas and brainstorm solutions to design challenges. Its versatility and accessibility make it an ideal tool for fostering creativity and collaboration among team members. However, teams must be mindful of potential biases in generated responses and supplement ChatGPT with human judgment and critical thinking.

Example 2: Data Analysis and Decision-making

IBM Watson's advanced analytics capabilities empower teams to analyze large datasets and derive actionable insights. By leveraging Watson's domain expertise, teams can make informed decisions based on data-driven evidence. However, the complexity and cost associated with implementing Watson may require dedicated resources and expertise.

Example 3: Integration and Deployment

Microsoft Azure's AI services offer scalability and integration with existing workflows, enabling seamless deployment of AI solutions in Senior Design projects. Teams can leverage Azure's pre-built models and APIs to expedite development and deployment processes. Nonetheless, navigating Azure's ecosystem and managing associated costs requires careful planning and expertise.

In the ever-evolving landscape of Senior Design projects and projects in general, the choice of AI platform is a pivotal decision that shapes project outcomes and learning experiences. While each platform offers distinct advantages and challenges, the key lies in understanding their capabilities and aligning them with project requirements and team expertise. By critically evaluating the pros and cons of platforms like ChatGPT, IBM Watson, and Microsoft Azure's AI services, teams can embark on a journey of exploration and innovation, harnessing the power of AI to unlock new frontiers of knowledge and discovery.

Incorporating ChatGPT-related resources into big projects can offer several valuable advantages for students in Senior Design or similar endeavors.ChatGPT can serve as a powerful tool for sparking creativity and generating ideas. When faced with complex problems or design challenges in big projects, students can use ChatGPT to explore different approaches, brainstorm solutions, and refine project concepts. Its versatility in generating human-like text responses based on prompts can inspire innovative solutions and help students think outside the box.

During the iterative design process, students may need to generate a variety of textual content, such as project documentation, user interface text, or communication materials. ChatGPT can assist in drafting and refining these textual components, providing quick and flexible support for content creation. Moreover, it can aid in prototyping conversations or interactions within the project, helping students simulate user experiences and refine project functionalities.

In big projects, conducting thorough research and literature reviews is often a crucial aspect of the process. ChatGPT can assist students in summarizing complex research papers, generating literature review sections, or answering specific research-related questions. By leveraging ChatGPT's ability to

comprehend and generate coherent text, students can streamline the research process and extract key insights more efficiently.

When encountering technical challenges or roadblocks in big projects, students can use ChatGPT as a resource for problem-solving and troubleshooting. By providing detailed descriptions of issues or errors, students can prompt ChatGPT to generate potential solutions, debug code snippets, or offer guidance on resolving technical issues. This can help students overcome obstacles more effectively and enhance their problem-solving skills in the context of real-world projects.

In collaborative projects involving multiple team members, effective communication and coordination are essential. ChatGPT can facilitate communication within the team by generating meeting agendas, summarizing discussions, or providing feedback on project deliverables. Additionally, it can assist in drafting project-related emails, documentation, or presentations, ensuring clarity and consistency in communication among team members.

On the other hand, while ChatGPT and similar AI-driven resources offer numerous benefits, there are also potential drawbacks that students should consider when deciding whether to use them in projects. Relying too heavily on ChatGPT for critical aspects of a big project can lead to dependency issues. Students may become too reliant on ChatGPT for idea generation, problem-solving, and even content creation. This dependency can diminish a student's ability to think critically or creatively and develop independent solutions. Over reliance on ChatGPT also creates a lack of ownership and leads to more plagiarism.

While ChatGPT can generate human-like text responses, its outputs are not always reliable. Students should exercise extreme caution when using ChatGPT for tasks that require precision, expertise, or domain-specific knowledge. AI-generated content may contain errors, biases, or inaccuracies, particularly in complex or specialized subject areas. Students should verify and cross-reference AI-generated content with credible sources to ensure its accuracy and relevance to the project.

ChatGPT may lack nuanced contextual understanding, especially in complex or ambiguous situations. While it can generate coherent text based on input prompts, it may struggle to grasp subtle nuances, understand implicit meanings, or interpret context-specific requirements. Students may encounter challenges when using ChatGPT for tasks that require deep comprehension, empathy, or cultural sensitivity, such as user interface design, customer interaction, or ethical decision-making.

When using ChatGPT-related resources, students should be mindful of privacy and data security risks. AI platforms may store or analyze user-generated data, including input prompts and responses, which could raise concerns about data

privacy, confidentiality, or intellectual property rights. Students should review and understand the terms of service, privacy policies, and data usage practices of AI platforms to mitigate potential risks and protect sensitive information.

The use of AI-driven resources in big projects raises ethical considerations related to transparency, accountability, and bias. Students should be honest about the use of AI-generated content in their assignments and always acknowledge any contributions from AI systems.

So, while ChatGPT is a valuable website that can enhance productivity and creativity in big projects, students should do their best not to rely on ChatGPT or any AI related programs like it. By being aware of the potential drawbacks of AI-driven resources, students can make informed decisions about when and how to integrate them into their projects while ensuring the integrity, quality, and ethical responsibility of their work.

Large Language Models (LLMs) represent a revolutionary advancement in artificial intelligence, transforming the landscape of natural language processing (NLP) and human-computer interaction. The history of LLMs is a testament to the convergence of scientific innovation, computational power, and the vast wealth of digital data available in the modern era.

The roots of LLMs can be traced back to the mid-20th century when engineers began exploring the concept of machines that could understand and generate human language. Alan Turing's early work on computational linguistics and the development of rule-based systems, laid the groundwork for future advancements in NLP.

Later on in the 20th century, statistical approaches to language processing gained prominence. Techniques such as n-grams, hidden Markov models, and probabilistic context-free grammars enabled computers to analyze and generate text based on statistical patterns derived from large corpora of text data.

The resurgence of neural networks in the 21st century ushered in a new era of NLP research. Breakthroughs in deep learning, fueled by advancements in computational hardware and algorithms, enabled the development of more sophisticated models capable of learning complex representations of language.

One of the pivotal moments in the evolution of LLMs came with the introduction of the transformer architecture. Proposed by Vaswani et al. in their landmark paper "Attention is All You Need" in 2017, transformers revolutionized sequence modeling by leveraging self-attention mechanisms to capture long-range dependencies in input sequences.

The years following the introduction of transformers witnessed an explosion of research and development in the field of LLMs. Researchers pushed the

boundaries of model size, training data, and computational resources, leading to the creation of increasingly larger and more powerful language models.

Breakthroughs in scaling techniques, such as distributed training, model parallelism, and data parallelism, enabled researchers to train LLMs on unprecedented scales. Models like OpenAI's GPT (Generative Pre-trained Transformer) series and Google's BERT (Bidirectional Encoder Representations from Transformers) achieved remarkable performance by leveraging massive datasets and computational resources.

The versatility of LLMs has led to their widespread adoption across a myriad of domains and applications. From language translation and sentiment analysis to chatbots and content generation, LLMs have demonstrated their utility in various real-world scenarios, driving innovation and reshaping industries.

Despite their remarkable capabilities, LLMs are not without challenges and ethical considerations. Concerns related to bias, fairness, misinformation, and privacy have prompted researchers and practitioners to explore ways to mitigate these risks and ensure responsible deployment of LLMs.

The development and deployment of Large Language Models (LLMs) have raised profound ethical questions and considerations that have shaped the trajectory of artificial intelligence (AI) research and application. As LLMs have become increasingly powerful and pervasive, the discourse surrounding their ethical implications has evolved, reflecting society's growing awareness of the potential risks and benefits associated with AI technologies.

Early in the development of LLMs, ethical considerations primarily revolved around issues of data privacy, security, and algorithmic bias. Researchers and ethicists raised concerns about the potential misuse of personal data, the security implications of AI systems, and the risk of perpetuating and amplifying societal biases present in training data.

While LLMs can produce impressive outputs based on provided prompts, it is essential for researchers to contextualize and critically evaluate the generated results. Merely relying on LLM-generated content without understanding its limitations and potential biases can undermine the credibility of academic work. Researchers must engage in a thoughtful analysis of the generated text, considering its relevance, accuracy, and alignment with existing knowledge in the field.

Effective utilization of LLMs begins with providing clear and specific prompts tailored to the research objectives. Moreover, fine-tuning the model on domain-specific datasets can enhance the accuracy and relevance of generated content. By fine-tuning LLMs, researchers can adapt the models to the nuances

of their research domain, thereby improving the quality of generated output and facilitating deeper insights.

Incorporating LLM-generated content into academic documents necessitates rigorous verification and cross-referencing with reputable sources and existing literature. While LLMs can offer novel perspectives and generate new ideas, researchers must validate the accuracy and reliability of the generated content through traditional research methods. Cross-referencing LLM-generated text with established scholarly sources ensures the integrity and credibility of academic work.

A seamless integration of LLM-generated content with human-written text is paramount for maintaining coherence and consistency in academic documents. Researchers should carefully blend LLM-generated content with their own writing, ensuring that the overall narrative remains cohesive and adheres to academic standards. By harmonizing LLM-generated text with human-authored content, researchers can leverage the strengths of both approaches to produce high-quality academic work.

Ethical considerations dictate the proper attribution of LLM-generated content and acknowledgment of the model's contribution to the research process. Researchers should clearly attribute LLM-generated text to the model and provide appropriate citations if applicable. Transparently documenting the role of LLMs in the research process demonstrates academic integrity and ensures proper recognition of the model's influence on the final output.

The ethical use of LLMs in academic writing requires researchers to uphold principles of academic integrity, plagiarism avoidance, and respect for intellectual property rights. Researchers must exercise caution to avoid inadvertently plagiarizing content generated by LLMs and adhere to copyright laws when incorporating external sources. Additionally, ensuring the privacy and consent of individuals involved in the research process is paramount when utilizing LLMs for data analysis or text generation.

In conclusion, the integration of Large Language Models into academic writing offers researchers unprecedented opportunities for innovation and exploration. However, to maximize the benefits of LLMs while upholding academic integrity, researchers must adhere to best practices and ethical guidelines. By critically evaluating generated content, fine-tuning models, verifying results, and transparently documenting the research process, researchers can harness the full potential of LLMs to advance scholarly knowledge while maintaining the highest standards of academic rigor and ethics.

https://chat.openai.com

Chapter 6 Hardware Design

6.1 Power System Design

For this project, we needed enough power for 3 motors of two different types, 2 cameras, a ML processing chip, a motor control interface MCU, a display for a GUI, 3 stepper motor drivers and various other smaller peripherals. For this we required a substantial power source that can then be routed where power needs to go.

As the PCB design specifics come into focus, there will be a more robust integration of different circuits so that our PCB can be as compact and with minimized points of failure.

We should provide good power integrity to our Power integrity refers to the ability of a PCB to deliver clean and stable power to all components, minimizing noise and voltage fluctuations. Achieving good power integrity involves our placing of decoupling capacitors near power pins to suppress high-frequency noise and provide transient current, utilization of dedicated power planes for different voltage levels to minimize impedance and voltage drops, and avoiding crossing high-speed signal traces over power planes to prevent noise coupling.

It is also important to be conscious of the layering of the PCB and the placement of vias so that there is margin for error in manufacturing. It is also critical to have effective thermal management for preventing overheating of components due to power dissipation. We should position high-power components near thermal vias or heatsinks to facilitate heat dissipation.

For power system design, we primarily used the Texas Instruments Tool, WeBench. WeBench is a web based tool that allows for the input of a desired input voltage and a desired output voltage, and from there it will automatically output a series of designs and schematics for input into PCB CAD software.

