

Telescope Autofocuser

Camille Watson

Joseph Basdeo

Albin Myscich

Alonna Too-Chiobi

CONCEPT OF OPERATIONS

REVISION – Draft
8 February 2022

CONCEPT OF OPERATIONS

FOR Telescope Autofocuser

TEAM 6

APPROVED BY:

Project Leader _____ **Date** _____

Prof. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
Draft	2/8/22	Telescope Autofocuser Team		Draft Release

Table of Contents

1. Executive Summary	1
2. Introduction	2
2.1. Background	2
2.2. Overview	3
2.3. Referenced Documents and Standards	4
3. Operating Concept.....	5
3.1. Scope	5
3.2. Operational Description and Constraints	5
3.3. System Description.....	5
3.4. Modes of Operations.....	6
3.5. Users	6
3.6. Support	6
4. Scenario(s).....	7
4.2. Low Performance of the Human Eye.....	7
4.3. Offset Temperature Related Contractions.....	7
4.4. Introduction for Amateur Astrophotographers	7
5. Analysis	7
5.1. Summary of Proposed Improvements.....	7
5.2. Disadvantages and Limitations.....	8
5.3. Alternatives.....	8
5.4. Impact.....	8

List of Tables

Model Number	6F5N
Telescope Series	Apertura Newtonian
Focal Ratio	f/5
Optical Design	Newtonian Reflector
Telescope Aperture	152.4mm (6")
Telescope Mount Type	No Mount - OTA Only
Type of Electronics	OTA Only - No Electronics
Focal Length (mm)	762
Secondary Mirror Central Obstruction	2" Diameter
Focuser Style	Crayford/Crayford Style
Focuser Size	2"
Focuser Speed	Single Speed
Telescope OTA Length (in.)	27
Telescope OTA Weight (lb.)	10.2

Table 1

List of Figures

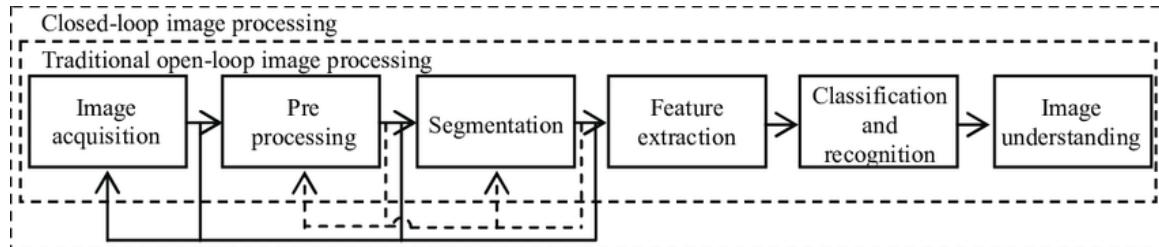


Figure 1

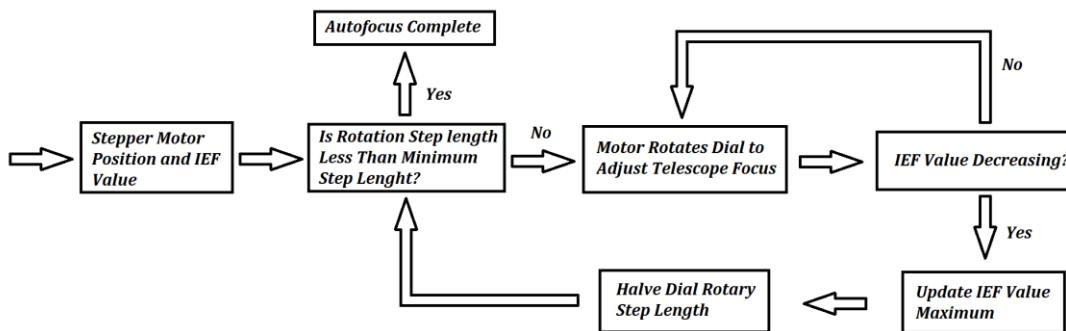


Figure 2

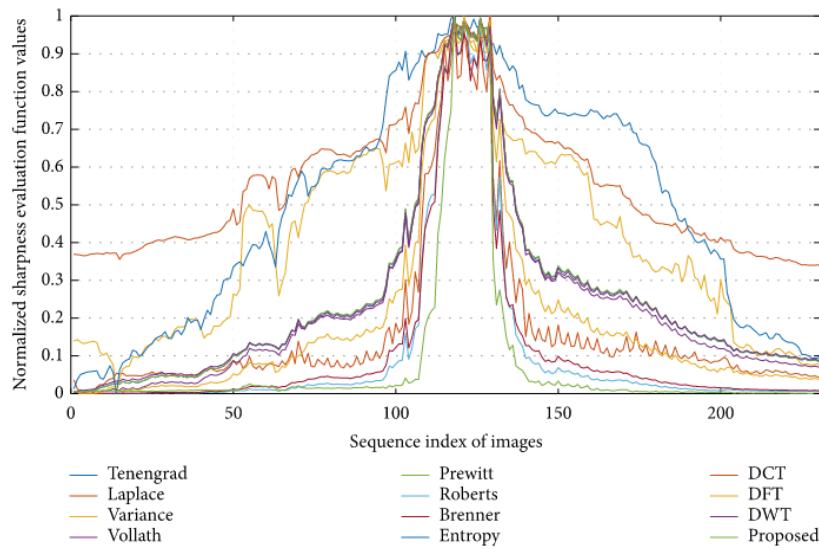


Figure 3

1. Executive Summary

Due to the rise of technological advances, the action of automating machinery has become a goal for most of our current technology. More specifically, astrophotography equipment, such as telescopes, have been enhanced to include autofocusing components, although most standard telescopes do not include this feature.

Our goal is to develop a telescope capable of automatically focusing on an object of interest. The Autofocusing Telescope will achieve this aim by utilizing a combination of motors, an astro analyzer, and an image capture/processor that will be able to focus, adjust, and capture a detailed image of the object of interest. The Telescope Autofocuser will save astronomers time by providing users with the ability to quickly/effortlessly obtain detailed images of their celestial target. To produce the best picture, we will improve the detail and increase light intensity of the image via image processing.

This device will reduce the astrophotography learning curve and encourage more individuals to embrace this hobby.

2. Introduction

In the sphere of astrophotography, observing natural phenomena are often recorded with digital images which are taken with an extreme zoom via telescope. A common issue centered around low brightness of an image is the result of telescopic zoom and the telescope's large focal ratios; therefore, it is challenging to adapt the focus to achieve sharper images. Observatories often have the advantage of advanced instruments to aid in mitigating this known problem with celestial imaging. However, hobbyists and members of the amateur astrophotography community frequently experience these problems and get low quality images as a result.

For the duration of the project, we plan to completely implement a coupled autofocusing module from which the telescope can achieve optimal sharpness when capturing an image. Since traditional methods of manual focus control offered by telescope producers aren't adequate for quality imaging, we plan to implement more intelligent imaging and focusing methods. The system consists of a pinion and rack focuser which is operated by a set of servos. Moreover, the image raster processing consists of intelligent real-time and post-processing models with the help of machine learning. Images are captured with a camera fastened to an eyepiece. To tie all components together, a cheap development board will be used to process images and optimize focus quality.

2.1. Background

Current telescope-based autofocus methods are split into two categories; passive and active autofocus.

The passive autofocus method is further partitioned into two separate categories known as contrast and phase detection. Phase detection autofocus can be examined where the received light ray is separated into two. Here, the hardware possesses micro lenses which contain separate image recorders; when the sensor finds that the image is focused, the recorder hardware will receive near-indistinguishable images. Alternatively, the technique involving contrast detection scans each pixel where the difference is assessed based on a consecutive set of images. As a result, the frame with the most contrast (most focused image) is algorithmically selected.

On the other hand, to employ an active autofocus technique, a measuring device is commonly used to determine range. These devices include but are not limited to infrared reflection and ultrasonic sound waves. Of course, each has their own set of benefits and limitations. In any case, the device determines the span between the target object and the affixed camera. Specifically, the algorithm calculating the distance will set the position of the lens. As a result, a focused image with a sharp contrast can be pipelined from the camera.

Imaging using high magnification and gradient-based contrast has an exceptional capacity for optimizing sharpness as opposed to algorithms dependent on edges, transforms, statistics, and correlation calculations. This concept explores gray coloration differences in adjacent pixels, which represent image sharpness. Examples of this raster image processing

include popular models such as the Hemli and Scherer's mean (HSM), Tenengrad (TGR), and Gaussian derivative (GDR) with the assistance of Sobel operators or Laplacian filters. Each of these methods make for great options for boosting the quality of gradient-based measurements.

2.2. Overview

The primary focus of the Telescope Autofocuser is to replace manually configuring the focal setting and image processing with an automated medium. As a preface, the project takes influence from most modern observatories that use a series of sophisticated techniques to optimize image quality. Despite the fact that Based on discussion and initial designs of the model, the apparatus is divided into three separate parts: eyepiece interface, dial manipulation, and image processing.

If a camera sensor is placed outside the focal plane of an ocular system, only a defocused picture with distorted boundaries can be recognized. Considering the frequency domain, high-frequency components of an image with distinctive shapes are filtered out as a result of the low-pass characteristics of the system's received frequencies. In order to take a clear image, the main notion of the autofocus equipment is to regulate the lens position in the system so that the camera is optimally positioned at the focal plane. From here, ample number of high-frequency elements are received to identify a definite contour of an image's boundaries. In most digital image processing techniques, a high-frequency filter is used to obtain the high-frequency components of the fast Fourier transform results of the image. If the scales of related high-frequency factors are larger than some present values, then the image is deemed as sufficiently sharp.

Based on the Apertura Newtonian 6F5N telescope system, it was investigated, as shown in Table 1. The basic system mainly consists of its parabolic primary mirror, white OTA, 2 inch Crayford-style focuser, f/5 focal ratio, and dovetail plate. It's important to note that the bare system contains no external electronics and must require specialized components to fit the design. External components will include, a PC management, stepper motors, breakout camera, and mechanical manipulation apparatus. To enhance the quality of captured images, a Schmidt corrector can be installed to the telescope front; a camera with a thin frequency band filter can be applied to receive images and convolute the sieved results to the PC management board. In this component, a stepper motor is employed to adapt the position of the main image of the telescope in the controller mechanism based on the image evaluation.

Here, the auto-focus procedure can be described as beginning with the camera which captures data concerning the tracked objects with the telescope. It then transmits the image to the PC management board. Then, dev board's software performs a calculation based on the image and directs instructions to rotate the stepper motor counterclockwise or clockwise. From here, a mechanical transformation unit will apply a rotational force into a driver that adapts the position of the telescope's main mirror forward or backward. As a result, it will successively adjust the imaging position and focus. The processes described are ultimately recurrent events until the camera recognizes the telescope's image plane. Therefore, a clear image is then obtained, which defines this system's schema as a closed-loop system.

Within the realm of software, the focus search algorithm, with the help of Hemli and Scherer's mean (HSM), Tenengrad (TGR), and Gaussian derivative (GDR) [Figure 3] are responsible for the focus quality and efficacy of the system. In the testing phase, the hill-climbing exploration approach can be used to accomplish a near optimal focusing effect [Figure 1, 2]. From here, the stepper motors will slowly rotate the telescope's knob at a suitable step length to adjust the system's focal length from which the image estimation function value is calculated to supply a constant stream of feedback. When the image estimation function begins to decrease, it implies that the optimal focusing location has been passed. As a direct response, the knob can be rotated a step in reverse. Next, the present step length is then cut in half, and the search procedure is recurred until the exploration is concluded with the lowest step size reached. As a result, the auto-focus procedure can be deemed complete.

2.3. Referenced Documents and Standards

1. Achim Mester, "Contrast Detection Autofocus (CDAF) for Telescopes using a Stepper Motor and a Raspberry Pi" July 6, 2014
2. Mir, H., Xu, P., Chen, R., & van Beek, P. (2015). An autofocus heuristic for digital cameras based on supervised machine learning. *Journal of Heuristics*, 21(5), 599–616. <https://doi.org/10.1007/s10732-015-9291-4>
3. Pech-Pacheco, J. L., Cristobal, G., Chamorro-Martinez, J., & Fernandez-Valdivia, J. (n.d.). Diatom autofocusing in Brightfield Microscopy: A comparative study. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. <https://doi.org/10.1109/icpr.2000.903548>
4. Vaquero, D., Gelfand, N., Tico, M., Pulli, K., & Turk, M. (2011). Generalized autofocus. *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. <https://doi.org/10.1109/wacv.2011.5711547>
5. Yao, Y., Abidi, B., Doggaz, N., & Abidi, M. (2006). Evaluation of sharpness measures and search algorithms for the auto focusing of high-magnification images. *SPIE Proceedings*. <https://doi.org/10.1117/12.664751>
6. Śliwiński, P., & Wachel, P. (2013). A simple model for on-sensor phase-detection autofocusing algorithm. *Journal of Computer and Communications*, 01(06), 11–17. <https://doi.org/10.4236/jcc.2013.16003>
7. Mateos-Pérez, J. M., Redondo, R., Nava, R., Valdiviezo, J. C., Cristóbal, G., Escalante-Ramírez, B., Ruiz-Serrano, M. J., Pascau, J., & Desco, M. (2012). Comparative evaluation of autofocus algorithms for a real-time system for automatic detection of mycobacterium tuberculosis. *Cytometry Part A*, 81A(3), 213–221. <https://doi.org/10.1002/cyto.a.22020>
8. Yan, X., Lei, J., & Zhao, Z. (2020). Multidirectional gradient neighbourhood-weighted image sharpness evaluation algorithm. *Mathematical Problems in Engineering*, 2020, 1–7. <https://doi.org/10.1155/2020/7864024>
9. Grigorescu, S. M., Ristić-Durrant, D., Vuppala, S. K., & Gräser, A. (2008). Closed-loop control in image processing for improvement of object recognition. *IFAC Proceedings Volumes*, 41(2), 5335–5340. <https://doi.org/10.3182/20080706-5-kr-1001.00899>

10. UNIVERSITATIS, FACTA & Ristic-Durrant, Danijela & Graeser, Axel. (2022). CLOSED-LOOP CONTROL OF SEGMENTED IMAGE QUALITY FOR IMPROVEMENT OF DIGITAL IMAGE PROCESSING UDC 62-52:004.932.

3. Operating Concept

3.1. Scope

The Telescope Autofocuser will increase the efficiency of stargazing by providing users with the ability to automatically focus on their star of interest and capture detailed images. Once the user has identified their star of interest, the telescope will then feed this image to an analyzer that communicates with motors to control the focus of the telescope. Once the analyzer has determined that the image is in focus, multiple images will then be captured and processed in order to produce a detailed picture of the star of interest. For demonstration and testing purposes, the telescope will be mounted in a room and a static image will be placed some distance away. While there are many telescopes on the market, the Telescope Autofocuser will replace the archaic action of having to manually focus your device. Not only will this benefit astronomers who don't have the time to manually adjust their telescope, but it will also benefit those who are beginners in stargazing.

3.2. Operational Description and Constraints

The Telescope Autofocuser is intended to be used by astronomers or those who have an interest in astronomy and would like to quickly/effortlessly obtain detailed images of their object of interest. An astro-analyzer will be attached to the telescope's eyepiece and a motor(s) will be attached to the telescope's focus knobs. Once the telescope has been directed towards the object of interest, the astro-analyzer will begin examining the clarity of the object. Depending on the clarity level, the analyzer will communicate with the motor(s) to adjust the focus of the telescope until the analyzer is satisfied with the object's clarity. Multiple snaps of this image will then be captured and sent to a program where it will be processed in order to improve the light intensity and detail of the object.

Based on this operational description, possible constraints include:

1. The motors must be mounted to the body of the telescope in a way that does not affect the telescope's vision.
2. The astro analyzer must be fastened to the eyepiece of the telescope so that it virtually prevents any movement and could block the user from peering into the telescope. Here, reliance on an external monitor might be necessary (assuming the project isn't headless).

3.3. System Description

1. Telescope: The telescope will be the body of the product as well as the subsystem responsible for identifying objects of interest. It will consist of the following components: the outer shell, eyepiece, lenses, mirrors, and a structural support. The telescope will provide our system with the ability to identify long range objects of interest to be focused on, captured, and processed. In order to prevent malperformance of the telescope we will avoid tampering with the inside.

2. Astro Camera (Analyzer): The astro analyzer will be responsible for determining if the object of interest is in focus. It will consist of a camera lens (responsible for perceiving the image) and an analyzer (responsible for determining the resolution of the image). This subsystem will give the product its main selling ability, which is auto focusing. To prevent faulty analysis, the lens of the analyzer will be cleaned routinely.
3. Motors: When the system has determined that the object is not in focus, signals will be sent to the motors who are responsible for adjusting the focus knobs of the system. This subsystem will consist primarily of focus knobs and motors, and it will also provide the overall system with the ability to sharpen its focus on the object of interest.
4. Image Capture/Processing: After the system has successfully determined that the object of interest (OI) is in focus, the image capture/processing subsystem will then capture multiple snaps of the same image and process them in order to increase the light intensity of the image and improve overall clarity. This subsystem will provide the user with a detailed picture of the object of interest.