An example of this can be seen below.

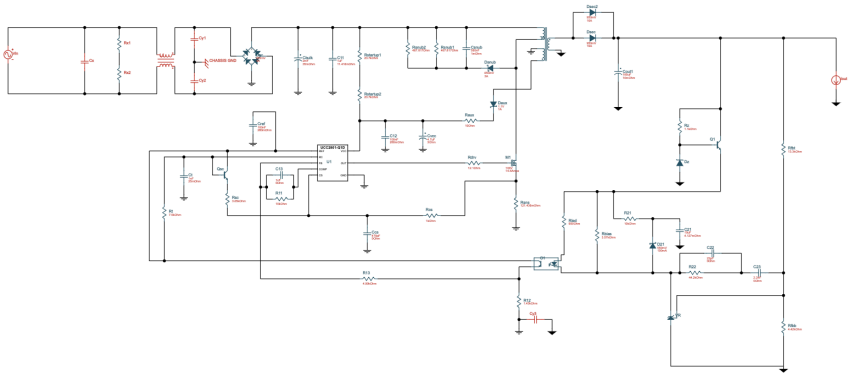


Figure 31

15Vrms -> 12VDC (Vin to Stepper Motor Driver)
Accessed from Texas Instruments WeBench

We were able to aggregate all of our power needs into a list of requirements and then, one by one, (as we did here for the motor driver) feed them into Webench and get a working circuit for each one, then manually connect their needs into each other's diagrams.

6.2 Guide Scope Focusing Hardware Design

The guiding scope will feature an automated focusing system. There are 4 physical components that will be needed in order to autofocus the guide scope. The first is a custom 3D printed part that will work as the means of rotating the focusing mechanism on the scope. The second key component is a custom gear attachment for the motor which will be used to allow the motor to interact with the custom 3D printed mechanism. The next is the baltimov's mask which will work to produce the input needed in order to determine the focus. The final physical component is the motor (insert actual motor) which serves as the main electrical component driving change within this system. These 4 components will all must work together in order to properly change the focus.

The 3D printed mechanism plays a key role in allowing us to interact with the guide scope. The mechanical nature of the focusing mechanism on the scope will require us to create a fixture to rotate it in order to adjust the focus. The approach being used is a tension based system that pushes against the guide scopes focusing mechanism and will be driven by a gear adapter on the motor in order to rotate the piece. The design below is the first prototype of the system designed to move the piece

Focusing Motor Design

Figure 32

The baltimov's mask is the third key component. The baltimov's mask relies on optical principles of light through a specific pattern in order to determine the focus of light passing through the sensor. The model is created with a thingiverse tool that requires specific dimensions from the guide scope to be provided. This fixture screws on the scope lens and allows imaging software to interpret the pattern as the light goes through the pattern. The image below is an example of the expected view when out of focus. The key points of interest within this image are the diffraction lines around the circular light source. If a vector is drawn to represent each diffraction line they should all intersect at the center point of the light source.

Out of Focus Example Bahtinov Mask Image

Figure 33

This image below displays what the camera would see when the lens is in focus. The same principle is followed as in the image above a set of vectors to represent each diffraction line would be drawn. In the image below it is clear that the lines all intersect at the midpoint of the light source.

Focused Example Bahtinov Mask Image

Figure 34

The motors will work through a gear adapter and will be driven by the embedded design system along with the other motors. The motors will have limitations to rotation as to not strain the focusing mechanism on the guide scope. The motors will choose how much to rotate the scope based on image processing done on the Raspberry Pi Compute Module 4. The motors will be communicating over I2C in order to create a feedback loop and only rotate after the next Image has been processed to avoid skipping the point of focusing.

The last physical component will be the Raspberry PI compute module 4. The CSI-2 port will be used to interface with the camera where an image will be derived. Then Python code will initiate the first image to be taken, interpret the image, and relay one of two possibilities: the scope is in focus or it is not. Then it will transmit this information to the microcontroller over I2C.

6.3 Embedded System Design

Tracking, Guiding, Machine Learning, Camera Interfacing were all accomplished on top of an Embedded System design. Through research we determined the best approach was to leverage the capabilities described in Section 3.8 of the document which features the Raspberry PI Compute Module 4S pro's and con's. Its benefits over a microcontroller or standard microprocessor approach allowed us to keep the entire processing and control on one embedded system while our motor control for the stretch board could have been handled and interfaced by adding a microcontroller specifically for motor control. We interfaced the ZWO camera to the Raspberry PI Compute Module 4 via usb which allowed us to leverage the existing INDI driver in order to both automatically take an image as well as allow us to then transfer it onto the boards memory where it will be stored until spectrum analysis occurs. The tracking and guiding camera which is the Raspberry PI HQ camera is interfaced via the Camera-Serial Interface CSI-2 which enabled us to have the quick and efficient real time processing required to achieve accurate tracking and guiding. The machine learning task in order to detect and classify stars leveraged the python programming language available on the Raspberry PI operating system to run complex classification models that focus on accuracy while leveraging the design's speed and memory capabilities. The Raspberry PI HDMI port was used as a way to allow us to interface a real display. This was a major design choice as we allowed the user to select any display of their choice when interacting with our product. We did not place any restrictions on what display device could be used.

The stretch goal was to allow for an additional microcontroller that can be used within the embedded design as a way of controlling the motors for the custom mount after the data is processed on the Raspberry PI compute module 4S.The stretch goal was designed such that we would be able to handle all of our vector calculations within a single device. The motor driver will be interfaced via INDI drivers as this is our main way of communicating with PHD2 which we use for tracking. This allowed for greater accuracy as motor control shields and methods

based on Raspberry PI architecture were slower and less accurate than microcontroller based motor control.

The design allowed for accuracy and our desired minimum exposure rate to be accomplished. The features available on the Raspberry PI allowed us to increase RAM up to 8GB which we leveraged to provide sufficient speeds and memory access rates for our image processing task. The multicore design also played a crucial role as we leveraged it to run multiple applications simultaneously. Execution time of a task is a critical feature as it dictated our accuracy and in the end we even determined we were close to being capped on processing power. Vector calculations on images in real time, image processing techniques, and Debayering all were major time intensive tasks we had to properly manage . This required leveraging all the features available within the Raspberry PI architecture. The standalone approach eliminates the need for external cloud computing and allows for an all-in-one product that does not use external processing for image processing or our temperature calculations.

The embedded system also possesses an additional FPC connector which provides power, ground, and a range of IO pins which was created to fit the needs of our stretch goals. This was a major discussion point as it allows the base hardware design to serve as a multifunctional approach given time would have allowed for more features to be implemented. With only about ⅓ of the pinout being used we mostly capped our selves on processing power. The hardware design, although simplistic, is feature rich and serves as the foundation for image capturing and processing of astral bodies.

6.4 Final PCB Design

The final iteration of our Main PCB included three power lines, one to the Raspberry Pi Compute Module 4, one to the peripherals (HDMI, USB's, etc) and one to the unrealized "Stretch Board" for Motor Control. The reason that we opted to use 3 different power lines for our board is by the recommendation of Dr. Arthur Weeks. From his experience the Raspberry Pi is extremely sensitive to fluctuations in power, and because of this we should keep its power line completely independent of all the others. We then split the "Stretch Board's" power from the rest of the board as the "stretch board" would need to drive the stepper motors to actually move the mount. This would likely cause a great shifting in the power line and thus it was separated.

Our final board had one HDMI port, four USB ports, one Mini USB port, two FPC ribbon cable ports, and three barrel jacks for power input. The power lines were smoothed using the techniques described in 6.1. And the necessary I/O for all the peripherals was mapped into the Raspberry Pi Compute Module 4 connectors.

When it came time to lay out the board, we opted to use a 2 layer design with wide traces on power lines to reduce cost. In the future, if we used a more highly layered design with diffusion layers to separate power lines, we could have greatly condensed the layout of the board, but as size was not of great concern, we did not do so.

Our first print of the board was mostly successful with two failures, we first accidentally printed male USB ports on the board instead of the necessary female USB ports, through some soldering on the back of the connectors with some female USB adapters, we were able to verify that the pinout of these connectors was correct. The second mistake that we made was the polarity of several data lines on the FPC connector to the Raspberry PI HQ Camera was invented from normal. Several lines had their data positive where data negative should go and vice versa. Though we did not know it at the time we also had the entirety of the pinout inverted from what it should be, but this was able to be remedied though some “manual manipulation” of the board later.

Attaching the Raspberry Pi Compute Module 4 mount was more difficult than expected. Because the pins were so dense and there were pins on both sides of the mount, when soldering paste was used there would be unfixable shorts on the inside of the connector. In order to remedy this, we traded a few unfixable shorts on the inside of the connector, for a larger amount of fixable shorts on the outside of the connector. When we placed the solder paste down on the print, we then placed them on the outer edge of the pad rather than down the center aisle of the pads. This allowed the pins to “suck” the paste into the pads soldering them, but when there was no connection, or the pins were shorted, the shorts were always on the outside of the connector and were thus able to be fixed with some flux and a soldering gun.

The second version of the board had three main improvements, it first corrected the printing of the male USB ports with female USB ports, it second fixed the polarity of the data lines on the FPC connector, and it finally added switches and LED Indicators to the power lines for quality of life. There also had to be some “manual manipulation” of the Raspberry Pi HQ Camera FPC Connector. As its pinout was accidentally inverted, we had to turn the connector around 180 degrees and solder it down in place on the pads, and secure it to the board using non-conductive super glue.

After these modifications, the Main PCB was fully functional.

6.5 Zemax Simulation

The spectrometer will be conceptualized using the Optic-Studio Zemax software to guarantee the system's functionality and provide a platform for evaluating its performance. Zemax enables us to fine-tune parameters to ensure the achievement of the targeted spectral resolution before the actual construction.

The simulation process will be conducted using the Non-Sequential Component Editor, which offers numerous advantages, particularly for designs necessitating three dimensions.[7]

The initial simulation phase involves creating a virtual telescope model and a light source representative of a star. Integrating as many relevant details as possible is integral to correctly simulating our system. A challenge arises in accurately replicating the wavefront emanating from the Schmidt-Cassegrain telescope since the front corrector plate is matched with the mirror, making the precise wavefront challenging to replicate in Zemax. Nevertheless, by employing an equivalent mirror and matching the focal ratio, an approximate wavefront can be generated, with the acknowledgment that additional spherical aberration—ordinarily compensated for by the telescope's corrector plate—will be present in the simulation.

To simulate a white light source, we use a Gaussian source replicating the spectral profile of a mercury vapor generic lamp. To validate the effectiveness of the simulated telescope and light source, a detector is placed within the simulation at a distance of 1500mm, corresponding to the telescope's focal length. To make sure our results match what we expected, a preliminary ray tracing exercise was performed to analyze the results.

Simulated Star Test Source. 1500mm focal length mirror with a gaussian source. Light propagating towards focus

Figure 35

This is a cross sectional view of the black line is a mirror with 1500mm focal length this is our simple simulation of the telescope. Light propagates first left then focuses from the mirror to the right.