3.4. Modes of Operations

The Telescope Autofocuser will have one mode of operation which we will call “automatic”. In this mode, the user will point the telescope in a general direction and select the OI to be focused on; else, the system will not begin focusing. In this mode of operation, the telescope will feed the image to an analyzer that communicates with motors. These motors will have control over the focus of the telescope. Once the analyzer has determined that the image is in focus, multiple images will then be captured and processed in order to produce a detailed picture of the object of interest.

3.5. Users

Our autofocusing system will be primarily marketed to astronomers and those interested in astronomy who want to capture high quality images of celestial objects. This product will reduce the time and effort needed to clearly identify an OI and it will do so with minimum user experience.

The Telescope Autofocuser could also be utilized by ships through installing the system on the upper deck to identify unknown vessels far out into the sea and prevent possible collision.

3.6. Support

Support for the Telescope Autofocuser will be provided in the form of a detailed user's manual. The manual will consist of instructions on how to set up the system, usage, and maintenance.

4. Scenario(s)

4.1. Deep Space Images Requiring Long Exposures

Deeper space images require longer exposure time, so manual refocusing is typically required every 5 to 10 minutes. To avoid the precision and consistency needed in manual refocusing, the Telescope Autofocuser will aid the user in automatically capturing and focusing long exposure images.

4.2. Low Performance of the Human Eye

The human eye is often not capable of quality optical performance in low intensity light. The Telescope Autofocuser utilizes an imager that is capable of capturing these low light images. In addition, the telescope will assist users with focusing on low light and producing astronomical images that a human eye cannot view naturally.

4.3. Offset Temperature Related Contractions

Telescope usage is often utilized during the evening and nighttime where temperatures begin to decrease. Critical components of the telescope, such as the optical tube assembly and focuser drawtube, will contract due to the lower temperatures. The Telescope Autofocuser will assist by autofocusing, relieving the user of manually refocusing the telescope to offset the temperature changes.

4.4. Introduction for Amateur Astrophotographers

In astrophotography, manual focus is a skill that is learned over time with an abundance of image capturing. Beginners in this field that desire high quality images with minimal experience can use the Telescope Autofocuser to ensure the same quality image capturing as taken by an experienced astrophotographer.

5. Analysis

5.1. Summary of Proposed Improvements

- An imager positioned on the eyepiece of the telescope. The imager will be capable of capturing images in low light intensity.
- A mounting system that is capable of attaching to an Adventura Newtonian Telescope without damaging any of the autofocusing instruments as well as the telescope's body.
- A stepper motor assembly will rotate a crayford focuser to achieve a focused image.
- An automated software solution will be applied to capture images in real-time and in post-processing. Here, machine learning will be leveraged to optimize the live feed of captured images. However, a more primitive, but effective

- implementation will stack multiple images into a final photo. Both methods offer effective ways to optimize the desired results.
- Autofocus techniques involving heuristic raster processing will allow the telescope controls to seek out an optimal contrast. In order to ensure quality image capture and performance, real-time processing techniques will offer excellent results for time-critical tasks.

5.2. Disadvantages and Limitations

The Telescope Autofocuser will have limitations that include:

- The assembly which is designed to work with the sizing and constraints of an Apertura Newtonian Telescope. It will not be able to cross different product types and brands if desired.
- The medium capturing images will need a sufficient level of contrast lighting to adequately adjust the telescope dials. Adjusting the dials will cause the feedback loop to sharpen captured images. For a distant, or dimly lit target, this can be challenging to achieve.
- The product will only perform well in standard telescope observing conditions, such as skies with limited visible clouds, low humidity levels, and minimal to no dust concentration.
- Design, prototyping, and fabrication must be operated within a finite budget.

5.3. Alternatives

- A sensor that when perceiving a focused image will signal the user either audibly or visually to stop manually rotating the Crayford focuser. Unlike our proposed solution, this requires the user to be more involved in the focusing process. This solution will better suit users that want more handle in the process and can enable them to make slight adjustments that would not be capable in our solution.
- instead of using an imager, a solution would be to attach a standard camera to the telescope eyepiece. The user can utilize in-built focus and zoom settings to get a desired image. Negative aspects in this solution include that there is no post processing for a better-quality image.
- Purchasing a built-in autofocus feature on a new telescope; however, this is a far more expensive approach than our proposed solution that will be functional for the same needs of the consumer.

5.4. Impact

Our Telescope Autofocuser will impact the following:

- **Astrophotography Community** The ability to autofocus on an object of interest will reduce the astrophotography learning curve which in turn will bring more individuals into the astrophotography community. This will have a positive effect on the astronomy industry/market.

- **Image Quality** The autofocuser will reduce human error when capturing images. In turn, the images that use this product will be clearer and more accurate, which can help astrophotographers retrieve proper data and reduce the number of wasted captures. To add, cheaper (and more accessible) hardware can leverage autofocus modules which in turn produce similar results to more expensive telescope models.
- **Open-Source Contributions** Since the growth of amateur engineers and hobbyist are increasing every year, providing a significant contribution to the “hacker” community would likely foster positive engagement. By offering free and open-source software and hardware solutions, project traction has a high chance of improving with regard to security, material needs, efficiency, and quality. As opposed to hidden proprietary solutions, open source offers more aspects of digital freedom to circulate popular implementations.
- **Ethical Concerns** In an era where privacy and control over personal devices is a desired trait, most modern devices appear to fall short in offering open access to the owners of said devices. Despite the fact that the code and designs implemented will be free and open source, the device’s application to the machine will likely make it difficult to use once mounted. With this in mind, we plan to offer a “hands-on” mode, but this will likely be impeded by the devices.
- **Security Concerns** It’s important to note that every device is imperfect and likely has an unknown (or known) security exploit. Specifically, development boards such as Raspberry Pis, Arduino’s, and Beagle Bones are infamous for their exploitability as a victim machine; the more features on the board, the more exploitable it becomes. Despite our best efforts to make the application secure, the hardware or embedded software has the potential to reveal user information.

Telescope Autofocuser

Camille Watson
Joseph Basdeo
Albin Myscich
Alonna Too-Chiobi

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 1

Functional System Requirements

Telescope Autofocuser

25 January 2018

Revision - 1

FUNCTIONAL SYSTEM REQUIREMENTS FOR Telescope Autofocuser

PREPARED BY:

Author **Date**

APPROVED BY:

Project Leader _____ **Date** _____

John Lusher, P.E. Date

T/A Date

Change Record

Rev	Date	Originator	Approvals	Description
-	2/23/22	Telescope Autofocuser Team		Revision 1

Table of Contents

List of Figures	6
Introduction	6
Purpose and Scope	6
Responsibility and Change Authority	6
Applicable and Reference Documents	9
Applicable Documents	9
Reference Documents	9
Order of Precedence	10
Requirements	11
System Definition	11
Characteristics (break down by subsystem)	13
Functional/Performance Requirements	13
Image Focusing Threshold	13
Image Processing Duration	13
Imager Pixel Size	13
Imager Quantum Efficiency	13
Imager Resolution	13
Motor Rotary Movement	14
Motor Energy Conversion	14
Mounting Assembly Load	14
Mounting Assembly Security	14
Mounting Assembly Accessibility	14
Physical Characteristics	14
Mass	14
Mounting	14
Motor Geometry	14
Mounting Assembly Material	14
Electrical Characteristics	15
Inputs	15
Power Consumption	15
Input Voltage Level	15
External Commands	15
Outputs	15
Image Output	15
Environmental Requirements	15
Thermal	15
External Contamination	15

Functional System Requirements
Telescope Autofocuser

Revision - 1

Rain	15
Humidity	16
Failure Propagation	16
Failure Detection, Isolation, and Recovery (FDIR)	16
Mounting Assembly Maintenance	16
Built In Test (BIT)	16
BIT Critical Fault Detection	16
BIT False Alarms	16
Isolation and Recovery	16
Support Requirements	17
Appendix A: Acronyms and Abbreviations	18
Appendix B: Definition of Terms	19

List of Tables

No table of figures entries found.

List of Figures

Figure 1. Project Conceptual Image	6
Figure 2. Closed-Loop Control System	12
Figure 3. Flow Control Logic	12
Figure 4. Block diagram of system	12

1.

1. Introduction

1.1. Purpose and Scope

In astrophotography, observing celestial bodies is often achieved via powerful long-range telescopes. However, operating these devices takes skill and practice in order to obtain a detailed image of a celestial object. Our aim is to develop a product that reduces the astrophotography learning curve by providing users with the ability to automatically focus on their body of interest. The Telescope Autofocuser will be able to automatically focus on an object and then send that image to an image processor where the picture detail will be improved. Our product will replace the primitive action of manually adjusting your telescope which will benefit astronomers by saving them time.

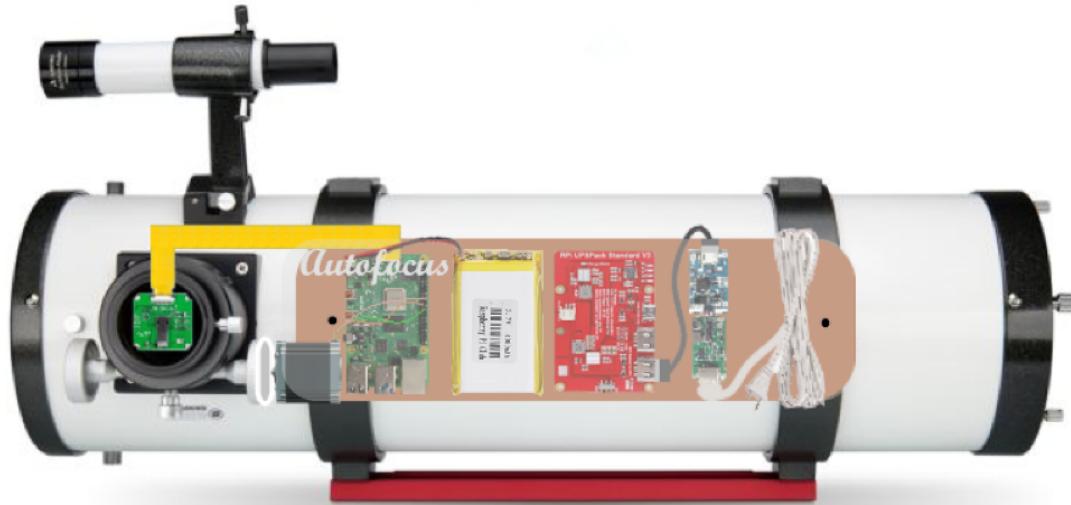


Figure 1. Project Conceptual Image

1.2. Responsibility and Change Authority

The team leader Albin Myscich will be responsible for ensuring that the requirements of the system are met. Subsystem owners will be responsible for the fulfillment of their respective subsystems. The team leader will also have the authority to make changes as long as the change has been agreed upon by the team. Subsystems are listed below:

2. Imager & Mounting Assembly (*Joseph Basdeo*)

The Imager or Astro Analyzer (AA) is responsible for viewing images through the eyepiece of the telescope. It then captures an image and feeds the information to the autofocus program subsystem. The imager will continuously transmit data until the current image is determined as “focused.” Then the imager will transmit multiple captures of the same image to be processed by the image processing subsystem. It is required that the imager is attached externally to the telescope via a mounting assembly. The assembly is able to be removed without damaging any of the telescope components, and placement should assist or not get in the way of the other subsystems.

3. Motor & Mounting Assembly (*Alonna Too-Chiobi*)

The motors are solely responsible for adjusting the focus of the telescope. Depending on the input it receives from the image processor, the motors will make precise clockwise/counterclockwise adjustments. The mounting assembly is responsible for ensuring the position and security of the motors.

4. Power Supply (*Camille Watson*)

The power supply will be responsible for providing the Raspberry Pi and motor subsystem with sufficient enough power to easily perform their functions. For the power supply, we are using a 12V/6A Li-Ion battery with charging capabilities. Additional components for the power supply subsystem include the charging cable, splitter cable, and a voltage regulation system to control the power output received by the loads.

5. Image Processing (*Albin Myscich*)

Image processing primarily deals with receiving real-time and post-processed images and applying an intelligent system interface capable of refining target images. This plan of action is ultimately achieved with machine learning techniques and statistically-based enhancements. The goal in mind is to accept an incoming stream of data from the camera into the development board, refine the images, and export the data to memory in the desired file format. To ensure quality data processing, testbench assessments of different image processing models must be implemented and compared.

6. Autofocusing Program (*Albin Myscich*)

As the name might imply, the autofocus program is responsible for receiving a continuous stream of data from the camera, and adjusting the focal distance from itself and the mirrored lens via a stepper motor. Here, most of the motor controls will be driven based on the raster image edge detection. Ideally, a minimal amount of wires are desired to ensure simplicity, organization, and performance. If images are deemed to bury, a closed

feedback loop will be implemented to ensure a directed dial adjustment sweep to sharpen images to the desired tolerance. This tolerance will be further defined with distinct normalized sharpness evaluation function values. To ensure quality data processing, testbench assessments of different image processing models must be implemented and compared.

2. Applicable and Reference Documents

2.1. Applicable Documents

Document Number	Revision/Release Date	Document Title
IPC A-610E	Revision E – 4/1/2010	Acceptability of Electronic Assemblies
IEEE 1349-2011	15/10/2021	IEEE Guide for Application of Electric Motors in Class I, Division 2 and Class I, Zone 2 Hazardous (Classified) Locations
NISTIR 8059	05/2015	Materials Testing Standards for Additive Manufacturing of Polymer Materials: State of the Art and Standards Applicability

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
IEC 60364-5-52	2nd edition/2001	Wiring systems

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings, or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

This section defines the minimum requirements for the Telescope Autofocuser. The requirements and constraints apply to performance, design, and reliability.

3.1. System Definition

The primary focus of the Telescope Autofocuser is to replace manually configuring the focal setting and image processing with an automated medium. The apparatus is divided into three separate parts; eyepiece interface, dial manipulation, and image processing.

To take a clear image, the main notion of the autofocus equipment is to regulate the lens position in the system so that the camera is optimally positioned at the focal plane. From here, an ample number of high-frequency elements are received to identify a definite contour of an image's boundaries. If the scales of related high-frequency factors are larger than some present values, then the image is deemed sufficiently sharp.

External components will include, a PC management, stepper motors, breakout camera, and mechanical manipulation apparatus. To enhance the quality of captured images, a Schmidt corrector can be installed to the telescope front; a camera with a thin frequency band filter can be applied to receive images and convolute the sieved results to the PC management board. In this component, a stepper motor is employed to adapt the position of the main image of the telescope in the controller mechanism based on the image evaluation.

The auto-focus procedure can be described as beginning with the camera which captures data concerning the tracked objects with the telescope. It then transmits the image to the PC management board. Then, the dev board's software performs a calculation based on the image and directs instructions to rotate the stepper motor counterclockwise or clockwise. As a result, it will successively adjust the imaging position and focus. The processes described are ultimately recurrent events until the camera recognizes the telescope's image plane. Therefore, a clear image is then obtained, which defines this system's schema as a closed-loop system.

Within the realm of software, the focus search algorithm, with the help of Hemli and Scherer's mean (HSM), Tenengrad (TGR), and Gaussian derivative (GDR) are responsible for the focus quality and efficacy of the system. In the testing phase, the closed-loop control system and flow control logic approach can be used to accomplish a near-optimal focusing effect [Figure 2, 3]. From here, the stepper motors will slowly rotate the telescope's knob at a suitable step length to adjust the system's focal length from which the image estimation function value is calculated to supply a constant stream of feedback. When the image estimation function begins to decrease, it implies that the optimal focusing location has been passed. As a direct response, the knob can be rotated a step in reverse. Next, the present step length is then cut in half, and the search procedure will recur until the exploration is concluded with the lowest step size reached. As a result, the auto-focus procedure can be deemed complete.

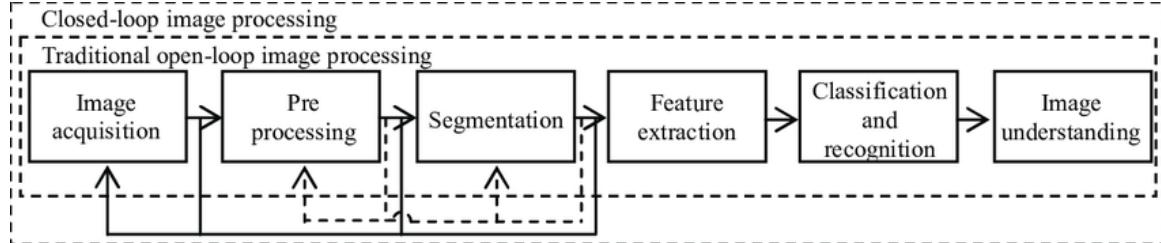


Figure 2. Closed-Loop Control System

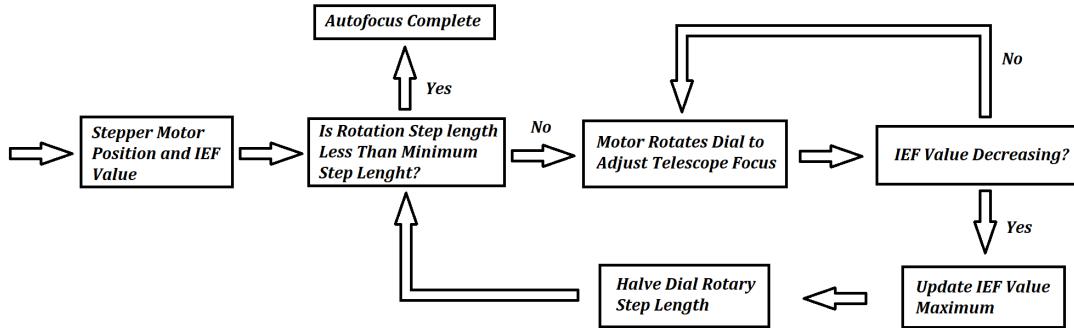


Figure 3. Flow Control Logic

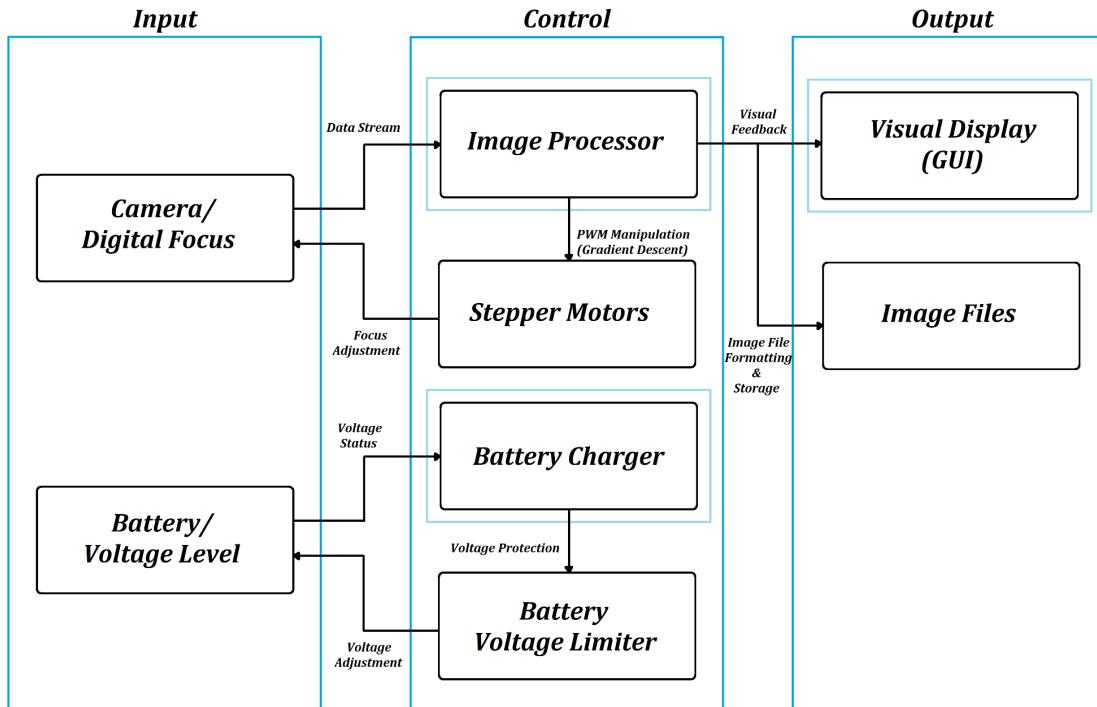


Figure 4. Block Diagram of System

There will be a power supply that provides power for the camera/focus, dev board, and motors. Once the system is activated, and the user has oriented the device in the

direction of the interest, the camera will take a picture of the celestial object and send that image to the image processor (located in the control unit). The image processor will then evaluate the image and then proceed with one of the following options:

1. In the case that the image is not sharp enough, the processor will send instructions to the motors to adjust the focal knobs of the telescope. Once the motors are done adjusting the camera will then take another image and once again send it to the image processing unit.
2. In the case that the image is deemed to be sharp enough, the CPU will send the image data to a GUI where the user can view it. The CPU will also convert the image into an image file that can be downloaded onto an external drive.

3.2. Characteristics (break down by subsystem)

3.2.1. Functional/Performance Requirements

3.2.1.1. Image Focusing Threshold

The focused image is determined when the width of stars is less than 5 pixels.

Rationale: The image is considered to be focused when the stars are at a width no larger than one pixel. Our goal is to get as close to that value as possible given the limited pixel resolution.

3.2.1.2. Image Processing Duration

The approximate latency for image processing should be no more than 10 milliseconds

Rationale: The image processing needs to be as fast as possible. It is dependent on the frames per second the imager can transmit to the Raspberry Pi.

3.2.1.3. Imager Pixel Size

The pixel size of the astro-analyzer should be no more than the minimum 3.75um.

Rationale: The Telescope Autofocuser operates at a higher performance the larger the pixel size. Having a minimum 3.75um pixel size allows the autofocusing program to cipher focused from non-focused images with better data.

3.2.1.4. Imager Quantum Efficiency

The quantum efficiency of the astra imager will be at least 70%.

Rationale: The higher the QE, the higher the performance of the imager. Best cameras can reach up to 95% QE depending on light wavelength that is detected. Having a moderately high QE will enable high-quality image capturing.

3.2.2. Imager Resolution

The minimum image resolution should be larger than 960 x 960 pixels.

Rationale: Higher resolution attributes to increase autofocusing performance. Below the threshold of 960 pixels, the focus threshold widens, and focus inaccuracy is common. The ideal resolution can be over 4096 but is unlikely due to cost constraints.

3.2.2.1. Motor Rotary Movement

The motor subsystem shall have the ability to make precise rotations in clockwise and counterclockwise directions.

Rationale: This ability allows the system to rapidly focus on the object of interest with optimized detail.

3.2.2.2. Motor Energy Conversion

The motor subsystem will have the ability to convert electrical energy into mechanical energy.

Rationale: This is the fundamental purpose of electrical motors and without it, the system would be unable to focus on an object of interest.

3.2.2.3. Mounting Assembly Load

The mounting assembly shall be able to support the weight of the motors and astro-analyzer (AA).

3.2.2.4. Mounting Assembly Security

The mounting assembly shall be able to securely hold both the motor(s) and AA.

Rationale: This is to prevent unnecessary movement of the motor and analyzer components.

3.2.2.5. Mounting Assembly Accessibility

The mounting assembly shall be able to release the motor(s) and AA. It shall also have the ability to be removed from the system.

Rationale: This allows for easy access to the individual subsystems.

3.2.3. Physical Characteristics

3.2.3.1. Mass

The mass of the system should not exceed 7kg

Rationale: To prevent the purchase of a higher rated tripod mount.

3.2.3.2. Mounting

Mounting for the imager and motor should not harm or negatively affect the telescope apparatus.

Rationale: Damaging the telescope can result in damages to key components such as the lens. A damaged lens will yield low-quality images.

3.2.3.3. Motor Geometry

The motor shall have a shape that optimizes its security by the mounting assembly.

Rationale: This will make design a mounting

3.2.3.4. Mounting Assembly Material

The mounting assembly shall be 3D printed using plastic filament.

Rationale: Print filament is lightweight and durable enough to stabilize both the imager and motor subsystems.

3.2.4. Electrical Characteristics

3.2.4.1. Inputs

A sequence of captured images or user setting configurations should not corrupt the system when autofocusing or image processing.

Rationale: The TA is designed to be capable of capturing, transmitting, and processing images.

3.2.4.1.1 Power Consumption

The maximum peak power consumption should not exceed 24W.

Rationale: The TA with proposed currents of each subsystem will not exceed 24W due to our project being limited by a battery only producing 12V.

3.2.4.1.2 Input Voltage Level

The Telescope Autofocuser should be able to run under the supply of 12V.

Rationale: The TA uses a 12V battery to operate all subsystems.

3.2.4.1.3 External Commands

The Telescope Autofocuser shall document all external commands in the appropriate ICD.

Rationale: The ICD will cover all interface details

3.2.4.2. Outputs

3.2.4.2.1 Image Output

The Telescope Autofocuser will output the processed final image to the user through the GUI.

Rationale: The purpose of the TA is to give a user high-quality and focused images.

3.2.5. Environmental Requirements

The Telescope Autofocuser shall be designed to withstand and operate in the environments and laboratory tests specified in the following section.

3.2.5.1. Thermal

The Telescope Autofocuser should operate in environments below 28 degrees Celsius.

Rationale: Temperatures above 85 degrees can negatively affect the imaging of the telescope. The telescope is kept in an indoor environment ranging from 70-75 degrees. For best results, the telescope should operate in minimum temperature-changing conditions.

3.2.5.2. External Contamination

The Telescope Apparatus should be kept in a clean environment and covered.

Rationale: External factors such as dust, dirt, and other materials pose a risk to the telescope. These particles can enter the telescope and harm the lens. There will be low-quality images with damaged components.

3.2.5.3. Rain

The Telescope Autofocuser will not operate in rainy weather conditions.

Rationale: Rainy weather conditions can damage electronics if the telescope is operated outdoors. Operating the telescope indoors while raining is still not advised as there are no clear skies to gather quality images.

3.2.5.4. Humidity

The Telescope Autofocuser should operate under 50% humidity.

Rationale: High humidity harms image quality. A high density of water particles clouds the atmosphere which will scatter and hinder the light on its path to the telescope. High humidity increases the risk of condensation which is dangerous for exposed electronics.

3.2.6. Failure Propagation

The Telescope Autofocuser System shall not allow propagation of faults beyond the Telescope Autofocuser system interface.

3.2.6.1. Failure Detection, Isolation, and Recovery (FDIR)

3.2.6.2. Mounting Assembly Maintenance

The mounting assembly will be periodically (once a week) evaluated for damages such as cracks, strains, and other visual defects.

Rationale: Routine maintenance will prevent catastrophes that will not only affect this subsystem, but also the subsystems it secures.

3.2.6.2.1 Built In Test (BIT)

The Telescope Autofocuser shall have an internal system that will alert the user when the image is considered focused.

Rationale: This will give the user autonomy on whether they would like to proceed with capturing the image or rerun the autofocus if deemed necessary.

3.2.6.2.1.1 BIT Critical Fault Detection

The Telescope Autofocuser should be able to detect a critical fault during the autofocus and prior to the image processing 95% of the time.

Rationale: This is to ensure that the system does not waste time in processing unfocused images.

3.2.6.2.1.2 BIT False Alarms

The BIT shall have a false alarm rate of less than 5 percent.

Rationale: This can be due to critical faults with post image acquisition as we are not accounting for image processing error.

3.2.6.2.2 Isolation and Recovery

The Telescope Autofocuser should provide the location of the fault followed by a reset of the system.

Rationale: This is for providing the user with the most likable reason for error, and will enable the user to continue using the system after an attempt to solve the issue by resetting.

4. Support Requirements

The customer will be provided with a manual that will include operating instructions and technical support for equipment used. Note the manual will not include technical support or operating instructions for external systems such as display instruments or storage devices. The user is responsible for providing their own display device that is compatible with an HDMI input. User is responsible for an external hard drive that they would wish to store instead of the storage on the Raspberry Pi.

The TA will include the following: Apertura 6" f/5 Newtonian OTA - 6F5N

- Imager (Astro-Analyzer) and imager mounting assembly
- Motor and motor mounting assembly
- Power Supply
- Raspberry Pi B+

7. Appendix A: Acronyms and Abbreviations

AA	Astro-Analyzer
TA	Telescope Autofocuser
BIT	Built-In Test
CCA	Circuit Card Assembly
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EO/IR	Electro-optical Infrared
FOR	Field of Regard
FOV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
LCD	Liquid Crystal Display
LED	Light-emitting Diode
mA	Milliamp
MHz	Megahertz (1,000,000 Hz)
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
mW	Milliwatt
PCB	Printed Circuit Board
RMS	Root Mean Square
TBD	To Be Determined
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
VME	VERSA-Module Europe
V	Volt
A	Amp
W	Watt

8. Appendix B: Definition of Terms

9.

10.

Telescope Autofocuser
Camille Watson
Joseph Basdeo
Albin Myscich
Alonna Too-Chiobi

INTERFACE CONTROL DOCUMENT

REVISION – 1
23 February 2022

INTERFACE CONTROL DOCUMENT

FOR

Telescope Autofocuser

PREPARED BY:

Author Date

APPROVED BY:

Project Leader _____ **Date** _____

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/21/22	Telescope Autofocuser		Draft Release

Table of Contents

Overview	8
References and Definitions	9
References	9
Definitions	10
Physical Interface	11
Weight	11
Imager Weight	11
Motors Weight	11
Mounting Assembly Weight	11
Power Supply Weight	11
Dimensions	12
Dimension of Imager	12
Dimension of Motor	12
Dimension of Mounting assembly	12
Dimension of Power Supply	12
Mounting Locations	13
Thermal Interface	13
Electrical Interface	14
Primary Input Power	14
Polarity Reversal	14
Signal Interfaces	14
Video Interfaces	14
User Control Interface	14
Communications / Device Interface Protocols	15
Wireless Communications (WiFi)	15
Host Device	15
Video Interface	15
Device Peripheral Interface	15

List of Tables

Component	Weight	Number of Items	Total Weight
Imager	5 g	2	10 g

Table 1: Imager Weight Specifications

Component	Weight	Number of Items	Total Weight
Motor	76 g	1	76 g

Table 2: Motor Weight Specifications

Component	Weight	Number of Items	Total Weight
Mounting Assembly (Dev Board Standoff)	2 g	6	12 g
Mounting Assembly (Step Motor Joint)	20 g	2	40 g
Mounting Assembly (Step Motor Mount)	20 g	2	40 g
Mounting Assembly (Imager Mount)	10 g	1	10 g

Table 3: Mounting Assembly Weight Specifications

Component	Weight	Number of Items	Total Weight
Power Supply (12V Li-Ion Battery and Charger)	635 g	1	635 g
Buck Converter	40 g	2	80g

Table 4: Power Supply Weight Specifications

Component	Weight	Number of Items	Total Weight
Raspberry Pi	250 g	1	250 g

Stepper Motor Driver	73 g	1	73 g
----------------------	------	---	------

Table 5: Development Board Weight Specifications

Component	Length	Width	Height
Imager	20 mm	20 mm	5 mm

Table 6: Imager Dimensions Specifications

Component	Length	Width	Height
Motor	42 mm	42 mm	82 mm

Table 7: Motor Dimensions Specifications

Component	Length	Width	Height
Mounting Assembly (Dev Board Standoff)	7 mm	7 mm	7 mm
Mounting Assembly (Step Motor Joint)	50 mm	50 mm	20 mm
Mounting Assembly (Step Motor Mount)	50 mm	50 mm	20 mm
Mounting Assembly (Imager Mount)	32 mm	32 mm	11 mm

Table 8: Mounting Assembly Dimensions Specifications

Component	Length	Width	Height
Power Supply (12V Li-Ion Battery and Charger)	138 mm	80 mm	39 mm
Buck Converter	85 mm	55 mm	10 mm

Table 9: Power Supply Dimensions Specifications

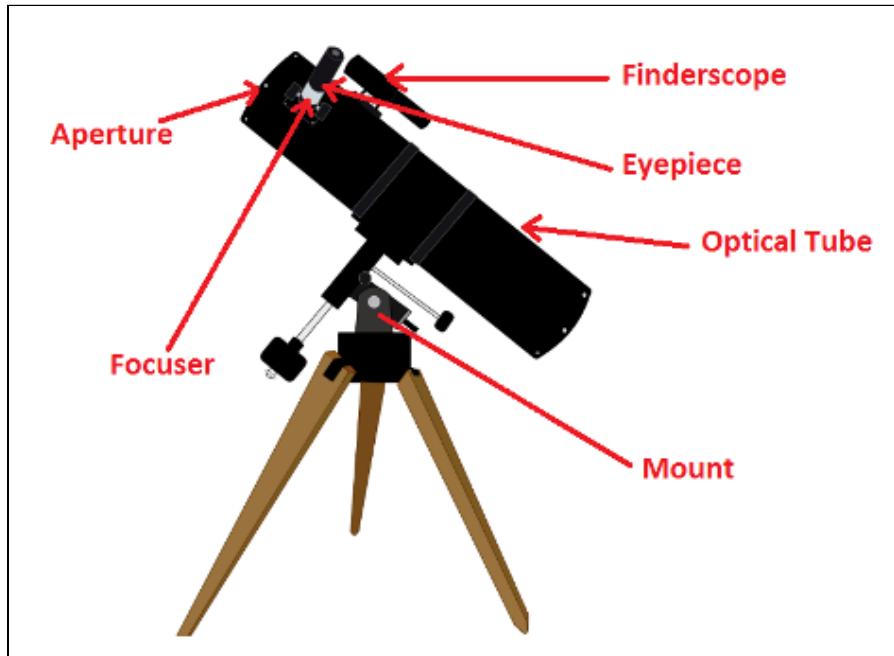
Component	Length	Width	Height

Raspberry Pi	85.1 mm	55.9 mm	20.3 mm
Stepper Motor Driver	112.2 mm	52.6 mm	25.7 mm

Table 10: Development Board Dimensions Specifications

List of Figures

1.