Detector View of the star at Focus

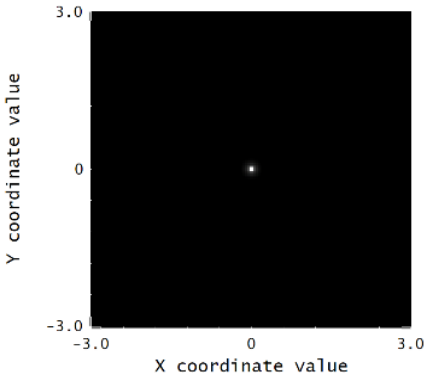


Figure 36

This is the view of the white detector rectangle in the next figure

6.6 Spectrometer

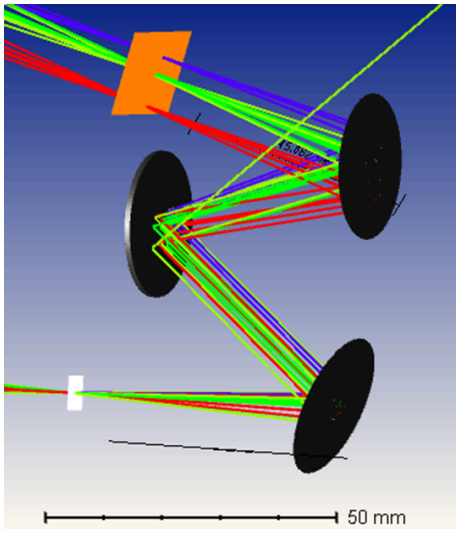


Figure 37

Light rays entering from the bottom left white rectangle then collimated by the first surface diffracted by the second surface then focused by the last surface onto the orange detector.

Exact Prescription Information for Spectrometer

Comment	Ref Object	Inside Of	X Position	Y Position	Z Position	Tilt About X	Tilt About Y	Tilt About Z	Material	Radius	Conic	Maximum Aper	Minimum Aper
Source sim	0	0	0.000	0.000	0.000	0.000	0.000	0.000	MIR...	3000.000	0.000	152.400	0.000
Gaussian...	0	0	0.000	0.000	20.000	180.000	0.000	0.000	-	20	1E+07	1.000	0
	0	0	0.000	0.000	1500.0...	0.000	0.000	0.000		3.000	3.000	150	150
Mirror 1	3	0	0.000	0.000	50.000	20.000	0.000	0.000	MIR...	-100.000	0.000	12.700	0.000
Mirror 2	3	0	0.000	42.000	55.000	0.000	0.000	0.000	MIR...	-100.000	0.000	12.700	0.000
Diffraction	3	0	0.000	30.000	15.000	180.000	0.000	0.000	MIR...	0.000	0.000	5.000	12.700
Camera	3	0	0.000	56.000	11.900	15.000	0.000	0.000		11.500	7.500	1500	1000

(in mm and degrees)

Figure 38

Output Spectrum for Mercury Vapor Lamp

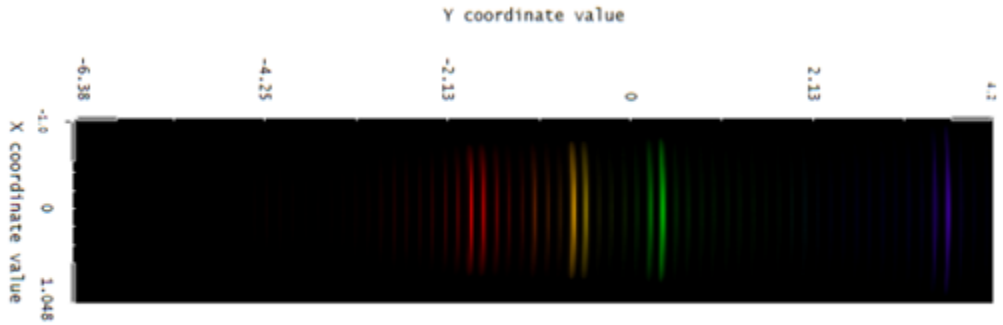


Figure 39

One feature present in our simulations is severe astigmatism. Some more advanced systems use the Czerny turner design and implement cylindrical lenses to reduce astigmatism; however, in our situation, we are using a modified design that does not use a slit. Because of that, we do not gain any performance in the system by removing this aberration. We cannot use a slit because our targets are stars, and the image created by the telescope will be less than 4 arc seconds. Keeping a star steady in a slit small enough for this angular size is nearly impossible without extremely advanced tracking, which we do not have any access to. However, since our product is a DIY system, the end user could opt to add a slit since our housing will support the addition of slits of varying sizes. It is best to leave this as an option for the end user since, in most cases, they will be unable to use the system should a slit be installed and non-removable.[38]

With a simulated model of our spectrometer, we can now begin to review the engineering requirement specifications for the components and verify that we have met our requirements to develop a working product. The table below includes each component and the engineering requirement specifications along with a measurable performance metric. With this table we have laid out a structured way to ensure we reach each goal and objective for spectrometer performance.

Component Engineering Requirement Specifications and Measurable Performance Metric

Component	Requirement	Performance Metric
Diffraction Grating	**600 lines/mm** Must provide enough angular separation of wavelengths to resolve absorption lines from stellar objects	Must be able to differentiate less than 1-nm separation in wavelengths
Mirrors	**450nm-750nm** Dielectric coating should support and reflect all wavelengths between the desired range	Reflect spectral range with 90%> efficiency. And produce an image of the spectrum on the camera sensor.
Camera	**24.1 Megapixels** support enough pixels to resolve absorption lines. Have a sufficiently large detector area. sensitive to wavelengths from 450 nm-750 nm.	Produce an image of the spectrum meeting the previous two metrics.

Telescope	**1500mm focal length** at f/10 must have a sufficient focal length to isolate stars.	Spectrum not contaminated with the spectrum of surrounding stars.

Table 14

Each component meets the following requirements, and we can determine each performance metric by checking the simulation results in the figures above. Now, we should review the simulated performance of our system and determine if we have sufficient spectral resolution to meet the engineering specification of performance set earlier. We must resolve two wavelengths separated by no more than 1 nm to meet our resolution requirement. the following figure is a simulation of the three wavelengths 550 nm 551 nm and 560 nm.

Test output for determining minimum wavelength separation ~1nm

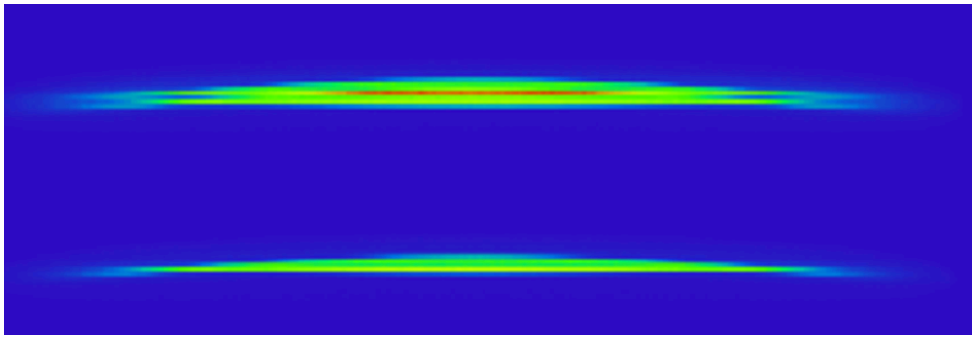


Figure 40

Our spectrometer barely enables us to resolve wavelengths separated by one nm. It does not meet our previous engineering requirement, which would require us to resolve a 0.5 nm difference at 500 nm; however, based on the output of the mercury vapor lamp, we can easily identify absorption lines, making me believe that our earlier requirement was set to a high for a budget stellar spectrograph. to increase the resolution of our system to the requirement set earlier we would need to use a more expensive diffraction grating. Right now, as the system stands, it will cost around $180, which is cheaper than even the cheapest transmission grating filters on the market, and we easily outperform. I consider the price-to-resolution tradeoff to be worth it in this case.

6.7 Design of the Encasement

The encasement was designed using Fusion 360 and built with modularity in mind. For the model that we used in this project we designed it for parts with a width of 12.7mm. And in particular used the parts tabulated above in the components section. Our product is being marketed to a STEM audience and will require some assembly. Our design aims to meet any customer's needs as cheaply as possible. The current design provides three main modular parts. First, our design can support any camera mount. Since the product requires assembly and 3D printing, the customer can simply add their camera mount to the spectrometer. The only constraint is that the camera sensor must be larger than 15mm x 20mm, ensuring you can recover the visible spectrum in one photo. The lens mount is also adjustable. To keep things simple, it will come with a standard T-mount, which most telescopes have an adapter for, but just as with the camera mount, this can easily be switched out. The last modular part of the encasement design is the slit. For customers lacking professional tracking and guiding, we have left the slit as an option. Stars are small enough that spectra can be resolved using only the star's image. However, if you have professional tracking and guiding hardware, a slit can be added to increase the resolution of the spectrometer further.

We have integrated a few features into the encasement to make things easier. The entire top of the enclosure detaches for ease of assembly, providing easy access to where mirrors and gratings need to be placed. The center points where each piece has to be placed have been precisely marked, so you only need to place your parts in the marked areas using double-sided tape. Since 3D printers are not created equal, I designed the spectrometer to use as few precise angles as possible. This way, even a cheap 3D printer can be used to manufacture the spectrometer. No steep angles are used, and multiple surfaces are at perfect 90-degree angles, making it easy on most printers. Overhangs were also avoided so various filaments could be used without worrying about sagging parts.

Spectrometer Prototype 1

Figure 41

With everything assembled, the Spectrometer should be rather small, about 4 inches by 3 inches. The Spectrometer must remain small to avoid crashing into the mount as the right ascension of the tracking mount changes, and anything sticking off the back of the telescope is liable to hit the mount if enough attachments are present. For this reason I designed the Spectrometer to have the light enter and exit from the same side. This allows us to put our camera on the interior side of the telescope, limiting the amount of space the full assembly takes up.

Verison two will optimize the print to use less filament to reduce print cost and integrate a new way to position the mirrors. This way we can make sure that they are accurately placed each time. This first version was very useful for prototyping and resulted in producing clean spectra but to further the accuracy of the system we used a different mounting system. Below is an image of the spectrum from prototype one. Note since we are not using a star source a slit is added to show emission lines.

Spectrum from version 1

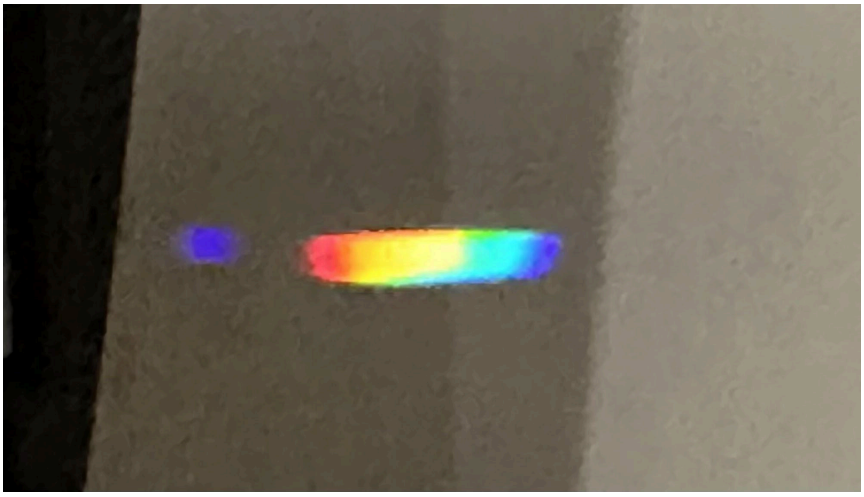


Figure 42

This image was taken with an iphone 14 pro camera and cropped in significantly. It is a rather muddy spectrum but this version of the spectrometer is held together with tape so the alignment of the mirrors and diffraction grating were off. There is also no camera mount in this version so we are just imaging the spectrum on a cardstock screen. However with that said you can see the spectral profile of the flash light. It appears that there is a separate led in the deep blue used to white balance the flashlight on the phone.