2.

Figure 1

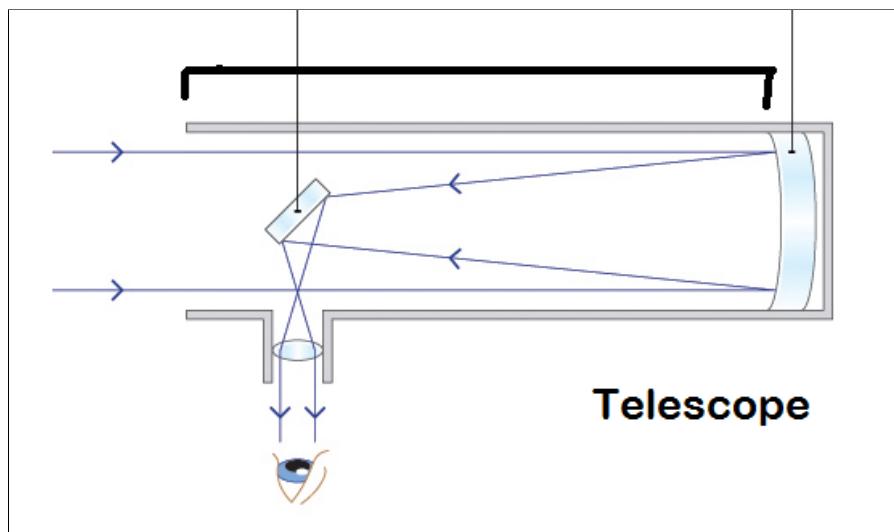


Figure 2

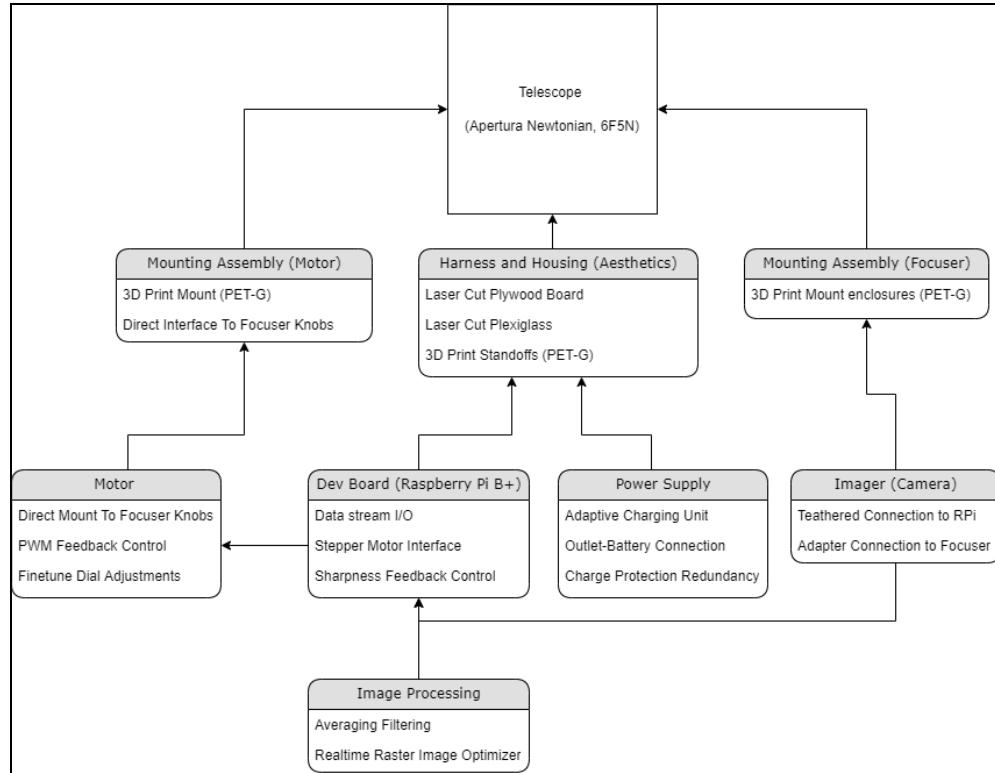


Figure 3

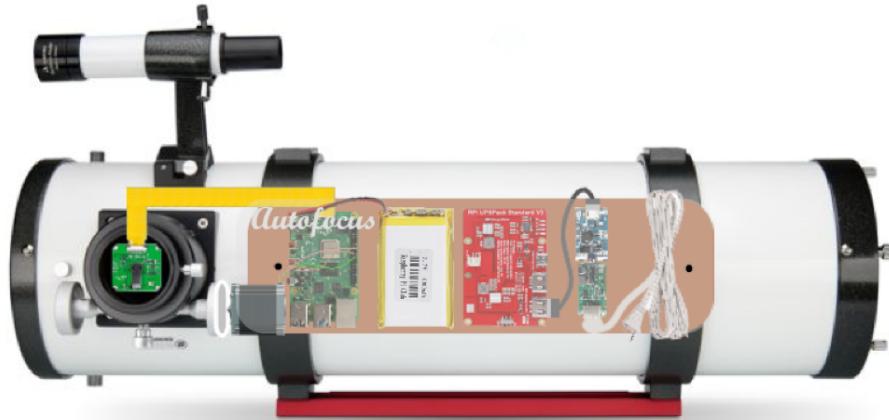


Figure 4

1. Overview

The primary focus of the ICD is to describe the set of interfaces and their interactions of the Telescope Autofocusing system. Here, details concerning the physical, thermal, electrical, and communication interfaces. More specifically, the set of subsystems encompasses the

development board, stepper motors, power supply, and imager. Each of these require an interface or adapter to effectively interact with adjacent subsystems. To generalize each of these interfaces, the three groupings are defined as software, hardware, and mounting; Image processing, dev board signal management (PWM feedback control by extension), and image I/O streaming belong to the software category. Power supply, imager, and motor belong to the hardware, and each of the mounting, harnesses, and housings belong to the mounting category [Figure 3].

2. References and Definitions

2.1. References

3. *Amazon.com: Talentcell Rechargeable 72W 100Wh 12V/8300mah ...* (n.d.). Retrieved February 24, 2022, from <https://www.amazon.com/TalentCell-Rechargeable-11000mAh-20000mAh-Portable/dp/B01337QXMA>
4. *Apertura 2" photo & visual coma corrector for newtonian telescopes - A-COMA.* Telescopes. (n.d.). Retrieved February 24, 2022, from <https://www.highpointscientific.com/apertura-2-newtonian-coma-corrector-photo-visual-a-coma>
5. *Lipo battery hat for Raspberry Pi.* The Pi Hut. (n.d.). Retrieved February 24, 2022, from <https://thepihut.com/products/lipo-battery-hat-for-raspberry-pi>
6. *Motorizing the IOT with battery-powered Stepper Motors - industry articles.* All About Circuits. (n.d.). Retrieved February 24, 2022, from <https://www.allaboutcircuits.com/industry-articles/motorizing-the-iot-with-battery-powered-stepper-motors/>
7. Name *. (2021, October 11). *ARDUCAM 18mp AR1820HS color 1/2.3" CMOS color rolling shutter camera module for Raspberry Pi.* Arducam. Retrieved February 24, 2022, from <https://www.arducam.com/product/arducam-ar1820hs-1-2-3%E2%88%92inch-18mp-color-camera-module/>
8. *Pulse Width Modulation for dummies.* Rich Youngkin. (n.d.). Retrieved February 24, 2022, from <https://youngkin.github.io/post/pulsewidthmodulationraspberrypi/>
9. *Rolling vs global shutter - learn.* Teledyne Photometrics. (2021, October 18). Retrieved February 24, 2022, from <https://www.photometrics.com/learn/advanced-imaging/rolling-vs-global-shutter>
10. *RPI-Cam-web-interface.* RPi-Cam-Web-Interface - eLinux.org. (n.d.). Retrieved February 24, 2022, from <https://elinux.org/RPi-Cam-Web-Interface>
11. *Stepper motor frame sizes.* Oriental Motor U.S.A. Corp. (n.d.). Retrieved February 24, 2022, from <https://www.orientalmotor.com/stepper-motors/stepper-motor-frame-sizes.html>
12. *Wedgelock Thermal & Clamping Force Lifecycle Test Methodology.* WaveTherm Corporation. (n.d.). Retrieved February 24, 2022, from <https://shop.wavetherm.com/pages/wedgelock-thermal-and-clamping-force-lifecycle-test-methodology#:~:text=Cold%20Wall%20%E2%80%93%20A%20heat%20sink,by%20the%20PCB%2Fheat%20frame.&text=Test%20Fixture%20%E2%80%93%20A%20representative%20cold,finned%20heat%20sinks%20and%20fans.>

2.2. Definitions

CCA	Circuit Card Assembly
mA	Milliamp
mW	Milliwatt
MHz	Megahertz (1,000,000 Hz)
TBD	To Be Determined
TTL	Transistor-Transistor Logic
VME	VERSA-Module Europe
Dev Board	Development Board (Raspberry Pi)
Imager	Camera/Photo-image Device

2.3.

3. Physical Interface

The physical interfaces of the system are composed of three categories mentioned in the overview (software, hardware, and mounting). The software will drive the general behavior of the separate hardware interactions on and with the telescope. Moreover, the mounting will ensure hardware adherence and aesthetics on the telescope [Figure 4]. Here, the development board (Raspberry Pi B+) will act as the brain of the system where data streaming into the board from the focuser will direct the stepper motors to adjust the focuser dials and effectively create a closed-loop system, which offers rotational feedback on image sharpness [Figure 2].

3.1. Weight

3.1.1. Imager Weight

The weight of the imager is represented in table 1. In addition, the imager's weight can be described as one of the lightest items in the project as minimal interference within the weight in the closed feedback loop is desired. It's important to note that the missing component, *Apertura 2" Photo & Visual Coma Corrector for Newtonian Telescopes*, will not be accounted for since it was an original component of the telescope. However, the supplemental digital wire extension tape will be taken into account as the wires are an extension of the camera.

3.1.2. Motors Weight

The weights of the motors are described in table 2. Despite the fact that the weight of the motor doesn't play a quintessential role in affecting the performance of the optical focus, fine-tuning and knob adjustments will remain as ideally manipulated as possible. From what we can observe, the motor weight is kept moderate where the power output is strong enough to be considered sufficient to manipulate the knobs with ease.

3.1.3. Mounting Assembly Weight

The mounting assembly consists of a set of high-strength ABS plastic, plexiglass, and plywood. Of the materials, the bulk of the weight comes in the form of a thin sheet of plywood and plexiglass. On the other hand, the ABS plastics are 3D printed and will make up the actual adapter, standoffs, and mounts. The overall weight of the plastics is approximated in table 3.

3.1.4. Power Supply Weight

The power supply easily possesses the most diverse set of components that contribute to its mass. Of which the portable battery and charging cable are included. The total weight of the components are laid out in table 4. Minor details involving the methods used to affix the battery and boards to the telescope are not included in this section.

3.1.5. Development Board Weight

The development board possesses a simple design which alone has a minimal weight attributed to the whole project. The weight of the development board is described in table 5.

3.2. Dimensions

The primary metric of dimensions is measured in the International System (SI) of Units. This decision was made to ultimately simplify the unit conversions and equations applied throughout the project's entirety.

3.2.1. Dimension of Imager

The dimensions of the imager are fairly conservative as the length, width, and height are all minimal by design. Moreover, the overall dimensions of the imager can be described in table 6. Note, the imager directly interfaces with the 3D printed adapter, which attaches directly to the component: *Apertura 2" Photo & Visual Coma Corrector for Newtonian Telescopes*. The interface and adapter are not accounted for since they are not defined with the imager itself, but rather as an interface or adapter.

3.2.2. Dimension of Motor

The dimensions of the stepper motor make up a considerable portion of the physical manipulation control platform on the focuser knob. Here the dimensions of the motor are kept minimal by remaining on the side of the focuser closest to the project mount. As mentioned previously, the stepper motor directly interfaces to the 3D printed adapter, which attaches directly to the adjustment dial. The overall dimensions of the motor can be described in table 7.

3.2.3. Dimension of Mounting assembly

This section is a little more difficult to gauge since it comprises many adapters, interfaces, and miscellaneous parts. In any case, the designs of each of the parts are created to be as minimal as possible to accommodate as many 3D printer bed sizes as possible since we are encouraging an open source design for at-home hobbyists. Of the selection, the standoffs are easily identified with the smallest dimensions, whereas, the motor stepper interface has the largest dimensions due to the inherently large dial. The overall dimensions of the mounting assembly can be described in table 8.

3.2.4. Dimension of Power Supply

The power supply was selected with distinguished size and capacity to supply the Raspberry Pi and driver. In addition, a voltage regulation system is included between the Li-Ion battery and the loads. Here, dimensions of the power supplies can be described in table 9.

3.2.5. Dimensions of Development Board

The development board itself is quite possibly the most well known item throughout the entire list. This is simply due to the fact that the Raspberry Pi B+ makes up the subassembly. It appears the dimensions in table 10 describes the sizing of the boards.

3.3.

3.4. Mounting Locations

Here the primary desired locations of which their interfaces are planned to possess an adapter, or mount exist on the adjustable focuser knobs, focuser, and parallel to the optical

tube's length [Figure 1, 4]. Here, each of the subassemblies are easily accessible and are organized in a clear and concise manner tangential to the telescope. Furthermore, direct adapters attached to the focuser, knobs, devboard, and chargers are each designed to fit around the part and offer easy access to the 3D printed housing. Here, supplemental parts will fit accordingly with the assistance of joints, or standoffs.

4. Thermal Interface

After reviewing the interplay between each subassembly, and alternative offer, it appears the Raspberry Pi, and motor controllers are going to overheat the most due to their mass of calculations necessary. Of course, as a natural response, applying a heat sink to the Raspberry Pi CPU, and the motor controllers would result in a negligible extra bulk and more pragmatic design. Since the Raspberry Pi is expected to sustain consistent and intensive computations due to image streaming and processing amid other neighboring device interactions. These heatsinks have been strategically placed to reduce chances of overheating (as the name might imply), and effectively extend the life of the board. Since traditional Raspberry Pi boards do not include cold walls, we will assume (for now), that this is not a strict requirement and will not be installed until deems necessary. A cold start heat sink would be nice to possess, but adds more essential items to the list of required items that hobbyists and generic reviewers must keep track of.

5. Electrical Interface

5.1. Primary Input Power

The primary input power will come from a 12V, 8300mAh Lithium-Ion battery to drive the Raspberry Pi and the driver; this of course is regulated between the power supply and the loads.

5.2. Polarity Reversal

At the moment, there is no need to reverse the polarity of any of the components to fulfill the desired designs.

5.3. Signal Interfaces

The signal interfaces will come in different forms to and from the development boards. From the standpoint of the image capture, and closed loop focus adjustment, a rolling shutter stream of data will be streamed directly to the Raspberry Pi for either post-processing, or real-time analysis. On the other hand, the development board will also handle and drive the PWM intensity and direction based on the feedback of the real-time image sharpness.

5.4. Video Interfaces

In order to squeeze the highest quality images out of available hardware, the *Arducam 18MP AR1820HS Color 1/2.3" CMOS Color Rolling Shutter Camera Module* video interface will be leveraged to assess a whopping 18 megapixels of raster images. Ideally, this should be able to satisfy the pixels-per-dollar assessment in comparison to other breakout cameras. As hinted to previously, this module is tethered to the Raspberry Pi directly for the fastest processing speed where the real-time image can be visually inspected from the HDMI cable to a nearby monitor of choice.

5.5. User Control Interface

This segment of the project plays a fundamental role of offering user feedback for quality control and accuracy screening. Here, a simple GUI will be designed to read the real-time image from the breakout camera. Here, the user can decide when to capture an image and to what file format to extend the results to. By design, the user has minimal interaction with the actual adjustment of the camera, unless the user desires overriding the autonomous finetuning. If this is the case, the user can either opt to resume the autofocus, or manually adjust the sharpness themselves.

6. Communications / Device Interface Protocols

6.1. Wireless Communications (WiFi)

At the moment, the current design doesn't make use of any wireless communication protocols. However, further exploration beyond the scope of the project would likely lead to less cluttered wiring, with the added caveat of complexity.

6.2. Host Device

The primary mode of the host device is the USB 2.0 standard. At the moment, none of the technology used possesses more advanced mediums. However, this doesn't limit the capacity of the project in any noticeable way, shape, or form.

6.3. Video Interface

Since we plan to implement a visual GUI into the project, an HDMI is planned to be used as a direct tether from the Raspberry Pi to the monitor of choice.

6.4. Device Peripheral Interface

At the moment, there are no definitive protocols which are needed for the scope of the project. However, a direct stream of data is piped directly from the breakout camera directly into the Raspberry Pi B+ with a rolling shutter exposure rate.