Final Spectrometer version

The final spectrometer design uses the same mirrors and diffraction grating as the original prototype but utilizes a different camera. Unfortunately, the DSLR camera body would not fit due to its dimensions and mounting point. The camera uses an EF mount, which has specific keying requiring that the camera be twisted into place. This is impossible with our design because the camera body is rectangular and crashes into the body of the spectrometer when you attempt to mount it. The ZWO 294MC pro is cylindrical and does not suffer from this fault, so we have opted to use this camera instead. The fully assembled spectrometer camera and telescope are seen below.

Spectrometer fully assembled and attached to the telescope

Figure 43

However, there is a small problem with this configuration. The ZWO 294MC pro is what is known as a true color camera. This means that the engineers designed it to model the human eye instead of having a flat spectral response. The following spectra are taken from the stars Arcturus and Rasalhauge.

Arcturus

Rasalhague

Figure 44

As you can see, in the case of the star Arcturus, almost no blue is detected since this star is relatively cold at around 4000k. There are very few high-energy photons. In contrast, the star Rasalhauge has much more greens and blues. This lack of sensitivity to the higher energy side of the spectrum biases our data significantly, preventing us from using Weins displacement law. However, we instead use the CIE 1931 color space, which also models the spectral response of the human eye to determine a correlated color temperature. With this color space, we can reweight the contributions of red, green, and blue and get an accurate temperature. In our testing, we are about 95% accurate. However, this

testing was rather limited given that we only had a few hours of clear sky over the entire time that the spectrometer and system were ready and the final demo day.

In closing the spectrometer system was a successful endeavor; we may not have been able to classify stars by the Morgan Keenan system but we did manage to classify stars by temperature.

Chapter 7 Software Design

7.1 Spectrum Analysis & Image Processing Software

The spectrum analysis and image processing posed many different challenges which we addressed over the course of this process. The issues are as follows: the light spectrum has no clear bounds that can be interpreted through software due to having missing portions it is unable to clearly detect where 450nm wavelengths spectrum begin and where the 750 nm wavelength ends. This was an issue as there were many different objects within our image ranging from multiple spectrums to different stars. There is no pixel to wavelength mapping so even with just analyzing RGB triplets we would be unable to match a color to a wavelength accurately. Next comes the star detection, in order to properly identify the spectrum we need to be able to locate a star however due to the spectrum looking for the brightest RGB value is not a reliable approach. Finally, once the spectrum is resolved an accurate mapping from 3D space to 2D space needs to be performed in order to create accurate graphs with simplified feature sets in order to have a machine learning model then handle the classification. This will take feature manipulation in order to have our machine learning models data closely match our and allow for more accurate classification.

The first issue we addressed was identifying the location of the star within our frame. The image detection of the star is quite precise as in order to solve the spectrum bounding we relied on fairly accurate star detection. The approach will be as follows: first we convert the image to grayscale; this allows us to narrow down the feature set of the star and allow for clearer bounding boxes to be created. Next, we leverage the photoutils library to allow us to identify the stars. The library works by identifying objects with the intensity over a certain threshold which we tuned based on our processed image and then based on their 2D gaussian shape. This worked to accurately produce the location of the stars within our image and then allowed us via our user interface to select the star of interest.

The next issue we solved is locating the spectrum that corresponds to our star. There are a multitude of issues with this making it a complex issue which requires a deeper understanding of the spectrometers output. The spectrometer returns only a spectrum with no other information to guide ourselves by which

possessed a problem. This means we only have a spectrum and noise within our image. The spectrum however includes both absorption lines which can be located anywhere within the image; it can also be included either at the start or end of the spectrum making it impossible to certainly define the spectrum based on its start and end color and their corresponding intensity to locate it. With the challenges discussed the best means of bounding the spectrum so we can then analyze it is as follows. We first make two assumptions, the first being that all stars are the same size with different intensities but the same size nonetheless just scaled either up or down uniformly. This allows us to find the center point of the star reliably which should be the midpoint between the diameter of the star. Once we have the middle of the star the following assumptions must be made. All stars are at such a constant distance (essentially infinity) that all star spectrums will be the same size. This means once we have the location of the center of the star given how the spectrometer works every star's spectrum will start at x pixels away from the star's center. Then based on the fact that every spectrum is the same, we are able to define the pixel range that contains the spectrum and begin analyzing the spectrum.

Once we are able to establish clear bounds on the spectrum of choice the next step is tackling the spectrum analysis. Spectrum analysis is broken down into a couple different parts. First we establish the theory to be used in order to calculate the temperature of the stars. The initial idea for determining color temperature was the using the standard black body curve approach and using wien's law once we had determined the wavelength which our peak value was present at. The issue with this approach is that our Intensity vs Wavelength graph was always biased towards the cooler side. This eliminated our ability to take wien's law as a viable solution and brought us to use the CIE 1931 color space x,y calculation method. We were able to first determine whether our star was on the colder side or warmer side based on whether the length of the red portion of our image was greater or less than the blue portion of our image. Once we determined colder or hotter we were able to bound the RGB triplets for either Green and Blue at 255 in the case of a warmer star and bind Red and Green in the case of a colder star. Then we took the relative proportion of the length of each of these different color segments and normalized them down to 255. The normalization was the last preprocessing technique needed to fit the data to be able to map it into the CIE 1931 color space.

Spectrum Analyses Block Diagram

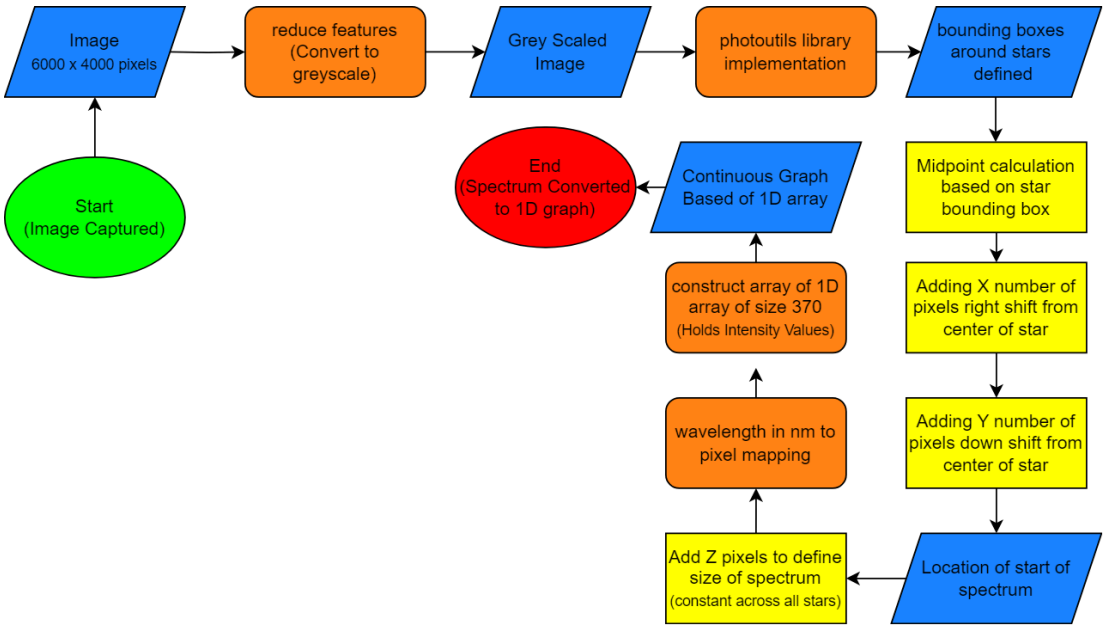


Figure 45

7.2 Image Processing

To capture and understand the images we are receiving from the telescope, an astrophotography software. These programs use image stacking software to pre-process and calibrate data from images. A few known image processing software include Adobe Photoshop, PixInsight, Siril, DeepSkyStacker, and Registax. Although each software uses similar pre-processing and post-processing steps as well as meets the same end goal, they each have their own qualities that help astrophotographers decide on the one that suits them.

The main factor in deciding which software to choose is the type of astrophotography being done as well as its ease of use. These types include, but are not limited to, Night Sky Photography, Piggyback Astrophotography, Prime Focus Astrophotography, Spectroscopy, Deep Sky Astrophotography, and more. When comparing the previously listed astrophotography software our decision was made based on cost, tools, complexity, and ease of use. Adobe Photoshop is great for those with a photography background. PixInsight was also a viable option for its more advanced features.[10]

Adobe Photoshop runs on Mac and PC, costs $21 a month, and is a manageable level of difficulty in terms of learning how to use the program. It contains many features that work with layers and layer control. Adobe Photoshop is also great for color enhancements and is user-friendly. The issue with Adobe Photoshop is

that it mainly works with images and is not designed specifically for astrophotography. Although it is useful, Adobe Photoshop does not provide enough tools that would benefit this project.

PixInsight runs on both PC and Mac, costs a flat fee of $275, and is more difficult to learn compared to the other astrophotography software. PixInsight is great for deep-sky astrophotography because this was the main reason the software was created. A couple of helpful processing tools it contains is the removal of gradients and star minimization. It is more advanced than most astrophotography software because of its high use of mathematical methods. These methods allow the program to preview non-linear adjustments, apply advanced processing scripts, and provide advanced background extraction as well as color calibration. The main reason we did not select PixInsight for this project was because of its price and complexity. PixInsight appears to be a very useful software for professional astrophotographers, but it is slightly out of the wheelhouse for college students.[10]

Unlike the other previously mentioned software, DeepSkyStacker runs only on Windows. DeepSkyStacker is also free and easier to use compared to Adobe Photoshop and PixInsight. It is mostly used for its ability to stack images and its post-process of astrophotography images. The software is great for noise reduction, image alignment and stacking, and is very easy to learn. DeepSkyStacker also has the ability to work with a variety of imaging formats such as RAW, TIFF, and FITS. One small issue with DeepSkyStacker is just how basic it is. Although it is user-friendly, its user interface does not catch the user's eye. The software also provides slightly worse results in terms of image quality.

Siril also runs on PC and Mac, is free, and is easy to learn. Siril has many beneficial features such as noise reduction, image alignment, stacking, photometric color calibration, and can run the third-party tool StarNet. StarNet is a program that allows you to remove stars from astrophotography images. The only issue with Siril is the amount of storage a file may swallow because of its stacking capabilities.[10]

Recently, artificial intelligence has also been used for more complex image processing. Because of the increase in volume of astronomical data, tools are being used and developed to withhold and create better images from the data. Artificial intelligence is primarily being used to combine images from different sources and detecting and classifying astral objects.

NASA's James Webb Space Telescope (JWST) is a prime example of a telescope system that is utilizing the rise of AI technology. The JWST now uses various imaging and spectroscopy instruments that cover several areas of the electromagnetic spectrum. Expanding the regions of the electromagnetic spectrum can provide more specific information from the images taken. The use of AI technology can create a high quality spatio-spectral image which would combine both high resolution and detailed spectra.[4]

Analyzing and identifying astral objects that may not be easily viewable is a very costly task. Being able to identify these objects requires higher complexity and is very time consuming. With AI technology, the cost of identifying these astral objects decreases significantly because astronomers can utilize computer vision techniques to classify the objects.