Telescope Autofocuser

Camille Watson
Joseph Basdeo
Albin Myscich
Alonna Too-Chiobi

EXECUTION AND VALIDATION PLAN

REVISION – Draft
30 April 2022

EXECUTION AND VALIDATION PLAN

FOR
Telescope Autofocuser

TEAM 6

APPROVED BY:

Project Leader Date

Prof. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
Draft	2/8/22	Telescope Autofocuser Team		Draft Release
Draft	4/30/22	Telescope Autofocuser Team		Edited Validation and Execution

Table of Contents

Table of Contents	I
List of Tables	II
List of Figures	III
1. Milestones	1
2. Validation Plan	1

1. Execution Plan

Subsystem	Timeline
Dev Board/Programming <ul style="list-style-type: none"> 1. Machine Learning Optimization 2. Statistical Image Optimization 3. Stepper-motor Interface 4. Q/A Testing 	<ul style="list-style-type: none"> 1. 2/24/22 - 3/12/22 2. 3/13/22 - 3/30/22 3. 3/31/22 - 4/15/22 4. 4/16/22 - 4/23/22
Power Supply <ul style="list-style-type: none"> 1. Design Charging-Battery Interface 2. Design Redundancy System 3. Charging-Battery Assembly 4. Q/A Testing 	<ul style="list-style-type: none"> 1. 2/24/22 - 3/16/22 2. 3/17/22 - 4/6/22 3. 4/7/22 - 4/15/22 4. 4/16/22 - 4/23/22
Motors <ul style="list-style-type: none"> 1. Design Motor Mounts (CAD) 2. 3D Print Motor Mounts 3. Design Stepper Motor PWM interface 4. Q/A Testing 	<ul style="list-style-type: none"> 1. 2/24/22 - 3/16/22 2. 3/17/22 - 3/18/22 3. 3/19/22 - 4/8/22 4. 4/9/22 - 4/16/22
Imaging <ul style="list-style-type: none"> 1. Design Imager Mount (CAD) 2. 3D Print Imager Mount 3. Image transmit to dev board 4. Q/A Testing 	<ul style="list-style-type: none"> 1. 2/24/22 - 3/16/22 2. 3/17/22 - 3/18/22 3. 3/19/22 - 4/9/22 4. 4/10/22 - 4/17/22

2. Validation Plan

Subsystem	Timeline
Dev Board/Programming <ul style="list-style-type: none"> 5. Machine Learning Optimization 6. Statistical Image Optimization 7. Stepper-motor Interface 8. Q/A Testing 	<ul style="list-style-type: none"> 1. Algorithm sorts images and chooses most focused? 2. Is performance increased by stacking more than 10 images? 3. Can interface move stepper motor interface by a specific margin?

<p>Power Supply</p> <ul style="list-style-type: none"> 5. Design Charging-Battery Interface 6. Design Redundancy System 7. Charging-Battery Assembly 8. Q/A Testing 	<ul style="list-style-type: none"> 1. Is voltage at max after charging battery? 2. Does each component receive operational values without damaging system components? 3. Does each component receive operational values without damaging system components? 4. Does measurements assembly fit within constraints of the telescope?
<p>Motors</p> <ul style="list-style-type: none"> 5. Design Motor Mounts (CAD) 6. 3D Print Motor Mounts 7. Design Stepper Motor PWM interface 8. Q/A Testing 	<ul style="list-style-type: none"> 1. Do dimensions operate within the constraints of the telescope? 2. Are parts the same dimensions as designed? 3. Does motor operate on specific speeds and positions? 4. Does motor operate without damaging the motor mount and assembly subsystem?
<p>Imaging</p> <ul style="list-style-type: none"> 5. Design Imager Mount (CAD) 6. 3D Print Imager Mount 7. Image transmit to dev board 8. Q/A Testing 	<ul style="list-style-type: none"> 1. Do dimensions operate within the constraints of the telescope? 2. Are parts the same dimensions as designed? 3. Does image fit desired image specifications (resolution, pixel width) 4. Can imager receive an image and transmit to Raspberry Pi

Telescope Autofocuser

Joseph Basdeo

Albin Myscich

Alonna Too-Chiobi

Camille Watson

SUBSYSTEM REPORTS

REVISION – 1

30 April 2022

SUBSYSTEMS REPORT
FOR
Telescope Autofocuser

TEAM <06>

APPROVED BY:

Project Leader Date

Prof. Kalafatis Date

T/A Date

Change Record

Rev	Date	Originator	Approvals	Description
1	4/30/2022	Telescope Autofocuser Team	Alonna Too-Chiobi	Final Report

Table of Contents

List of Tables	4
List of Figures	5
Introduction	6
Devboard/Programming Subsystem	7
Subsystem Introduction	7
Subsystem Details	7
Subsystem Validation	8
Subsystem Conclusion	11
Power Supply Subsystem	12
Subsystem Introduction	12
Subsystem Details	12
Subsystem Validation	13
Subsystem Conclusion	15
Motor Subsystem	16
Subsystem Introduction	16
Subsystem Details	16
Motor	16
Motor Driver	17
Raspberry Pi B+	18
Motor Coupler	22
Subsystem Validation	23
Subsystem Conclusion	24
Imager Subsystem	24
Subsystem Introduction	24
Subsystem Details	25
Subsystem Validation	31
Subsystem Conclusion	32

List of Tables

Table 1: Integrated Circuit Specifications (LM2673T-5.0)	12
Table 2: Buck Converter Tested with Li-Ion Battery	12
Table 3: Line Regulation Table of Buck Converter	13
Table 4: Load Regulation of Buck Converter	14
Table 5: Half-step truth table	18
Table 6: Imager Simulation Specifications	23
Table 7: Imager Simulation Results	23
Table 8: Camera Specifications	26
Table 9: 2-Lane MIPI Pinout Reference	27

List of Figures

Figure 1: Original Image of Mars	7
Figure 2: Distorted Image of Mars	8
Figure 3: Predicted Image of Mars	8
Figure 4: Original, Distorted, and Predicted Images of Mars	9
Figure 5: Distorted Versus Predicted Images of Mars	10
Figure 6: Block Diagram of Power Subsystem	11
Figure 7: Multisim Simulation of Buck Converter	12
Figure 8: Line Regulation Graph of Buck Converter	13
Figure 9: Motor Subsystem Functional Block Diagram	15
Figure 10: Torque-Speed Curve	16
Figure 11: Colors of Lead Wires	16
Figure 12: L298N Driver	18
Figure 13: L298N Driver Configuration	18
Figure 14: Half-Step Motor Stator Current Sequence	19
Figure 15: Half-Step Current Vs. Time Diagram	19
Figure 16: Python Script Part 1	21
Figure 17: Python Script Part 2	22
Figure 18: CAD model (left) and 3D printed (right) motor coupler	23
Figure 19: Back Illuminated Structure Diagram	28
Figure 20: Imager Mount 1st Iteration	30
Figure 21: Imager Mount 2nd Iteration	31
Figure 22: Sample Arducam Image in Low Light Intensity	31
Figure 23: Printed Imager Mount	32

1. Introduction

In the sphere of astrophotography, observing natural phenomena are often recorded with digital images which are taken with an extreme zoom via telescope. A common issue centered around low brightness of an image is the result of telescopic zoom and the telescope's large focal ratios; therefore, it is challenging to adapt the focus to achieve sharper images. Observatories often have the advantage of advanced instruments to aid in mitigating this known problem with celestial imaging. However, hobbyists and members of the amateur astrophotography community frequently experience these problems and get low quality images as a result.

For the duration of the project, we plan to completely implement a coupled autofocusing module from which the telescope can achieve optimal sharpness when capturing an image. Since traditional methods of manual focus control offered by telescope producers aren't adequate for quality imaging, we plan to implement more intelligent imaging and focusing methods. The system consists of a pinion and rack focuser which is operated by a set of servos. Moreover, the image raster processing consists of intelligent real-time and post-processing models with the help of machine learning. Images are captured with a camera fastened to an eyepiece. To tie all components together, a cheap development board will be used to process images and optimize focus quality.

2. Devboard/Programming Subsystem

2.1. Subsystem Introduction

At its core, the development board and programming interface play an essential role to the behavior of hardware and software interactions. Since the operational function and design of the hardware is solely dependent on the performance and efficiency of the software interface ensuring a reliable system is needed to enable quality results from the entirety of the machine. The devboard and programming elements of the system are broken down into several parts pertaining to the Raspberry Pi, heuristic scaling (pre-processing), and post processing. Each of these components relies on each other for long-term performance and maintenance quality.

2.2. Subsystem Details

The details of the subsystem are outlined by the number of constraints predetermined by the hardware input/output values, physical limitations, and obstacles detailed in the devboard software.

To begin, the hardware, specifically the motor, operates by taking a positional input in the form of a step count (which equates to a certain degree value). Based on the technical details outlined in the specifications sheet, the motor is expected to operate within a tolerance of $\pm 5\%$ of any given step. Since the operational capacity of the 42BYGHM809 Stepper Motor is only traversing at half-steps, we can expect the tolerance differences to be low, but shouldn't be unaccounted for.

Next, the physical constraints of the hardware-software interfaces primarily encompass the wiring from the Pi to the motor. In total, 11 wires are connected to the motor driver. Here is their breakdown: The 4 motor leads are connected to the 4 output ports of the driver, the 4 driver inputs are connected to GPIO pins on the Pi, the positive and negative wires from the 12V power supply are connected to the 12V and GND input on the driver (respectively), and in order to ensure a common ground between the Pi and driver, another wire connects the GND port of the driver to the GND of the Pi.

Finally, the software limitations are more subtly highlighted by the memory and processor capacity of the Raspberry Pi 3 B+. The CPU specs are able to reach frequencies of 1.4 GHz with a 64-bit quad-core processor. Conversely, the memory One-Time Programmable (OTP) memory block with a full capacity of 8Gb, which can be scaled to larger memory values (i.e. 64Gb, 128Gb, etc.). With these memory limitations, it was important to consider facilitating the size and training location of the machine learning models. Since the Pi itself is physically unable to sustain training DnCNN models with over 500 high resolution images, the models were trained on a remote computer and ported to the raspberry pi for deployment. In particular, Google Drive and Google Colab were used to promote versatility, cloud usability, and data portability among the resources available. Hence, time (processing speed) and memory could be retained in the reasonable manner without needing to risk hardware strain or potential damage from overuse (Pi's are infamous for overheating when the processor is in high use). When considering the sample set of data images collected to represent the planetary bodies, high resolution depictions of mercury, venus, the moon, the sun, Mars, Jupiter, and Saturn were consolidated for model training. The remaining set of

planets were ignored since the telescope is physically unable to view such planets with its focal and magnification power.

The post-processor itself is comprised of a multi-level technique which leverages a set of sample images (training set) to train and validate the DnCNN model. The deep learning model is structured such that it's defined as a sequential SRCNN consisting of a three-stage 2D convolution layers with an adam optimizer possessing a learning rate of 0.0003. After running two-hundred epochs, the following metrics were produced:

```
loss: 2.0566e-04
mean squared error: 2.0566e-04
val loss: 1.5256e-04
val mean squared error: 1.5256e-04
```

Of the metrics collected, these appear to be reasonable, given the quality of the results and metrics calculated in the validation segment.

Please be advised, Dr. Wong requested only a demo and discussion pertaining to the post-processing elements of the project's software aspects. However, if further details concerning the imager's heuristic scaling (pre-processing), please feel free to contact members of the team.

2.3. Subsystem Validation

The subsystem validation procedures mainly pertain to the post-processing elements of the devboard. To break down the validation procedures, an image was taken from a telescope of similar caliber to the telescope used in our project. This original image was taken with an estimated amount of noise that would not disturb or "deviate" the quality of image taken. For an example reference, an image of Mars was taken to comparatively assess the quality of results.

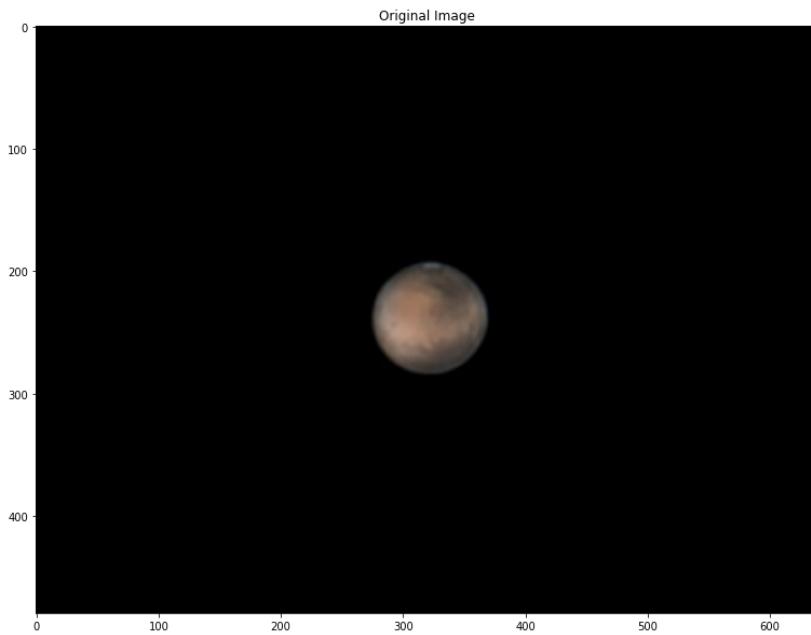


Figure 1: Original Image of Mars

To insinuate a noisy, or blurry image, a distortion was injected into the original image (Mars). This of course would decrease the quality of the original image and simulate a low-resolution, or low quality image.

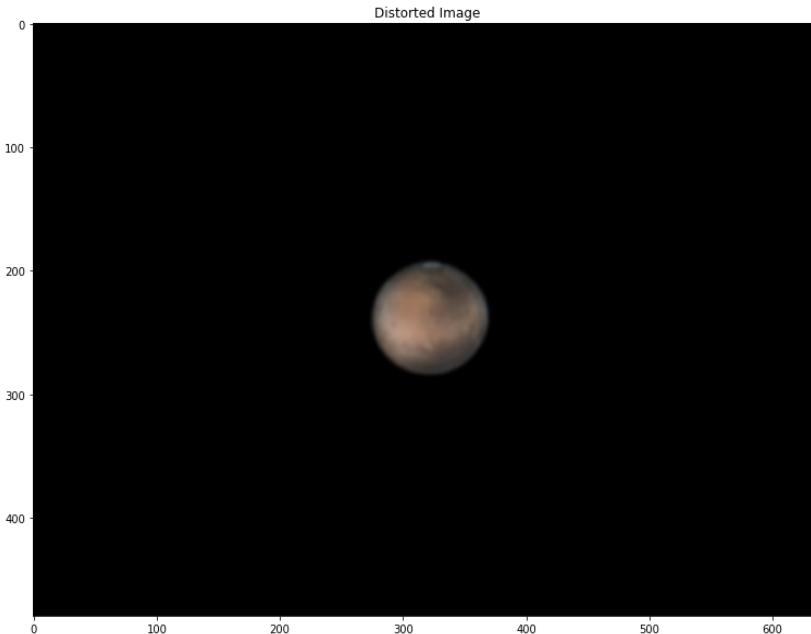


Figure 2: Distorted Image of Mars

Once the deep learning model completed the post-processing over the distorted image, the predicted image was comparatively measured against the original image and the distortion measurement was calculated over the original image as well.

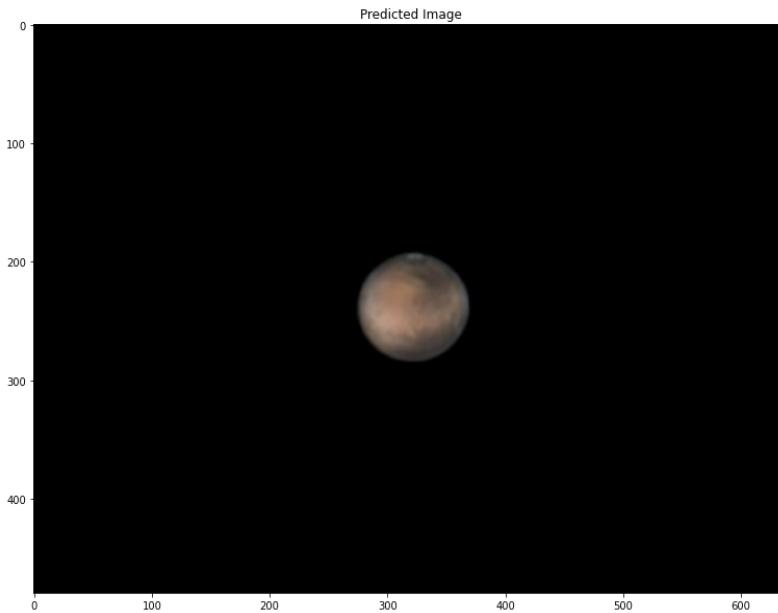


Figure 3: Predicted Image of Mars

Each of these metrics are defined as peak signal-to-noise ratio (PSNR, expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation),

$$PNSR = 10 \log \left(\frac{MAX_I^2}{MSE} \right)$$

mean squared error (MSE, deviation of an estimator measures the average of the squares of the errors),

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

and structural similarity index measure (SSIM, method for predicting the perceived quality of images).