NASA is applying a system called Morpheus that can make use of AI technological methods. Morpheus is a deep learning algorithm that analyzes astral images pixel by pixel. It is able to classify galaxies using data based on morphological criteria. Although AI is still new in the world of astronomy, it will be extremely useful in the near future to identify objects, lower complexity, and decrease consumption time.[4]

To achieve image processing, we wanted to look at the astrophotography software Siril because of its cost efficiency and its ease of use. Pre-processing is done in a much easier way because of the scripts written within the software. The tools Siril provides were useful to us and what we would have liked to accomplish with our project. By primarily using the C and C++ programming languages, Siril captures and processes data from a telescope, and stacks the data to create images of the celestial bodies seen in outer space. Siril is “specially tailored for noise reduction and improving the signal noise ratio of an image from multiple captures.”[2]

The way Siril works is it takes in astrophotography data for pre-processing. During the pre-processing phase, the data is going under flat field correction as well as the correction of other basic imaging problems. The next step is stacking. Stacking is a technique used to increase the quality of an image by compiling multiple, roughly 10,000, layers of an image into a singular image. The process takes these images, aligns them over one another to reduce noise and increase the signal-to-noise ratio. Once stacking is complete, then begins the processing. During processing, the astrophotographer does whatever is needed to further enhance the image to their liking. Actions include but are not limited to color correction, noise reduction, and image sharpening.[18]

From this information, we hope to gather significant data from the spectra and celestial objects we encounter using the telescope. This data includes star classification, temperature of the object, as well as the object’s mass.

Siril is being used by our group because of its many advantageous and effective features. For example, Siril’s calibration allows for quality image enhancement. Siril can utilize files for BIASES, DARKS, and FLATS. Siril is also great with analysis in that it can recognize and process the scientific values found within astronomical images. Siril is equipped to provide the necessary information we require from objects in space such as star brightness. It can also provide astrometry interactions between various astral objects within the image.[35]

The program contains several pre-processing and processing algorithms designed specifically for astronomical imaging. Siril is great at processing deep-sky lucky imaging data because it supports a video file format known as “Sequence Extracted from a RAW” (SER) videos. All these features allow for multiple image processing from astral objects seen through the telescope. This is all done with a good storage capacity as well as an admirable processing speed.[11]

7.3 Guide Camera Control Software

Tracking and guiding required user input in order to properly calibrate the required initial parameters needed in order to attain the correct camera settings. This is something that is done via live imaging from a camera and allows for tweaking of camera related settings, initial alignment and camera operation controls in order to begin imaging. The guide camera relies on the Picamera2 Python library along with the PHD2 INDI driver tools.

The pyCamera INDI driver is our primary way of interfacing with the Raspberry Pi HQ Camera. This driver is a raspberry pi default driver that allows a diverse toolset in order to operate the camera. We used this as a means of capturing images, controlling camera settings and binding these tools to our GUI interface. In order to establish user controls are using numerous built in INDI driver tools. The main workflow is described as follows. The part of the user interface will be a button that will prompt the user to initiate the guiding subsystem. This will lead to opening PHD2 and allowing for the guiding tracking live view and allow for modification of the system settings via 2 primary settings that can be altered. The first is the analogue gain; this value will range from 1-99. The next slider is for exposure time this is calculated in microseconds on the PHD2 interface but for facility are instead chose to scroll through them as the standards in the industry indicate which are via fractions of seconds up to whole seconds our bottom range will be 1/800 seconds and our top range for the guiding system will be 15 seconds. Once these settings are selected a third button with the option to save and activate guiding can be selected. This button will begin the linear right ascension and declination of the system and where the actual tracking and guiding will begin.

Our GUI works as the advanced GUI for the guiding scope this allows us to concentrate defining all user settings into one location. This integration allows us to fully control the HQ camera and use this interface before reaching PHD2 as a means of aligning the camera to the telescope as well as working to define focus. This will play a crucial role as the GUI offers multiple complex methods which will allow for an intuitive feel and allows us to add go-to features from our stretch goals inside a front end environment which will be easy to navigate and will allow us to build upon to implement the extensive feature set which we attempted to reach. The openCV library and the Siril api tools will also work alongside our GUI in order to create an all in one tool for both tracking and guiding and any aid in

image processing and temperature calculations. These core GUI features work as our foundation due to their rich feature set and customizability within the python environment which allowed us to push for a more complex feature set.

7.4 Guiding & Tracking Software

Tracking and Guiding is a major subsystem that requires accurate image detection and complex image processing in order to reach the desired error of only 3 arcminutes. The limitation of a smaller more lightway camera with a smaller telescope and less powerful lens makes it so that we had to do more complex image preprocessing before we could accurately determine the correct vector to move to. The process is broken down into 4 steps. Calibration,Image preprocessing,image detection, and then vector creation. The first is tracking the system will have to be calibrated via polar alignment this process will allow for the initial values required to be set up. This will be the primary step as allowed us to reach the correct tracking rate. Once the initial values are arranged we are able to continue onto the software guiding and tracking directions.

Python is the primary programming language for this subsystem. The OpenCV library is used to convert our image to grayscale; this allows us to reduce our feature set and allows for more accurate spectrum bounds to be created. OpenCV will then be leveraged in order to employ the ORB alignment algorithm that allows us to move both of the images into the same coordinate plane allowing for accurate vector calculations that will take into account multiple possible small errors previously discussed in the research section. This is the most important step as everything moving forward can only work if the stars are accurately shifted to their real relative positions from one another.

The system calibration establishes the baseline for the tracking system. The next step is based on accurately translating the movement of the earth to the motors. Due to the system design being focused around star tracking, our initial movement requires us to shift the right ascension continuously. This requires us to move the mount after the initial calibration at 15.04 arcseconds per to the right. In order to do this we leverage a motor controller included in our stretch board with partial steps in order to accurately move to the required distance based on motor distance mapping. The right ascension is the only thing the tracking will handle as the core idea of equatorial mounts is that after the object has been aligned only the right ascension must be altered.

Next we used the photoutils library which defined our stars bounding boxes and allowed for the system to detect the star of choice. The user interface will allow for selection of the target star of interest which is the primary source of observation and where our accuracy is measured in relation to. This step also focuses on matching the stars as once they are on the same coordinate plane we are able to accurately detect the distance between the two and use a mapping

conversion to translate this information to the motors.

In order to send information to the stepper motors, 4 DIN wires for each motor are needed, 2 for each of the coils, as the coils opposite each other replicate what the initial coil is doing in order to lock the motor in place. For this a series of signals must be sent out through an array specific to the motor in the code. This array is the sequence of on/off signals for each coil that would result in the internal gear turning, which is then geared down to the output physical interface.

7.5 User Interface Design

Our user interface was fully designed in Python. Using a multitude of extensions, our user interface was designed to accept actions and commands from buttons and textboxes for ease of use. But, this was only what was done to make the program look appealing to the eye. In the background, we imported libraries (more specifically the INDI library) that connected to all of the drivers used in our project.

An important piece of the software was initializing servers. Without the initialization of servers and virtual machines, we would not have been able to have an optimal system where anyone can use the user interface. Through bash script files, we were able to streamline the initialization of these servers which made the startup process for our program take place in under ten seconds as opposed to one minute.

With the help of the custom tkinter library, our user interface contained features that required and were granted a very modern appearance due to its vast customizability features. Not only did custom tkinter allow us to design the user interface to our liking, but we were able to utilize its capabilities to display live images of both of the cameras we were using. Custom tkinter was a great tool in our software development, and the library being a Python extension made understanding its operations much easier.

The most vital portion of the user interface included the communication to the INDI drivers. The INDI drivers allowed us to control both the ZWO camera, the Picamera, as well as the Celestion mount that moved our telescope. Without the connection to the drivers, the user interface would have proven worthless.

Connecting to the drivers of our devices, we were able to set up settings and specifications necessary to have the devices complete the tasks that we needed. The user interface is designed to communicate settings such as camera gain, exposure time, image formatting, where to save captured images, which directions the mount should move, and how fast the mount should move. There are countless other controls we could have implemented into the user interface but these features were the most important for this project.

Chapter 8: System Fabrication/Prototype Construction

8.1 Initial Guiding Images

Initial tests on the guiding and tracking system proved to be quite a success and thoroughly showed the embedded systems design ability to capture and analyze stars. The first test revolved around basic imaging and making sure we were able to accurately capture items and see them via the live view and interact with the images after saving them.

The first round of test was focused around detecting the perfect focal length of our camera after adapting the M-12 mount to the MC mount on the lens. This played a crucial role within our prototyping stage as without figuring this out it would be nearly impossible to make out clear images. To do this we designed the adapter in such a way where we had a maximum of 100mm of focal length to recreate and tuned down the focal length after extensive testing on each until we reached accurate and clear images. In the end we determined that a focal length of 51 mm is where we stood with an adapter of 30mm of height. These settings were crucial to determine before any astral images could be acquired. In the end this initial test proved successful and we were able to move on to the next step in our testing.

Next came our initial test to image stars and other astral bodies as a means of analyzing our relative positions in the sky and the ability to fit enough stars within a single frame as needed. We started with the basics, to further test if our focal length was accurate and make sure we had reached the ability to focus at infinity, the test subject was the moon and the goal was to make out the texture of the moon. The initial test gave some issues due to the lack of implementation of the stellaris GUI; navigating to even a clear object such as the moon can oftentimes prove challenging. However, we were able to navigate to it and from that capture our first astral body with appropriate focus and texture. The image below is a result of that image and shows the ability for the Raspberry PI HQ camera to capture the moon's texture.

First Astral Body Image From Guide Camera

Figure 46

The moon image confirms our ability to visualize larger astral bodies but now we had to make sure we had such resolution to accurately image the stars and other small astral bodies. This required some level of manual alignment as we had to pick out a region of notable stars which would work as the baseline of comparison. This was a challenging process for the HQ camera as it pushed both the live view features and the exposure time to the test but without clear ability to view stars this would not be a suitable approach. The initial star image took a substantial time to acquire focusing, accurate exposure time, and aiming the system all provided evident issues which the tracking and guiding mount will eventually solve. In the end however the HQ camera on the Raspberry PI compute module 4 was able to capture the Bode's Galaxy towards the middle of the frame with an extensive amount of other stars all within visible scope proving the Raspberry Pi HQ camera indeed does have the ability to accurately image astral bodies and stars.

The Image below shows what the Raspberry PI HQ camera was able to see with the 55mm lens. These results were based on 3 second exposure time and have little to no star trails, vivid lighting, and display a suitable amount of stars within a single frame. This frame is located around the bode galaxy and features multiple stars and nebulas of different light magnitudes which will serve to create guidelines of limiting magnitudes.

First Guide Camera & Guide Scope Image

Figure 47

The conclusion of this initial test was determining whether the popular plate solving applications were able to make out the objects within frame. The importance of this was in determining whether both the pixel size, focal length, and resolution were suitable in order to plate solve which will be a major point given a go-to system is implemented. The image below shows the results after plate solving with no additional preprocessing meaning image resolution and results can only be further improved.This test proves the Raspberry PI with the HQ camera is a suitable device in order to accurately track and guide our main telescope due to its ability to detect stars on relatively low exposure with little to no star trails and be plate solved in order to determine what astral bodies are within the current frame.

First Plate Solved Star Image Sufficient Resolution Reached

Figure 48

8.2 Bahtinov Mask Simulated Testing

The Bahtinov mask will be our primary input into the autofocus system. This mask is a custom made piece that relies on camera and lens specification in order to operate properly. The overall goal is to create diffraction lines which vectors intersect at our light source in order to determine optimal focus. Focus is a major issue within astral imaging as it significantly impacts image accuracy which will be needed for both spectrum processing and tracking and guiding. The first step is designing the appropriate mask which was modeled and printed and can be seen in the image below. The idea is the pattern creates diffraction lines which only intersect at the midpoint of the source when focusing is optimal.

First Test Bahtinov Mask

Figure 49

The initial test was done with a light source and a pin hole to attempt to accurately simulate the emission we might expect from a star. This is a rough estimate but is primarily being used in order to iteratively improve the mask and determine the best focusing software parameters. The initial test proved unsuccessful and required changing the mask in order to create cleared diffraction lines and limit the light source over saturating the image that would be processed. After multiple iterations however the Bahtinov mask was successful in providing accurate focusing and the Raspberry Pi High Quality Camera was able to detect the diffraction lines which are then being used in order to focus our system.

The images below show the iterative steps used in order to focus the lens based on the image derived using a Bahtinov mask. The most important features can be seen on the diffraction lines created by the mask. Too far to either side will distort the lines completely and as we approach perfect focus we see diffraction lines begin to converge at the center of the light source.

Focusing Example on Bahtinov Mask (Out of focus -> Focused)

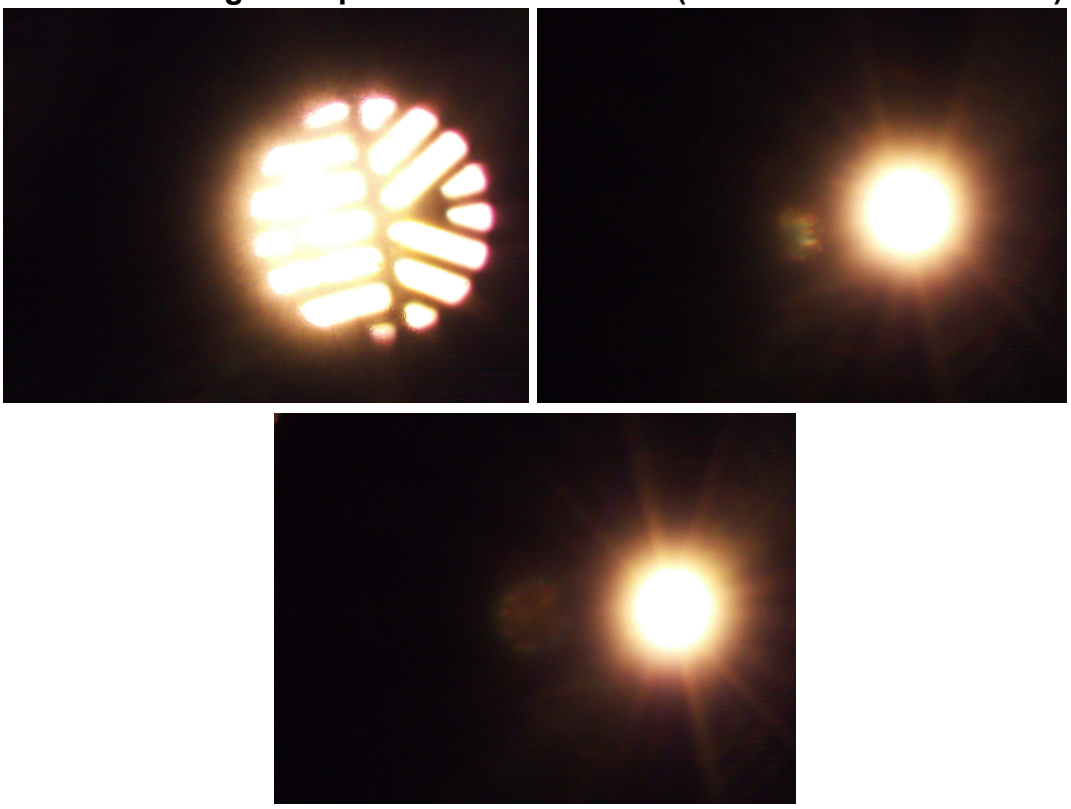


Figure 50

8.3 Encasement Prototyping

We use the 3D printed design from the above figure to properly prototype our system. By removing one side of our spectrometer, we can ensure that the light is traveling through the system correctly. In this first design, however, we opted to tape the parts in the correct location. For an initial prototype, our system worked and generated a spectrum from our test source, which for now is just a cell phone flashlight. We quickly realized that to have our system function correctly, we needed to design new mounts for the mirrors and diffraction grating so that everything lined up well and we did not risk damaging the optics during assembly and use. However, despite the issues with the first print, we recovered a spectrum nonetheless. Next, we need to design proper lens mounts and test actual stars.

8.4 Mirror Mounts

The mirrors we are using are first surface mirrors, so it is important that our mounting method does not touch the fragile front coating. The best solution to this problem is to create our own edge that can be handled. To do this, we 3D printed a ring roughly the same size as the mirror, 12.7mm in diameter, and tightened it to the mirror using a screw. This will give us an edge that we can use to mount things. Then, in the 3D print, we added slots so that each component

can be slotted into a precise location, ensuring accuracy. The diffraction grating is square, but we use the same method with just a 12.7mm by 12.7mm square. This design will make it easier to assemble the spectrometer and reduce the chance that the optics in the system are damaged.

8.5 Test Spectrum

Once the final system is assembled, we need to take some test spectrum to calibrate our code. In the lab, we use various light sources since they contain absorption lines that we can use to ensure our system is working. After testing various light sources, we switch to a tungsten lamp to mimic blackbody radiation. If we can get a reasonable estimate of temperature, then we can move on to test stars. With this structured approach, we can identify various problems. With the gas sources, we can make sure that the resolution of the system is sufficient; otherwise, absorption lines will not be resolvable. When testing the tungsten filament lamp, we can ensure we are capturing a wide enough spectral range. If we are, we should be able to perform a curve fit of the spectrum so that we can determine approximate temperature. Lastly, with both of those metrics verified, we can move on to capturing stellar spectra, where we can test more advanced aspects, such as distinguishing a combination of spectra due to a binary star system.

8.6 Setup Procedure

To get the best performance out of the stellar spectrometer there must be great care take while doing the setup. There are 3 major parts that need to be considered. First is telescope setup, second is mount and balance, third is initial calibration, and lastly is tracking setup. In this section we dive in depth and end with an easy checklist you can follow to ensure best performance with metrics to check for accuracy.

8.7 Telescope Setup

First, we want to place our telescope on the mount. Make sure to attach it securely using whatever means the mount supports it. Next, we moved to the guide scope. A small rail on most telescopes is designed to sit any components that need piggyback on it. Attach the guide scope to the telescope at this mount point, followed by the accompanying camera. Be careful to route the wires in such a way that they are never under tension. If the mount is impeded in any way, the alignment can be ruined. Next, we recommend a dew shield, especially for more humid areas. Keeping the ambient temperature of the front corrector plate away from the dew point is necessary to prevent condensation buildup and reduce any turbulence in the air directly in front of the telescope. Lastly, we balance the telescope, unlock the right ascension, and be careful, as the telescope will be free to move. If you are not careful, you could damage the telescope. With the right ascension unlocked, counterweights are added so that

the center of mass is directly over the axis of rotation. Adjust the balance once the telescope is balanced so the telescope side is slightly heavier. This slight imbalance reduces backlash and slightly improves tracking accuracy. Lastly, we balance the declination. Unlike the right ascension, which requires a slight imbalance, the declination should be perfectly neutral. In an ideal situation, the declination should not change when locked onto an object. By the end of the first step, the telescope should be set up in such a way that all attachments that go to the optical tube assembly are attached, and the setup should be balanced.

8.8 Initial Calibration

For initial calibration, we need to do a few things. First, we calibrate the guide scope so that both the telescope and the guide scope view the same objects. Then, we level the telescope and perform a polar alignment procedure. To calibrate the guide scope first use the Bahtinov mask and a star to ensure perfect focus. Then point the main telescope at a far-off object like a distant tree or a radio tower. Then, adjust the guide scope so the same object is centered. This way, the telescope and guide scope look in the same place, but the guide scope has a much wider field of view. Lastly, we need to polar align the telescope. Polar alignment is critical because it aligns our telescope with the same axis on which the earth rotates. This step is crucial because we want to limit changes in declination and use only one axis to keep objects in the frame. Carefully point the telescope so that the right ascension axis is pointed north. Then, align Polaris with the polar scope. If your alignment is good, the angle of your telescope should match the latitude of your current position. Lastly, the telescope is leveled using a bubble level, and then polar alignment is double-checked. Should this step be adequately completed, your telescope will face true north and be perfectly level. Moreover, objects visible in the telescope should also be visible in the guide scope anywhere you look.

8.9 Tracking Setup

Lastly, we need to set up the tracking. This step is done by entering the following information into the user interface and knowing enough astronomy to locate a few calibration stars in the night sky. With polar alignment, the telescope is set to counteract the Earth's rotation, but we have yet to determine where we are on Earth. All a polar alignment tells us is the direction of the Earth's pole and confirms if we are in the north or south hemisphere. To begin, enter the time, date, and GPS location of the telescope. This will give the telescope a general idea of what the sky looks like and what is visible. Then, we pick from a database of visible stars to start the two-star alignment. It is best practice to choose stars far apart from each other. Look in the sky for a bright star you recognize, such as Sirius. Enter that you want to use Sirius as your first star. The telescope will attempt to slew to the position of Sirius; however, it will fail to do so unless your polar alignment is perfect. Next, find the star Sirius in the guide scope, then center it with the telescope. Confirm that Sirius is centered, and then find another star. After a one-star alignment, the telescope should be able to slew to the

second star but not center it within the camera. After centering the second star, the alignment should be set up. Assuming perfect alignment, an object in the telescope's frame should remain still for the entire night, never drifting. Unfortunately, this is never truly the case. So, when making observations, the guide scope will be used to lock onto stars and fine-tune the positioning to keep everything centered.

8.10 Astronomy Resources and Quick Tutorial on Coordinate System

To operate this system the customer will have to have some knowledge of the right ascension and declination coordinate system. In this section I will provide some resources and give a quick tutorial on the cord system. Then we go over a very useful astronomy tool.