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Each of these measurements are the de-facto measurements for image comparisons. By comparison, we can assess the quality comparisons between each of the images. First, we can observe the original image as possessing a slight blur, where the distorted image contains an obvious amount of noise injected into the image. Finally, the predicated image can clearly represent a similar or higher quality image in comparison to the original. In the instance of Mars, it appears that the more subtle details of the original image were sharpened and as a result, produced a higher quality image. In particular, the darker features of Mars (upper and lower right portions of the planet) are more defined upon close inspection. In a practical sense, we will treat the distorted image as the images that are taken by the astrophotographer.

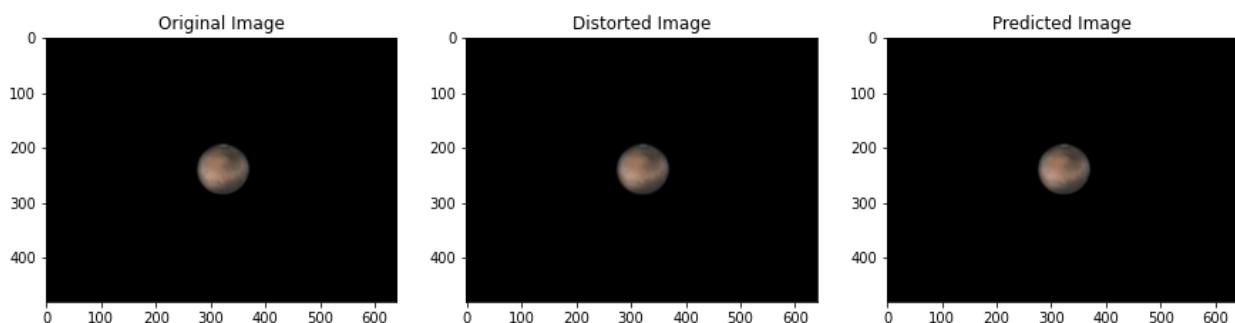


Figure 4: Original, Distorted, and Predicted Images of Mars

As a side-by-side comparison, the original image and predicted image appear to be comparable. Subjectively speaking, the predicted image appears to have clearer features. Qualitatively speaking, the comparisons made entailing the Mars image are as follows

Metrics for original and distorted image

PSNR: 59.2085164

MSE: 0.2340723

SSIM: 0.9995585

Metrics for original and predicted image

PSNR: 62.3994227

MSE: 0.1122689

SSIM: 0.9997594

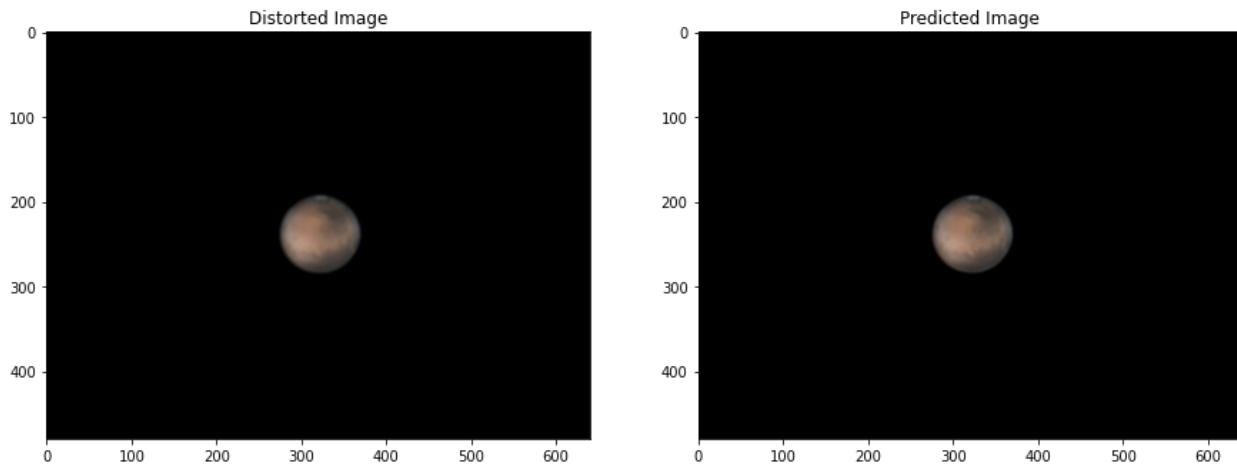


Figure 5: Distorted Versus Predicted Images of Mars

Further validation was assessed with in-person subjects (ECEN 403 class participants, approximately 15 students) where a single student was able to correctly distinguish the original from the predicted among a set of two images (a butterfly and side profile of a brick house).

2.4. Subsystem Conclusion

Of the development board software interactions, post-processing plays a significant role in the system's overall performance and improves the existing techniques involved in the model. Based on the observable results, the post-processing plays a complementary, yet essential role in defining the feature improvements on the imager. Despite the early successes of the model, further improvements to the model can be made as more training data, more sophisticated model architectures, and more model features such as early stopping can be applied to the existing methods employed. Based on the design plan, validation procedures, and results, this particular subsystem appears to have been a stunning success.

3. Power Supply Subsystem

3.1. Subsystem Introduction

The power subsystem is designed to provide regulated power to two outputs, consisting of the Raspberry Pi and the stepper motor driver. The main focus of this subsystem was to not only provide power to the loads, but to ensure that there is a predetermined voltage and regulated current flowing through to the loads. To perform this power management, a buck converter was designed and inserted between the power supply and the two loads. The completed power subsystem was then tested to verify its functionality in terms of voltage capacity, power stability, and consistency.

3.2. Subsystem Details

A functional block diagram of the power subsystem is shown below.

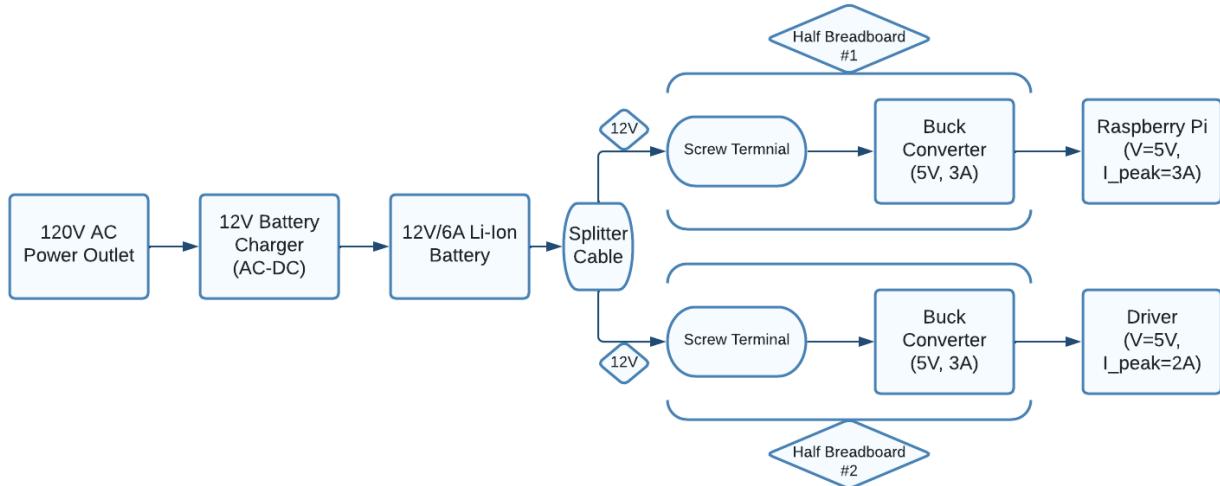


Figure 6: Block Diagram of Power Subsystem

As can be seen from the diagram above, the main components of this system include a Lithium-ion battery connected to two identical buck converters. Between these components, a splitter cable was stripped and inserted into screw terminals to provide the required 12V to each half breadboard.

The design of the buck converters included a voltage regulating integrated circuit with its specifications shown below.

Category	Specification
Function	Step-Down
Output Configuration	Positive
Output Type	Fixed

Input Voltage (Min.)	8V
Input Voltage (Max.)	40V
Output Voltage (Fixed)	5V
Output Current	3A
Switching Frequency	260kHz

Table 1: Integrated Circuit Specifications (LM2673T-5.0)

In addition to the IC detailed above, various components were required to ensure the correct voltage and current values were delivered to the two loads. The complete design of the buck converter constructed using Multisim is depicted below.

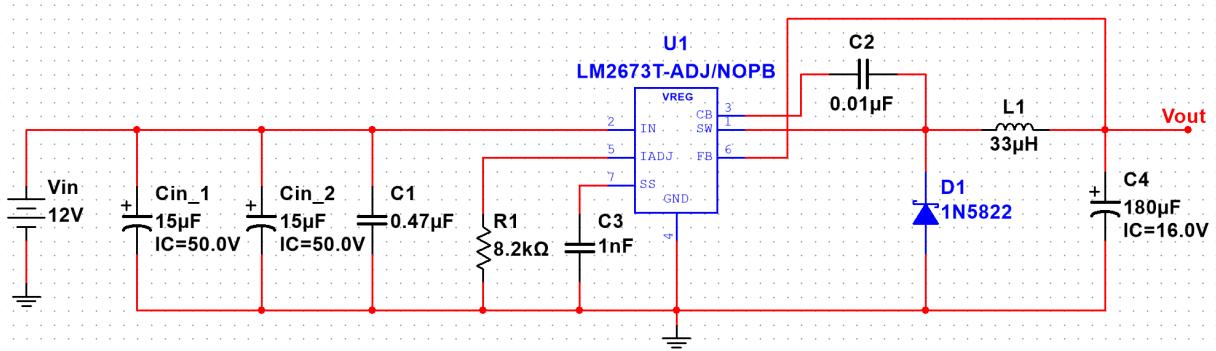


Figure 7: Multisim Simulation of Buck Converter

3.3. Subsystem Validation

The buck converter was first validated by testing the output voltage provided by the 12V Li-Ion battery. Because this system was designed to have the 12V battery as the sole supplier of power, validation for this setup was performed first and measured using the multimeters in lab. As can be concluded from the table below, the buck converter was able to successfully step-down the 12.6V input voltage to the required 5V output voltage.

Circuit	Vin (V)	Vout (V)
#1 (Raspberry Pi)	12.5778	4.9882
#2 (Driver)	12.5777	4.9919

Table 2: Buck Converter Tested with Li-Ion Battery

Next, the line regulation of the system was tested by measuring the output voltage against a range of input voltages. To do this, the lab power supply was utilized to provide a range of input voltages and the lab multimeter was used to measure the corresponding output voltages. The buck converter proved to work as expected against the in-range input voltages (8V - 40V). Although the IC was defined to have a minimum input voltage of 8V, the system was still able to function correctly as long as at least 7V was supplied. A table and graph for these measurements is shown below.

V_{supply} (V)	I_{in} (A)	V_{in} (V)	V_{out} (V)
4	0.004	4.0219	2.3150
5	0.004	5.0251	3.4645
6	0.005	6.0289	4.5213
7	0.005	7.0327	4.9878
9	0.005	9.0401	4.9879
12	0.005	12.0513	4.9879
15	0.005	15.0631	4.9878
18	0.005	18.0522	4.9879
20	0.005	20.0350	4.9880

Table 3: Line Regulation Table of Buck Converter

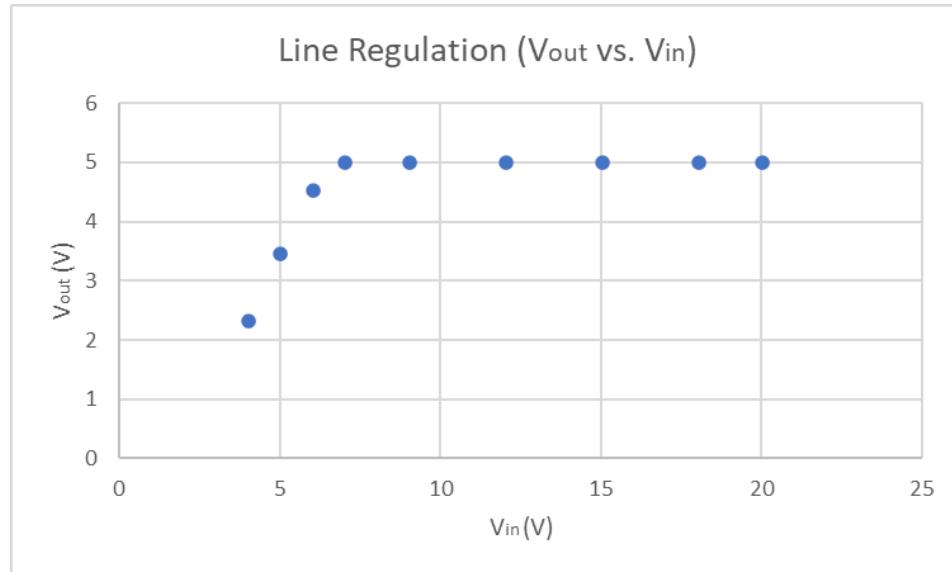


Figure 8: Line Regulation Graph of Buck Converter

Lastly, I tested the load regulation for the buck converter by inserting load resistors with a range of values (50Ω - $8.2k\Omega$) at the output of the circuit. The system's power was supplied using the 12V Li-Ion battery and measured using the lab's multimeter. The load current was then able to be calculated using basic voltage laws and is depicted in the table below.

Resistor Value (Ω)	V_{in} (V)	V_{out} (V)	R_{out} ($M\Omega$)	I_{load} (μA)
50	12.567	4.9872	0.4935	0.1011
100	12.558	4.9894	0.4936	0.1011
1000	12.566	4.9916	0.4939	0.1011
2000	12.571	4.9918	0.4939	0.1011
5100	12.573	4.9921	0.4939	0.1011
8200	12.569	4.9918	0.4939	0.1011

Table 4 : Load Regulation of Buck Converter

Because the battery is defined to have 8.3Ah at 12V, the amount of energy flowing through to the buck converter is 99.6Wh. The estimated lifetime of this battery entails that it would last around 6 days before needing to recharge.

3.4. Subsystem Conclusion

After various testing and validating, the designed power system proved to function as intended. The 12V Li-Ion battery effectively provided power to the two buck converters, which in turn performed their function of stepping down the voltage to a fixed 5V.

Due to the lack of complexity within the buck converter, my goal is to design my own buck converter on a PCB that operates with the same principles as my current design. Taking my knowledge of how this voltage regulating IC operates, I plan to create a similar IC of my own design that will connect to the same outside components. This new buck converter would be finalized before the start of the next semester.

4. Motor Subsystem

4.1. Subsystem Introduction

The motor subsystem is solely responsible for adjusting the focus of the telescope. Depending on the input it receives from the image processor, the motors will make precise clockwise/counterclockwise adjustments. The mounting assembly is responsible for connecting the motor arm to the telescope's focusing shaft and ensuring the position and security of the motor.

4.2. Subsystem Details

This section is organized by explaining the technical specifications of the various functional blocks that make up this subsystem, as well as their configuration and how they interact with each other.

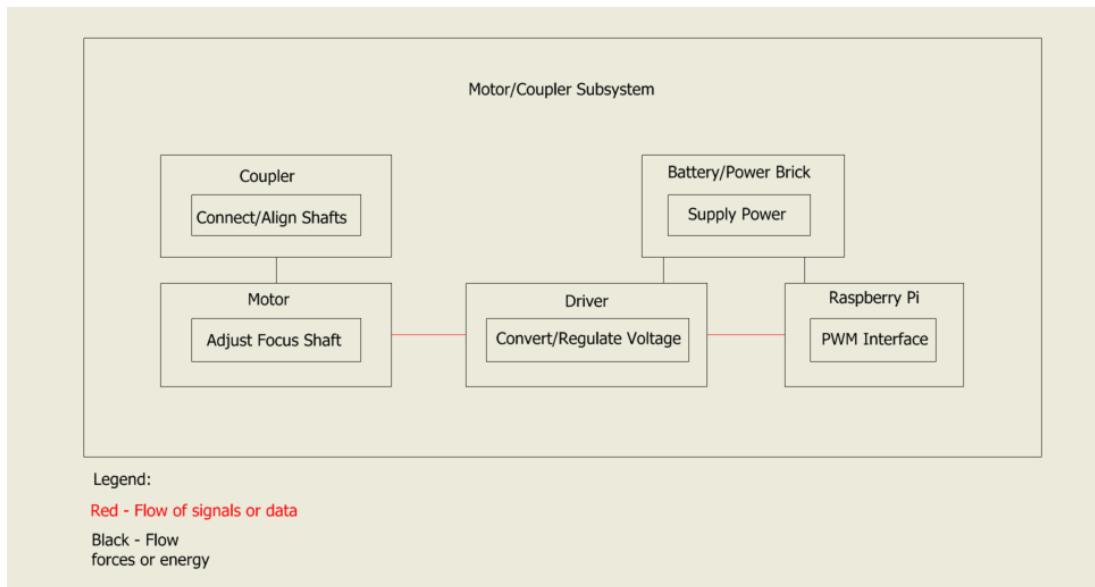


Figure 9: Motor Subsystem Functional Block Diagram

4.2.1. Motor

The motor used for this project is a 2 phase 42BYGHM809 stepper motor with a rated step angle of $.9^\circ$ (which was later reduced to $.45^\circ$ through a half stepping technique) equating to a rated step count of 400 (later increased to 800). This motor follows a NEMA 17 form factor. As you can see from torque-speed curve D below, the pull out torque (or max torque) is achieved in a relatively small time frame which is necessary when trying to obtain a focused image in the shortest amount of time.