Astronomical coordinates use two values: right ascension and declination. Right ascension is normally written in hours minutes and seconds and declination in degrees minutes seconds. These cords define the direction you are looking in a polar coordinate with its apex at the north star polaris. With this coordinate system you can locate objects in the night sky. Here is a visualization of the grid in the night sky from stellarium.[39]

Visualization of the Right ascension and Declination lines

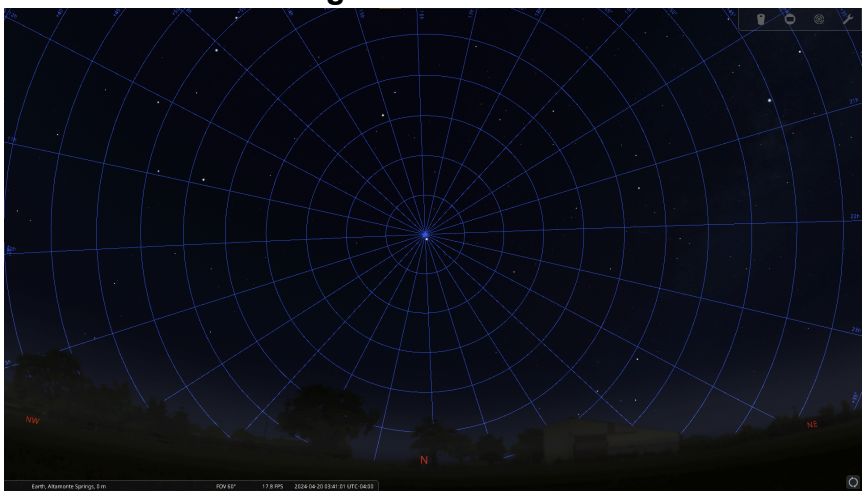


Figure 51

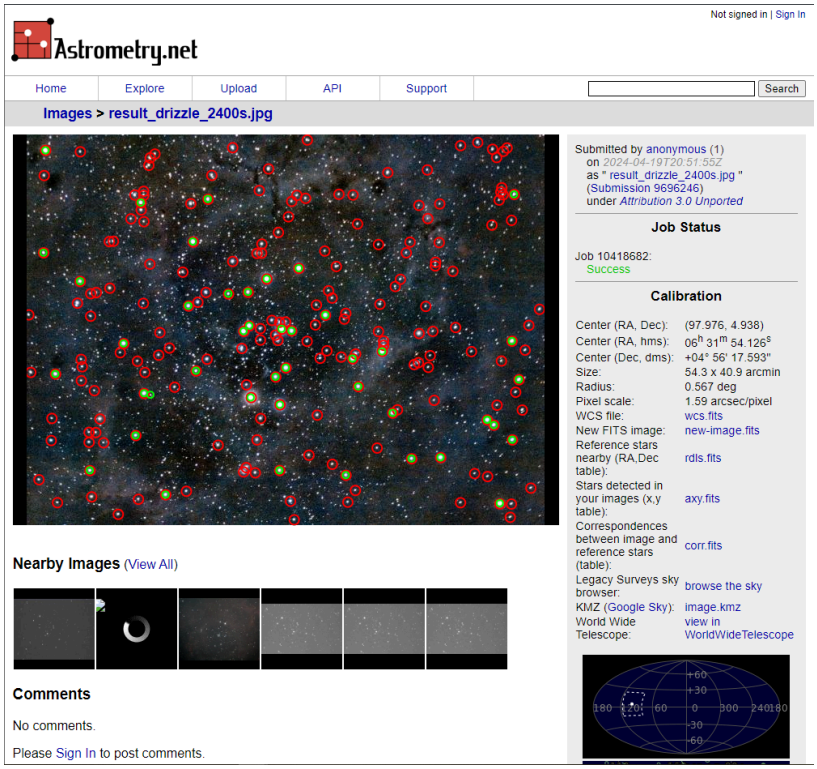
The lines coming out from the apex in this image represent the right ascension and the concentric circles the declination. With this coordinate system everything can be precisely notated in the night sky.

Astrometry.net

Astrometry is a great website for all of your astrometry needs. You can upload a photo to their system and get a read out of all visible objects within the frame as well as the coordinates for your image. With this website we can be

sure that we are tracking accurately and our mount is reading out correct coordinates. This powerful tool can even provide you with a sky survey that overlays your image and shows you exactly where it is with an interactive map. Here is an example of one of my image plate solved, and some of the information the website can provide me. In the image below each circle represents an object that was identified. And we can also see that the image was taken at a right ascension of 6 hours and 31 minutes and a declination of 4 degrees and 56 minutes. With a resolution of 1.59 arcsec/pixel. And a total angular window of 54.3 x 40.9 arcmin. This free software is incredibly useful for determining various aspects of your image. One of the main features of this program is that it requires no more information than your photograph. Other services that do plate solving such as Siril usually require focal length and pixel size.[9]

Astrometry.net analysis of the rosette nebula.



Rosette nebula by Addison Long

Figure 52

Along with all the previous mentions this service also has an API so that we are able to interface with our system. This can be used in conjunction with the guide scope to check cords and perform other astrometry operations. Refer to the section on Astrometry in the software part of the report to learn exactly how this system is integrated.

8.11 Checklist for Proper Calibration of STSC-MS

Task	Parts Involved	Indication Task is Complete
Telescope accessories attached	Guide scope Dew shield Cameras Spectrometer	All attachments going on the telescope should be present.
Route Wires	Camera wires and power cords	No cables should be under tension across the entire range of right ascension.
Balance the telescope.	Telescope mount	Right ascension should be slightly heavy on the telescope end. Declination should be neutral and not tilt when released.
Guide scope calibration	Telescope and guidescope	The telescope and guide scope should have the same object visible and centered in frame.
Polar alignment	Telescope mount	With proper polar alignment the telescope's right ascension axis should point true north and the angle of tilt to point at polaris should match your latitude.
Leveling	Telescope mount	Bubble level should read zero
Enter time date and gps location	Mount controller	Confirm time date and gps coordinates with the mount controller.
Perform two star alignment	Everything	Follow the instructions on the mount controller to complete two star alignment if everything is done stars should appear perfectly round even with exposures greater than 60 seconds.

Table 15

If you follow this checklist then the system should be performing perfectly and ready to take accurate spectroscopic measurements.

8.12 Camera Settings

After everything is set, it is time to start collecting measurements. Setting the camera up is hugely important. There are just two settings that are extremely important and frequently set incorrectly: exposure time and ISO/gain. First, we start with ISO/gain; at nighttime, many are tempted to set the ISO on the camera to the highest possible settings, but this is incorrect and often leads to garbage data. At high ISO, any signal received is significantly amplified, which makes noise begin to overtake the image. At first glance, a high iso will return a very bright spectrum, but fine lines will be difficult to differentiate due to the random noise. Every camera is different, but a general rule of thumb is to set the ISO to 1600. If your camera requires a gain number, your best bet is to take a few test photos. Ideally, in each individual frame, you should only see a few bright stars and a pitch-black background. If the stars appear large and the background looks more gray than black, your iso is to high. We want to be sure that we only amplify light and not any dark noise. Next, we need to set the exposure time. A few things limit your exposure time. First, the tracking and guiding accuracy of your mount. Should you have poor tracking and guiding, take test exposures until you begin to see star trailing once your stars trail back off a few seconds. Take ten test photos, and if 5 of the ten have no stretched stars, you have found your maximum exposure time. If tracking and guiding are not your limiting factor, we have to play a balancing act. Both ISO and exposure length contribute to the brightness of your photo. If you can expose the sensor for 10 minutes, iso 1600 will be way too much gain. So then you lower the gain to about 800 iso and take another test frame. Repeat this process until you find your star trailing limit or you reach the system's base gain, which should be specified by the camera you are using. With perfect settings, your image will look darker than you may expect. The data is still there; however, you must build it up over multiple exposures. This approach requires photo stacking but will give the highest dynamic range and reduce overall random photon noise as the amount of photons captured is large. Below is a standard single exposure of the m3 globular cluster. I chose to use m3 as an example since it is full of fine detail, similar to the fine detail of the spectrum we generated. With this exposure, you can see the background is a flat black, and only the very brightest of features are visible. Stars also appear as tiny pinpoints and only show very minimal stretching. After photo stacking and image processing are complete, many images will be combined, resulting in a bright image with a high dynamic range. As the comparison below shows, the stacked frame brings out lots of detail, and many previously invisible stars are now visible. By tailoring our exposures for this approach, we can pull more detail out of our spectrum than the competitors.[38]

M3 globular cluster single frame and stacked frame examples

Figure 53

Chapter 9: Administrative Content

9.1 Cost & Budget

Component	Quantity	Unit cost	Total
Orion EQ-13	1	$179.99	$179.99
Low power MC	2	$30	$60.00
Motors	2	$100	$200.00
RaspberryPi Compute Module 4	1	$80	$80.00
Reflective Diffraction Grating	1	$80	$80.00
First surface concave mirrors	2	$30	$60.00
Raspberry Pi HQ Camera	1	$45.00	$45.00
Total Cost			$690

Table 16

9.2 Milestones

As the Senior Design lectures continue, there may be adjustments to these milestones if and as additional deadlines and requirements are announced.

Task	Start date	End date	Duration
Brainstorming	January 11th	January 14th	3 days
Division of work	January 18th	January 18th	0 days
Research	January 18th	Ongoing	
Conceptual test with transmissive grating	January 28th	January 28th	0 day
Divide and conquer	January 18th	February 2nd	15 days
Acquire Raspberry pi and connect with camera	February 2nd	February 8th	6 days
Website	February 2nd	February 10th	8 days
Initial Motor Test	February 4th	February 10th	6 days
Spectrogram Initial Software Test	February 10th	February 20th	10 days
FPGA communication Research	February 20th	March 5th	15 days
60 page draft	January 18th	March 22nd	63 days
Initial Prototype Design	March 22nd	April 22nd	31 days
120 Page Paper	January 18th	April 23rd	120 days
Guiding and Tracking Prototype	May 1st	May 24th	24 days

Spectrograph Analysis Prototype	May 1st	May 24th	24 days
PCB Design	May 1st	May 24th	24 days
UI interface design	May 24th	June 24th	31 days
First PCB Order	May 24th	June 4th	11 days
Second PCB Order	June 4th	June 14th	11 days
Final PCB Order	June 14th	June 24th	11 days
Integration & Testing	June 24th	July 24th	30 days
Final Design Assembly	July 24th	May 24th	31 days

Table 17

9.3 Table of Work Distribution

Task	Primary	Secondary
Optical Design	Addison Long	N/A
Raspberry Pi Integration	Aaron Moreno	Sebastian Rowe
Spectrum Analyses	Aaron Moreno	Addison Long
MK Classification	Alejandro Olivo	Addison Long
User Interface Design	Alejandro Olivo	Aaron Moreno
MCU Integration	Sebastian Rowe	Alejandro Olivo
MCU Programming	Sebastian Rowe	Alejandro Olivo
Motor Control	Sebastian Rowe	Aaron Moreno
Power System	Sebastian Rowe	N/A

Table 18

Chapter 10: Conclusion

In summary and conclusion, STSC-MS is an endeavor comprising four interconnected systems geared towards capturing and analyzing stellar spectra using the Morgan Keenan classification system. This approach will then facilitate the determination of a star's position on the main sequence, offering insights into its relative mass, temperature, and elemental composition, while also enabling the identification of binary star systems.

To achieve these objectives, we are in active development of a suite of subsystems using specialized instruments: a spectrometer, tracking system, guiding system, and imaging system. The spectrometer, employing a modified Czerny turner design tailored for scotopic conditions, emphasizes the use of reflective surfaces to preserve signal fidelity and prevent spectral distortion caused by refractive optics.

The guiding system is engineered to swiftly resolve stars with short exposure times, ensuring precise tracking through continuous monitoring and adjustment of the target star's position. Equipped with a large aperture lens and a sensitive CMOS Raspberry Pi camera, supplemented by autofocus functionality, this system guarantees optimal focus and accuracy in image acquisition.

Meanwhile, our tracking system, meticulously calibrated for accuracy, and powered by the stellarium library, facilitates seamless navigation and spectrum acquisition of celestial objects of interest. Recognizing the critical importance of precise tracking in minimizing spectral distortion, we have chosen to develop many components of our tracking system ourselves, alongside the stellarium library.

Moreover, the software underpinning these systems is being crafted by our team, offering user-friendly interfaces accessible via an on-device display. To enhance adaptability and performance across diverse environments, we employ advanced image processing techniques like image stacking, which bolster dynamic range and signal-to-noise ratio albeit with increased processing time.