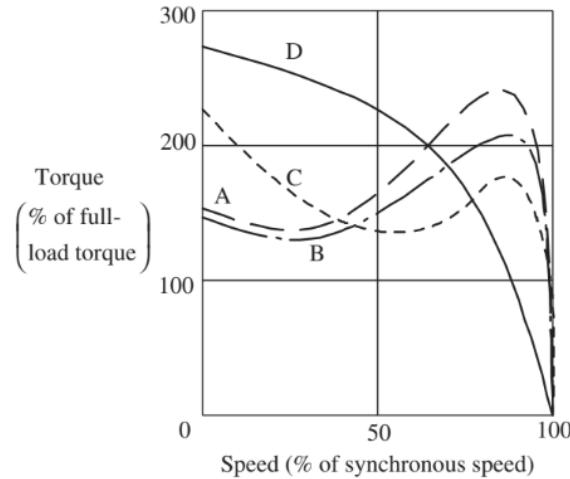


Figure 10: NEMA Torque-Speed Curves from *Electric Machines and Drives-Ned Mohan*

In order to operate this motor, the 4 leads (Black, Green, Red, and Blue) were connected to the output ports on the L298N Motor Driver. The black and green wires direct current to A in the + and - direction, while the Red and Blue wires direct current to phase B in the + and - direction (respectively). Below is an illustration of the phase diagrams as well as the setup of the motor.

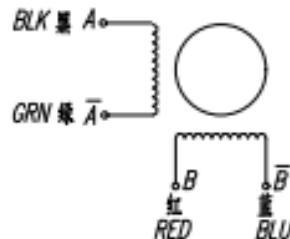


Figure 11: Colors of Lead Wires

4.2.2. Motor Driver

The motor controller used in this project is a L298N dual h bridge driver. This driver is optimal for controlling dc stepper motors and has the capability to drive one 2-phase, one 4-phase, or 2 dc motors. It can provide an input voltage range of 5 - 35 V with a peak current of 2 amps (A).

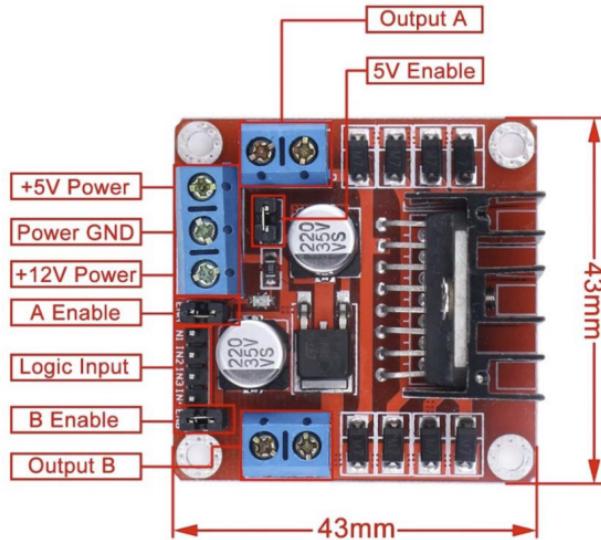


Figure 12: L298N Driver

At a high level, the motor driver acts as the bridge between the stepper motor and the microcontroller. As mentioned earlier, the 4 phase leads of the stepper motor (black, green, red, and blue) are connected to the driver output ports 1, 2, 3, and 4 (respectively). Inputs 1, 2, 3, and 4 of the driver are connected to the general purpose input/output ports (GPIO) of the raspberry pi, specifically GPIO 12, 11, 13, and 15 (respectively). The driver is powered by an external 12V power supply, but has a 5V regulator that ensures the motor isn't receiving a damaging voltage amount.

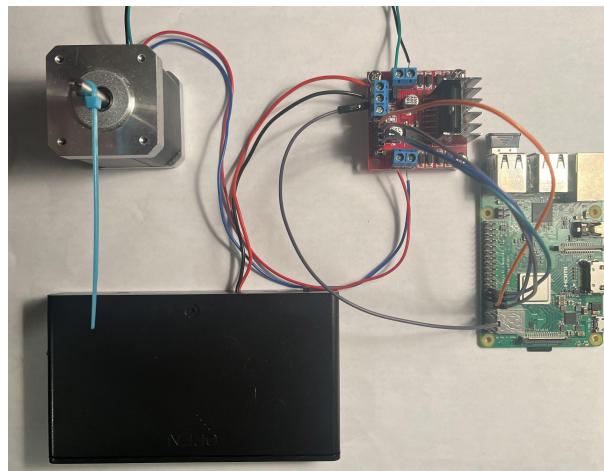


Figure 13: L298N Driver Configuration

4.2.3. Raspberry Pi B+

The brain role of this subsystem was handled by the Raspberry Pi (pi). The specifications of the pi won't be discussed because they aren't important to this discussion; instead, we will discuss the python script that controls the movements of

the stepper motor. In order to achieve a more precise angular positioning of the shaft, a technique called “Half-Stepping” is incorporated in the script. For a full-step drive, the motor driver activates the two coils of the two-phase stepper motor (1 phase at a time) via electric pulses. Each pulse results in the motor moving at the rated step angle. This is a very simple design but results in motor vibrations at low speeds and loud noise. In order to improve this, the half-step drive technique was used. With this technique the driver can send pulses to a combination of 1 or 2 phases at a time enabling the rotor to make a half-step angle and stop in the middle of 2 adjacent full step positions. Compared to the full-step method, the half-step method doubles the precision of the stepper motor by increasing the step count from 400 steps/revolution to 800 steps/revolution resulting in a $.9^\circ \rightarrow .45^\circ$ step angle improvement! Below you'll find an illustration of the motor stator current sequence in a half-step drive configuration as well as it's current vs time diagram and a truth table explain the digital logic.

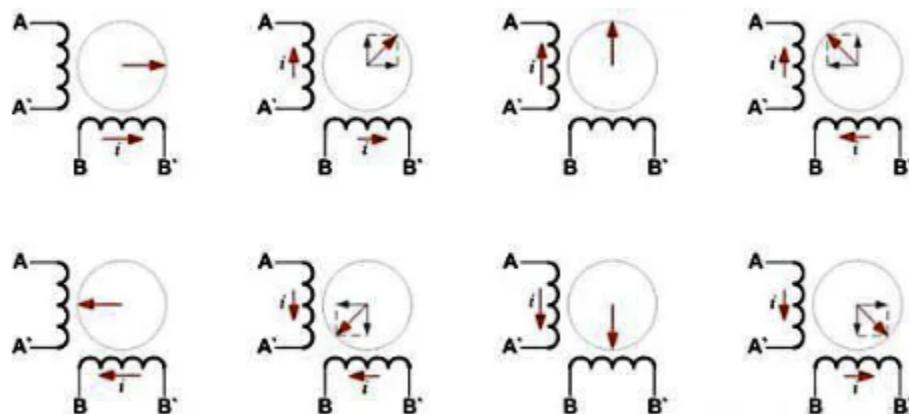


Figure 14: Half-Step Motor Stator Current Sequence

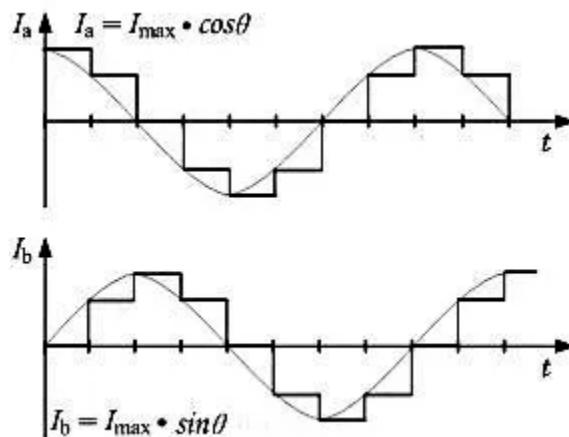


Figure 15: Half-Step Current Vs. Time Diagram

Step	Phase			
	A	B	A-	B-
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Table 5: Half-step truth table

Below you can see figures that contain the portion of the script that controls the excitation of the stepper motor coils when a negative step count is imputed by the user, thus executing the half-step technique. The full script has been omitted due to redundancy.

```

27   try:
28     while(1):      #infinite loop
29       #initially sets all outputs to 0
30       GPIO.output(out1,GPIO.LOW)
31       GPIO.output(out2,GPIO.LOW)
32       GPIO.output(out3,GPIO.LOW)
33       GPIO.output(out4,GPIO.LOW)
34       x = input()
35       x = int(x) #sets the input x as an integer
36       if x>0 and x<=400:    #sets range of input values to be 0 < x <= 400
37         for y in range(x,0,-1):  #starts at value x, and returns to 0 in increments of -1
38           if negative==1:
39             if i==7:
40               i=0
41             else:
42               i=i+1
43               y=y+2
44               negative=0
45             positive=1
46             #print((x+1)-y)
47             #if the value of i = 0 step = 1
48             if i==0:
49               GPIO.output(out1,GPIO.HIGH)
50               GPIO.output(out2,GPIO.LOW)
51               GPIO.output(out3,GPIO.LOW)
52               GPIO.output(out4,GPIO.LOW)
53               time.sleep(0.03)  #suspends execution for .03 seconds
54               #time.sleep(1)
55             #if the value of i = 1 step = 2
56             elif i==1:
57               GPIO.output(out1,GPIO.HIGH)
58               GPIO.output(out2,GPIO.HIGH)
59               GPIO.output(out3,GPIO.LOW)
60               GPIO.output(out4,GPIO.LOW)
61               time.sleep(0.03)
62               #time.sleep(1)
63             #if the value of i = 2 step = 3
64             elif i==2:
65               GPIO.output(out1,GPIO.LOW)
66               GPIO.output(out2,GPIO.HIGH)
67               GPIO.output(out3,GPIO.LOW)
68               GPIO.output(out4,GPIO.LOW)
69               time.sleep(0.03)
70               #time.sleep(1)
71             #if the value of i = 3 step = 4
72             elif i==3:
73               GPIO.output(out1,GPIO.LOW)
74               GPIO.output(out2,GPIO.HIGH)
75               GPIO.output(out3,GPIO.HIGH)
76               GPIO.output(out4,GPIO.LOW)
77               time.sleep(0.03)
78               #time.sleep(1)
79             #if the value of i = 4 step = 5
80             elif i==4:
81               GPIO.output(out1,GPIO.LOW)
82               GPIO.output(out2,GPIO.LOW)
83               GPIO.output(out3,GPIO.HIGH)
84               GPIO.output(out4,GPIO.LOW)
85               time.sleep(0.03)
86               #time.sleep(1)
87             #if the value of i = 5 step = 6

```

Figure 16: Python Script Part 1

```

88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

    elif i==5:
        GPIO.output(out1,GPIO.LOW)
        GPIO.output(out2,GPIO.LOW)
        GPIO.output(out3,GPIO.HIGH)
        GPIO.output(out4,GPIO.HIGH)
        time.sleep(0.03)
        #time.sleep(1)
    #if the value of i = 6 step = 7
    elif i==6:
        GPIO.output(out1,GPIO.LOW)
        GPIO.output(out2,GPIO.LOW)
        GPIO.output(out3,GPIO.LOW)
        GPIO.output(out4,GPIO.HIGH)
        time.sleep(0.03)
        #time.sleep(1)
    #if the value of i = 7 step = 8
    elif i==7:
        GPIO.output(out1,GPIO.HIGH)
        GPIO.output(out2,GPIO.LOW)
        GPIO.output(out3,GPIO.LOW)
        GPIO.output(out4,GPIO.HIGH)
        time.sleep(0.03)
        #time.sleep(1)
    if i==7:
        i=0
        continue
    i=i+1

```

Figure 17: Python Script Part 2

4.2.4. Motor Coupler

Designing a shaft coupler isn't as trivial as it may seem. Many factors go into ensuring that the shaft can provide a smooth transition of torque and power between the motor arm and the telescope's focusing shaft. The main requirements that were considered for this coupler were:

1. Connection of the 2 shafts
2. Proper alignment of centers
3. Smooth torque transmission

In order to meet these requirements a flexible motor coupler was designed. Both ends had different diameters (.18 in and .21 in) to account for the different diameters of the focusing shaft and motor arm (respectively). The hallmark of this design is the helical incision in the center of the coupler.

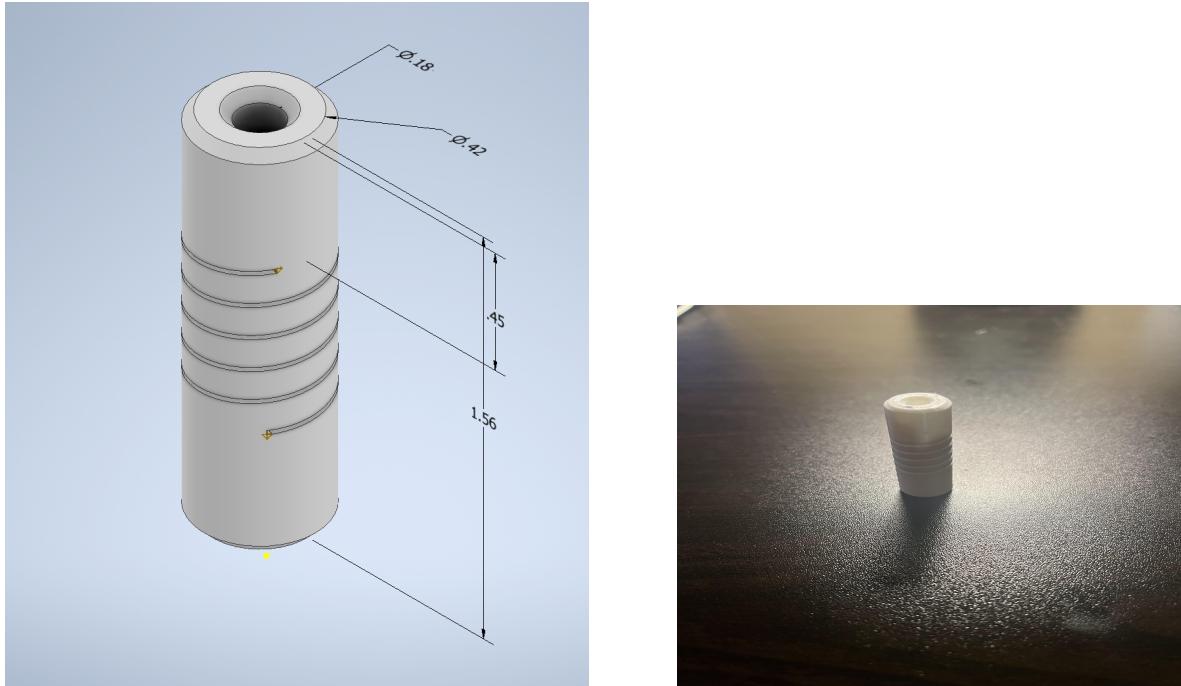


Figure 18: CAD model (left) and 3D printed (right) motor coupler

By including a helical cut in the model, this gives the design flexibility which helps to compensate for misalignment and movement between shafts. This compensation is crucial because perfect alignment of two shafts is extremely difficult and rarely achieved. The better the alignment, the smoother/better the torque/power transfer is between the two shafts.

4.3. Subsystem Validation

The validation criteria defined at the beginning of the semester is found below.

Task	Requirements
Design Motor Mounts (CAD)	Do dimensions operate within the constraints of the telescope?
3D Print Motor Mounts	Are parts the same dimesions as intended?
Design Stepper Motor PWM interface	Does motor operate on specific speeds and positions?
Motor Quality	Does motor operate without damaging the motor mount and assembly subsystem?

Requirement 1: Do dimensions operate within the constraints of the telescope?

This requirement **passed** and was validated through observation.

Requirement 2: Are parts the same dimesions as intended?

To account for the shrinkage of the 3D material, .018in was added to the designed dimensions of the coupler. A comparison of the actual dimensions vs the designed are listed in the table below.

	Designed Dimension	Intended Dimension	Actual Dimension
Focusing shaft diameter	0.18 in	0.16	0.16 in
Motor arm diameter	0.21 in	0.19	0.19 in

As you can see, the actual dimensions of the coupler match the intended dimensions and therefore **passed** the validation criteria.

Requirement 3: Does motor operate on specific speeds and positions?

As seen in the script above, the motor's speed is controlled by the "time.speed()" attribute and its input is based on the desired angle movement. For example, if the motor is required to move +180°, the user would enter an input of 400 and -400 if the motor needed to move -180°. Therefore this validation criteria is **passed**.

Requirement 4: Does motor operate without damaging the motor mount and assembly subsystem?

This requirement was also validated by observing any damage that occurred when the motor coupler was attached to a moving motor shaft. No damage was observed, therefore this validation criteria is **passed**.