We are prioritizing simplicity and accessibility in design. Modular construction and 3D-printable parts engineered with broad tolerances ensure ease of assembly and customization, enabling amateur astronomers (like us) to tailor the system to our individual needs without necessitating high-quality 3D printing capabilities.

In actualizing the project, we first need to get all our components tested, including and especially the display, and the guiding motors. After that, Optics can focus on their filter design, Software can focus on tracking and GUI design, and hardware can focus on Power delivery and PCB design.

Preliminary testing of subsystems can and should be done as soon as possible to get rough accuracy. For example the guiding motor subsystem, attaching a laser pointer to the rig and then being able to have it rotate to point at specific points should be a good litmus test for accuracy. But even then there would be a significant difference between the accuracy of a laser pointer rotating around and the actual tracking of stars. The focusing motor could be tested alongside an algorithm to have a feedback loop from the image to rotate the focal length until the camera is in focus.

Also the tracking system could be tested by pointing a stationary camera into the night sky and getting positioning data from comparing that image against the stellarium database.

After these individual subsystems are functional, no time should be wasted in their integration. To actually test the accuracy of the motors, the robustness of the mask, the accuracy of the tracking, and the final output image, there is simply no substitute for actually going out into the field at night with an integrated prototype and capturing images of spectral phenomena.

Overall, from the work of our team Addison Long (PSE), Aaron Moreno (CPE), Sebastian Rowe (EE), and Alejandro Olivo (CPE) in these past several months we believe that through our consultation with one another meetings and in the writing of this document we have all that is needed. We have an achievable project. We have a concrete plan to make it happen. And we together have the skillset to actualize the STSC-MS.

Appendix A-Reference

[1] "1 new message." Accessed: Apr. 20, 2024. [Online]. Available: https://pwskills.com/blog/what-is-c/

[2] "About," Siril. Accessed: Apr. 20, 2024. [Online]. Available: https://siril.org/about/

[3] "Advantages and Disadvantages of Python – Make a Favorable Decision," Squareboat. Accessed: Apr. 20, 2024. [Online]. Available: https://squareboat.com/blog/advantages-and-disadvantages-of-python

[4] "Algorithms for analyzing the images from the James Webb telescope - Hello Future Orange," Hello Future. Accessed: Apr. 20, 2024. [Online]. Available: https://hellofuture.orange.com/en/ai-is-facilitating-the-exploitation-of-the-astronomical-amount-of-space-data/

[5] "Amazon.com: Motor Driver Board, 5 pcs/Set ULN2003 Driver Controller Board Stepping Module Electric Motor Control Module for 4 Phrase Stepper Motor : Industrial & Scientific." Accessed: Apr. 22, 2024. [Online]. Available: https://www.amazon.com/ULN2003-Controller-Stepping-Electric-Control/dp/B07P5C2KWX/ref=asc_df_B07P5C2KWX?tag=bngsmtphsnus-20&linkCode=df0&hvadid=80264480051516&hvnetw=s&hvqmt=e&hvbmt=be&hvdev=m&hvlocint=&hvlocphy=&hvtargid=pla-4583864003218970&psc=1

[6] "Amazon.com: STEPPERONLINE Nema 17 Bipolar Stepper 0.9deg(400 steps/rev) 1.68A 44Ncm(62.3oz.in) Motor : Industrial & Scientific." Accessed: Apr. 22, 2024. [Online]. Available: https://www.amazon.com/STEPPERONLINE-Bipolar-Stepper-0-9deg-62-3oz/dp/B00PNEQMLY

[7] "Ansys Zemax OpticStudio | Optical Design and Analysis Software." Accessed: Apr. 19, 2024. [Online]. Available: https://www.ansys.com/products/optics/ansys-zemax-opticstudio

[8] "Arduino Uno Rev3 — Arduino Official Store." Accessed: Apr. 22, 2024. [Online]. Available: https://store.arduino.cc/products/arduino-uno-rev3/

[9] "Astrometry.net." Accessed: Apr. 19, 2024. [Online]. Available: https://nova.astrometry.net/

[10] "Astrophotography Image Processing Software | Top Choices," AstroBackyard. Accessed: Apr. 20, 2024. [Online]. Available: https://astrobackyard.com/image-processing-software/

[11] "Astrophotography Image Stacking Software | Which One is Best?," AstroBackyard. Accessed: Apr. 20, 2024. [Online]. Available: https://astrobackyard.com/astrophotography-image-stacking-software/

[12] "ATMEGA328P." Accessed: Apr. 22, 2024. [Online]. Available: https://www.microchip.com/en-us/product/atmega328p

[13] "ATMEGA328PB." Accessed: Apr. 22, 2024. [Online]. Available: https://www.microchip.com/en-us/product/atmega328pb

[14] "ATMEGA328PB-AU," DigiKey Electronics. Accessed: Apr. 22, 2024. [Online]. Available: https://www.digikey.com/en/products/detail/microchip-technology/ATMEGA328PB-AU/5638812

[15] "Atomic Spectra Database," *NIST*, Jul. 2009, Accessed: Apr. 19, 2024. [Online]. Available: https://www.nist.gov/pml/atomic-spectra-database

[16] "Auto-Focus - Nighttime Imaging 'N' Astronomy." Accessed: Apr. 19, 2024. [Online]. Available: https://nighttime-imaging.eu/docs/master/site/advanced/autofocus/

[17] "Deep Learning: Classifying Astronomical Images — AstroML Interactive Book." Accessed: Apr. 20, 2024. [Online]. Available: https://www.astroml.org/astroML-notebooks/chapter9/astroml_chapter9_Deep_Learning_Classifying_Astronomical_Images.html

[18] "Definitions and workflow — Siril 1.2.0 documentation." Accessed: Apr. 20, 2024. [Online]. Available: https://siril.readthedocs.io/en/stable/Workflow.html

[19] "Everything You Wanted To Know About TensorFlow," Databricks. Accessed: Apr. 20, 2024. [Online]. Available: https://www.databricks.com/glossary/tensorflow-guide

[20] "Explained: Neural networks," MIT News | Massachusetts Institute of Technology. Accessed: Apr. 20, 2024. [Online]. Available: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[21]
E. D. Klenske, M. N. Zeilinger, B. Schölkopf, and P. Hennig, “Gaussian Process-Based Predictive Control for Periodic Error Correction,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, Art. no. 1, Jan. 2016, doi: 10.1109/TCST.2015.2420629.

[22]
“How to identify four-wire stepper motor coil pairs with a multimeter.” Accessed: Apr. 22, 2024. [Online]. Available: http://www.steppernews.com/2018/08/how-to-identify-four-wire-stepper-motor.html

[23]
“Interactive Atomic Spectra.” Accessed: Apr. 19, 2024. [Online]. Available: https://atomic-spectra.net/spectrum.php

[24]
“ISO Standard Clean Room Information.” Accessed: Apr. 19, 2024. [Online]. Available: https://www.clean-rooms.org/iso-cleanroom-standards/

[25]
“MIKROE-1530 Mikroe | Mouser.” Accessed: Apr. 22, 2024. [Online]. Available: https://www.mouser.com/ProductDetail/Mikroe/MIKROE-1530?qs=uPV7ZbtEjc6Q0KoZC6AETw%3D%3D

[26]
“OpenCV: Optical Flow.” Accessed: Apr. 22, 2024. [Online]. Available: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html

[27]
“Part 5: Spectral Resolution - B&W Tek.” Accessed: Apr. 19, 2024. [Online]. Available: https://bwtek.com/spectrometer-part-5-spectral-resolution/

[28]
J. Clark, “Photographing the Moon and Planets,” in *Viewing and Imaging the Solar System: A Guide for Amateur Astronomers*, J. Clark, Ed., New York, NY: Springer, 2015, pp. 57–80. doi: 10.1007/978-1-4614-5179-2_4.

[29]
“Photutils — photutils 1.12.0.” Accessed: Apr. 22, 2024. [Online]. Available: https://photutils.readthedocs.io/en/stable/index.html

[30]
“Python Language advantages and applications,” GeeksforGeeks. Accessed: Apr. 20, 2024. [Online]. Available: https://www.geeksforgeeks.org/python-language-advantages-applications/

[31]
R. P. Ltd, “Raspberry Pi,” Raspberry Pi. Accessed: Apr. 22, 2024. [Online]. Available: https://www.raspberrypi.com/

[32]
“Raspberry Pi Compute Module 4”.

[33] "Raspberry Pi Documentation - Camera." Accessed: Apr. 22, 2024. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html

[34] "SF2421-10B11 SANMOTION | Mouser." Accessed: Apr. 22, 2024. [Online]. Available: https://www.mouser.com/ProductDetail/SANMOTION/SF2421-10B11?qs=l4Gc20tDgJJRHcqrqEUN4Q%3D%3D

[35] "Siril." Accessed: Apr. 20, 2024. [Online]. Available: https://siril.org/

[36] "Spectrometers, monochrometers and spectrographs." Accessed: Apr. 19, 2024. [Online]. Available: https://www.horiba.com/int/scientific/technologies/spectrometers-and-monochromators/spectrometers-monochromators-and-spectrographs/

[37] D. R. Paschotta, "standard spectral lines." Accessed: Apr. 19, 2024. [Online]. Available: https://www.rp-photonics.com/standard_spectral_lines.html

[38] "Stellar Spectra." Accessed: Apr. 19, 2024. [Online]. Available: https://casswww.ucsd.edu/archive/public/tutorial/Stars.html

[39] G. Zotti, S. M. Hoffmann, A. Wolf, F. Chéreau, and G. Chéreau, "The Simulated Sky: Stellarium for Cultural Astronomy Research," *Journal of Skyscape Archaeology*, vol. 6, no. 2, pp. 221–258, 2019, doi: https://stellarium.org/.

[40] "The spectrography bookmark." Accessed: Apr. 19, 2024. [Online]. Available: http://www.astrosurf.com/aras/bookspec.htm

[41] "Thorlabs.com - Service and Support." Accessed: Apr. 19, 2024. [Online]. Available: https://www.thorlabs.com

[42] E. Technology, "Types of Motors - Classification of AC, DC & Special Motors," ELECTRICAL TECHNOLOGY. Accessed: Apr. 22, 2024. [Online]. Available: https://www.electricaltechnology.org/2021/01/types-of-electric-motors.html

[43] "Understanding Surface Quality Specifications." Accessed: Apr. 19, 2024. [Online]. Available: https://www.edmundoptics.com/knowledge-center/application-notes/lasers/understanding-surface-quality-specifications/

[44] "Understanding Surface Roughness." Accessed: Apr. 19, 2024. [Online]. Available: https://www.edmundoptics.com/knowledge-center/application-notes/optics/understanding-surface-roughness/

[45] "What Is C: Differences Vs C++, Advantages And Disadvantages." Accessed: Apr. 20, 2024. [Online]. Available: https://pwskills.com/blog/what-is-c/

[46] "What is PyTorch?," NVIDIA Data Science Glossary. Accessed: Apr. 20, 2024. [Online]. Available: https://www.nvidia.com/en-us/glossary/pytorch/

[47] "Why you should learn C," Educative. Accessed: Apr. 20, 2024. [Online]. Available: https://www.educative.io/blog/now-is-the-perfect-time-to-learn-c

Appendix B-Copyright Permissions

Appendix C-Software Code

Appendix D-Etc.

Was not able to generate citation based on metadata

[48] https://courses.cs.washington.edu/courses/cse576/06sp/notes/HarrisDetector.pdf

[49] https://www.digicamdb.com/specs/canon_eos-rebel-t7/