4.4. Subsystem Conclusion

Based on the results in section 4.3, we can conclude that this subsystem passed all of its validation requirements. However, improvements can be made, and will be next semester. Some of the improvements we intend to make include:

1. Adding a dampening effect to the script that allows the motor to speed up and slow down gradually as it approaches the desired position. Otherwise known as "microstepping".
2. Add another motor mount to secure the body of the motor to the telescope

5. Imager Subsystem

5.1. Subsystem Introduction

The imager is responsible for viewing images through the eyepiece of the telescope. It then captures an image and feeds the information to the autofocus program subsystem. The imager will continuously transmit data until the current image is determined as "focused." Then the imager will transmit multiple captures of the same image to be processed by the image processing subsystem. It is required that the imager is attached

externally to the telescope via a mounting assembly. The assembly is able to be removed without damaging any of the telescope components, and placement should assist or not get in the way of the other subsystems.

5.2. Subsystem Details

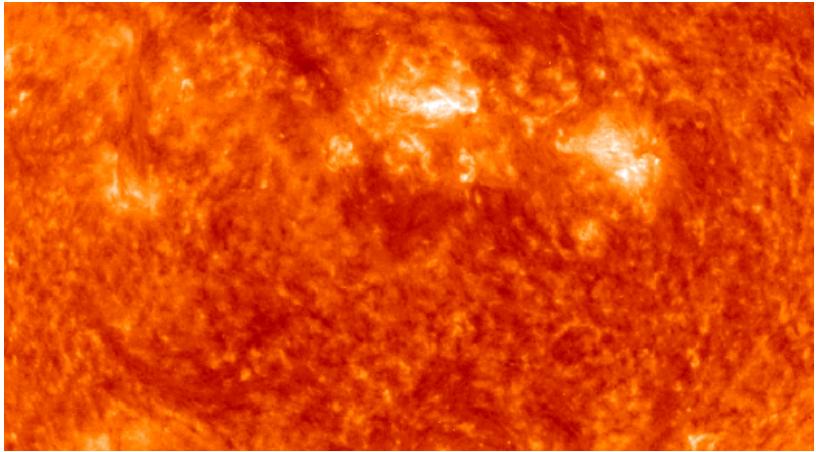
5.2.1. Simulations

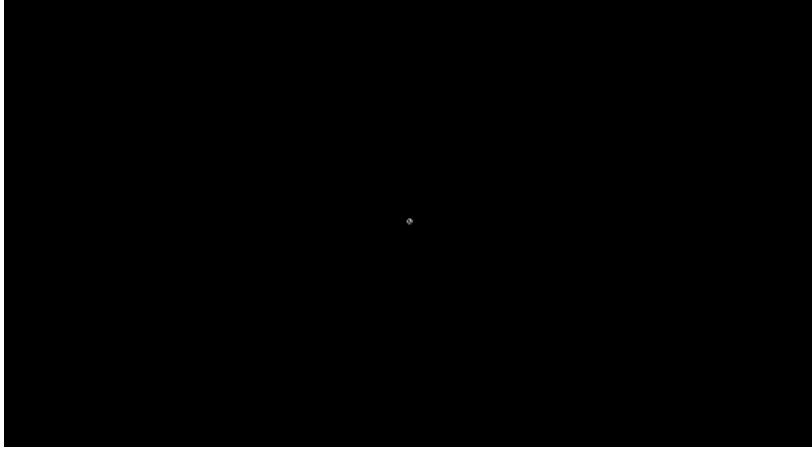
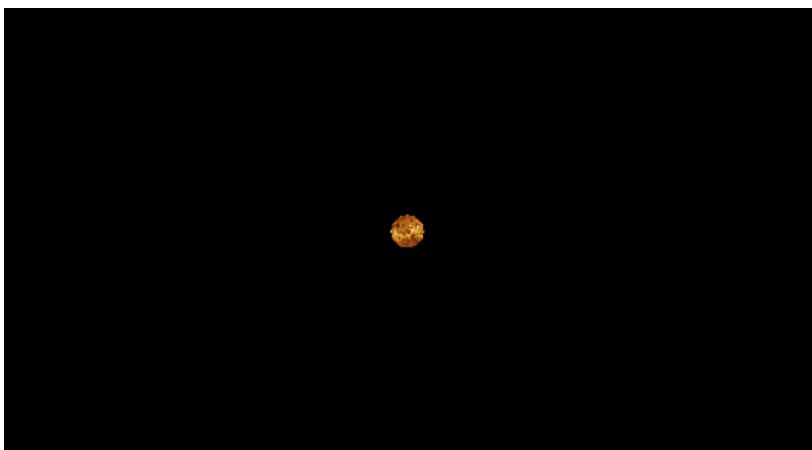
The behavior of the imager when attached to the telescope eyepiece was predicted using a field of view calculator where the specification of both the camera and telescope outputs a mock planetary image across the solar system. Using the calculator, there is the opportunity to test camera specifications and get ideal images that would feed into the dev board programming for autofocus and image post processing. The following images were obtained using the website astronomy.tools/calculators/field_of_view.

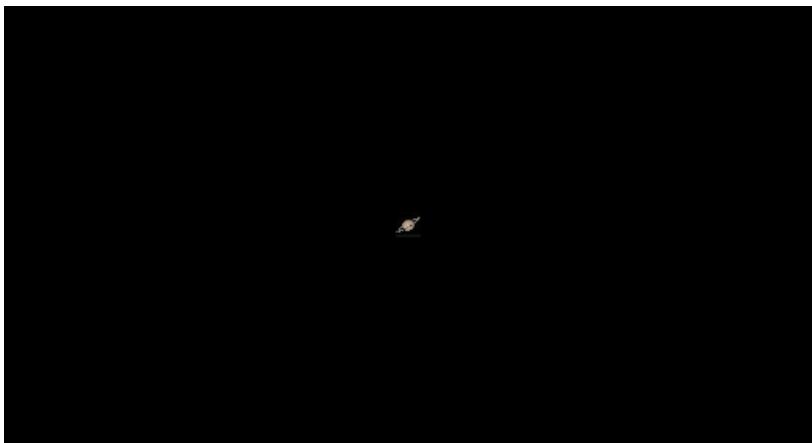
Table 6: Imager Simulation Specifications

Telescope Aperture (mm)	152.4
Focal Length (mm)	762
Resolution (px)	1920 x 1080
Pixel Size (um)	2.9 x 2.9

Table 7: Imager Simulation Results

Space Object	Simulated Image (pre-processed)
The Sun	

Mercury	
Venus	
The Moon	

Mars	
Jupiter	
Saturn	

5.2.2. Camera

The camera chosen for the imager was the Arducam Raspberry Pi Arducam Low Light Pivariety Camera. Using the resolution and pixel size specifications with the

field of view calculator, the simulated images satisfied the quality and size desired before post-processing. The table below lists the different specifications of the chosen camera.

Table 8: Camera Specifications

Image Sensor	2MP IMX462
Resolution (px)	1920 x 1080
Pixel Size (um)	2.9 x 2.9
Max FPS	60
Minimum illumination (lux)	0.1
Interface Type	2-Lane MIPI

The IMX462 cmos sensor uses a back illuminated image structure allowing for the sensor to operate in low light conditions. This is done by placing the photodiodes directly behind the sensor lens instead of behind the sensor circuitry. This structure method shortens the distance the light travels from the source to the photodiodes increasing the sensitivity and allowing visible images under 0.1 lux. Traditionally this cmos sensor structure is used for security purposes, but it can also be applied to astrophotography as the imager will normally operate in low light sensitivity environments.

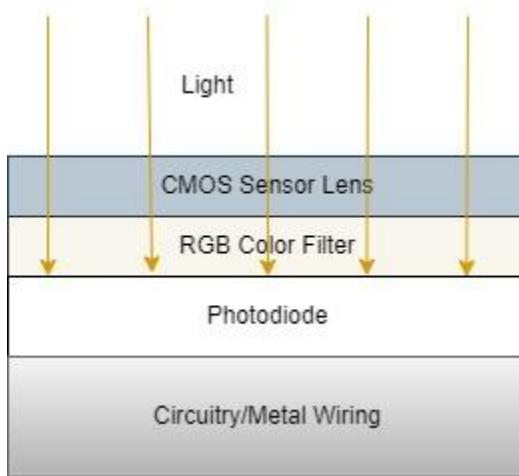


Figure 19: Back Illuminated Structure Diagram

The camera was also chosen due to the ability to interface easily with the project devboard. The arducam camera uses a 2 lane MPI interface that the Raspberry Pi supports and is able to stream the data using the 15 pin out cable. The output voltage needed for the camera is 3.3V, and using the 15 pinout connection cable, the camera is able to be powered directly to the Raspberry Pi without any external power supply. This aided in the decision to go with this sensor as it simplifies the other subsystems such as the power supply

subsystem. The devboard is able to save the current image which is useful in the process of comparing two images when integrating with the motor and programming subsystems. As two images are compared in terms of focus quality, the result will signal to the motor which direction needs to rotate to achieve a more focused image. Below is the table allocating all 15 pins and their description in reference to the Raspberry Pi.

Table 9: 2-Lane MIPI Pinout Reference

Pin #	Description
1	Ground
2	MIPI Data Lane 0 Negative
3	MIPI Data Lane 0 Positive
4	Ground
5	MIPI Data Lane 1 Negative
6	MIPI Data Lane 1 Positive
7	Ground
8	MIPI Clock Lane Negative
9	MIPI Clock Lane Positive
10	Ground
11	Power Enable
12	LED Indicator
13	12C Bus Clock Line
14	12C Bus Data Line
15	3.3V Power Output

5.2.3. Mount

The goal for the mount design was to implement a mounting system without damaging any components of the telescope or any other subsystems. The mount must secure the camera to the telescope eye-piece and ensure there is no obstruction in the process of autofocusing. The first iteration used a shaft collar design that would mount directly to the eyepiece shaft. An additional bracket would then attach directly to the shaft collar directly placing the imager over the eyepiece. The imager would attach to the bracket with the same bolts that are used in the imager assembly. The bracket would attach to the shaft collar using slots that are friction fitted.

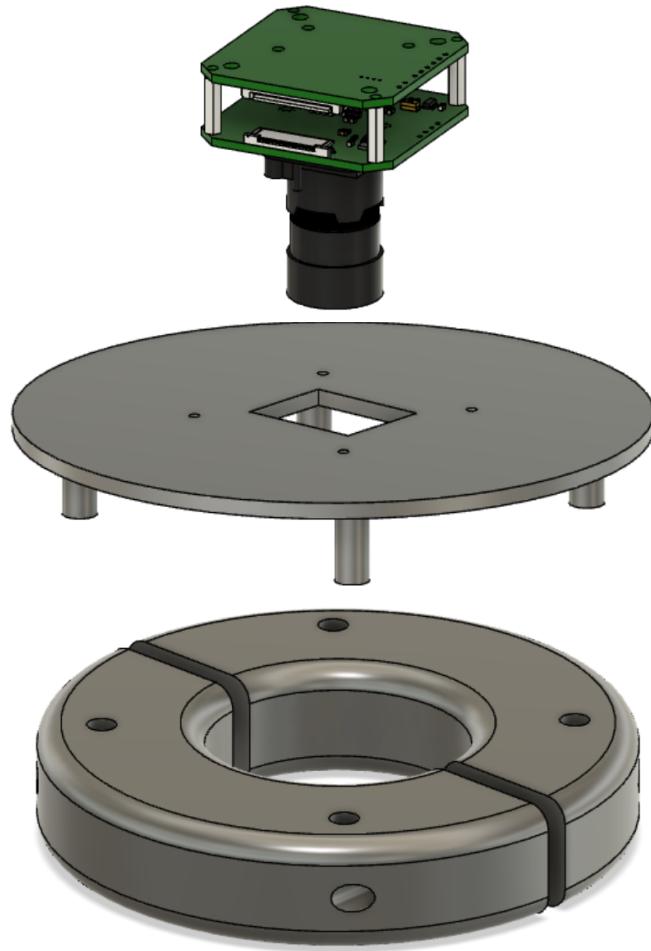


Figure 20: Imager Mount 1st Iteration

The main issue with this design was the image shaft collar was attached to the eye-piece shaft rather than eye-piece lens. When focusing, the telescope rotates a Crawford style focuser that in turn either increases or decreases the length of the eye-piece shaft resulting in a better quality image. Attaching a shaft collar of that size limits the amount of movement that the system can adjust and introduces a scenario that we can not transverse fully to the desired focus. The redesigned mount would attach directly to the eye-piece lens where there is no contact with the shaft whatsoever. This gives full focusing range when implemented with the autofocusing and motor subsystems. The new design reduces the complexity of the mounting assembly as it only consists of one piece instead of two allowing for less chances of failure. The second iteration resembles a similar design to the lens cap of a camera, and fits snugly over the eye-piece lens.

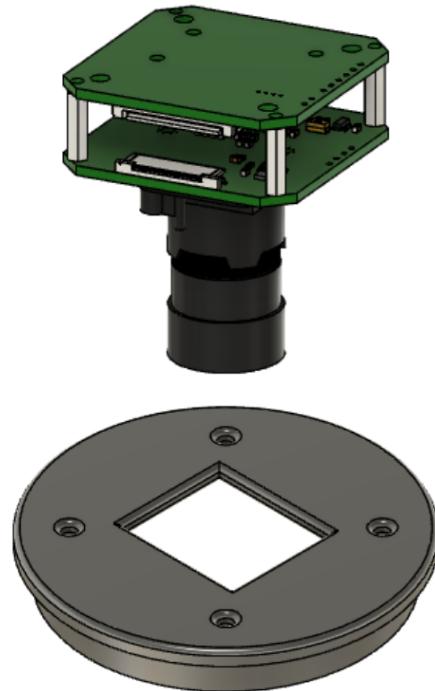


Figure 21: Imager Mount 2nd Iteration

5.3. Subsystem Validation

To test the imager I initially tested the sensor's functionality by interfacing it with the raspberry pi and taking sample images in a low light environment. The imager was able to obtain and transmit image data that fulfilled the specified requirements. The resolution remained as the expected 1920 horizontal by 1080 vertical pixels and was able to operate in little to no visible light. The raspberry pi was able to save the image in a designated folder and could load the image for future comparisons when introduced to an autofocus program. The image can be deleted if it is not deemed the most focused image to ensure there is no overcrowding or limiting of storage.

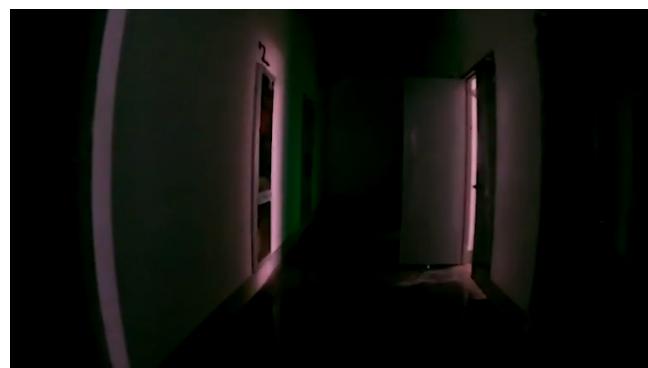


Figure 22: Sample Arducam Image in Low Light Intensity

The 2nd iteration of the mount after printing measured closely to the original cad design. The mount is able to rest on top of the eye-piece and does not affect the eye-piece shaft when the telescope is focusing. The mount does not interfere with any other hardware or the other subsystems and does not damage the telescope during operation.

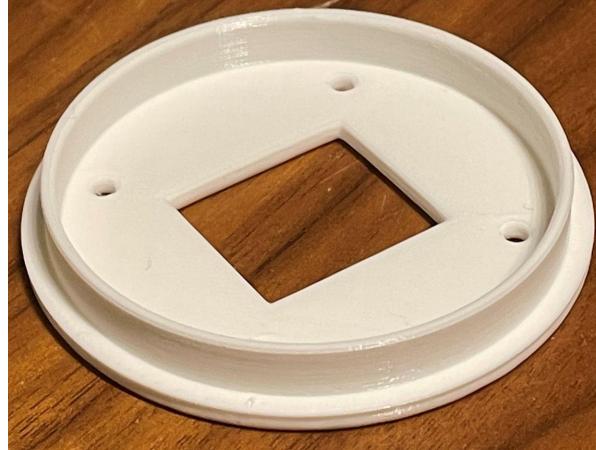


Figure 23: Printed Imager Mount

5.4. Subsystem Conclusion

In conclusion, the imager worked as expected and was able to be utilized by the devboard with little to no additional hassle. The camera comes with libraries that can be used to capture, record, store, load, etc. These functions can be helpful for the future implementation with the motor and programming subsystems when there will be a constant feedback loop of images until the desired focused image is found. The second mount design was more successful than the first as it eliminated complexity with multiple pieces needing to function together, and removed the possible failure of restricting the focuser from achieving the desired image due to interrupting the movement of the eye-piece shaft. The goals for the next stage of the project is to work on implementation with the motor and programming subsystems. Specifically, to capture a focused image of a planetary object and compare to the simulations done previously using the specifications of the telescope and imager. The other goal is to coordinate with the other subsystems on aesthetically combining all components when mounting the systems on the telescope. The components will be mounted on a laser cut panel that is bolted on the telescope rail.