



**UNIVERSITY OF
CENTRAL FLORIDA**

Department of Engineering and Computer Science
CREOL, The College of Optics and Photonics

Dr. Lei Wei, Dr. Samuel Richie, and Dr. Aravinda Kar

Laser Beam Profiler for Measuring Beam Diameter and M^2 Quality

Sponsor: CMS Laser
Mentor: Dr. Ali Abdulfattah

Group 2 Members

Matthew Boyles
Photonic Science and Engineering
mboyles05@gmail.com

Justin Orellana
Electrical Engineering
justinsleven@gmail.com

Jessenia Forney
Electrical Engineering
jesseniaforney@gmail.com

Bryce McIntosh
Computer Engineering
brmc12@gmail.com

Table of Contents

1.0 - Executive Summary	1
2.0 - Project Description	2
2.1 - Motivation	2
2.2 - Project Goals and Objectives	2
2.2.1 - Goals	2
2.2.2 - Objectives	2
2.3 - Requirement Specifications	3
2.3.1 - Core Features	3
2.3.2 - Advanced Features	4
2.3.3 - Stretch Features	5
2.4 - Overall Block Diagram	5
2.5 - Marketing and Engineering Requirements	6
3.0 - Research Related to Project	6
3.1 - Existing Products	6
3.1.1 - Scanning-Slit / Knife-Edge Method	6
3.1.2 - Camera Direct Capture	8
3.2 - Relevant Technology	9
3.2.1 - M2 and Beam Diameter Measurements	9
3.2.1.1 - M2 Measurement	9
3.2.1.2 - Beam Diameter Measurement	11
3.2.2 - Laser Beam Attenuation	12
3.2.3 - Beam Dumps, Traps, and Blocks	14
3.2.4 - Power Supply	15
3.2.4.1 - Battery	15
3.2.4.2 - Voltage Regulators (DC-DC Converters)	21
3.2.5 - Motors and Drivers	25
3.2.5.1 - Motors	25
3.2.5.2 - Motor Drivers	28
3.2.6 - Microcontrollers	30
3.2.7 - OSI Model	34
3.2.8 - Physical Link Layer In Design	34
3.2.9 - Data Link Layer in Design	35
3.2.10 - Communication Mediums	35
3.2.11 - Development Environments	36

3.2.12 - Version Control	37
3.2.13 - 3D CAD Software	37
3.2.14 - Tkinter Python Library	38
3.3 - Component and Part Selection	38
3.3.1 - Laser Attenuator	38
3.3.2 - Beam Dump / Block	42
3.3.3 - Focusing Lens	43
3.3.4 - Optical Telescope	46
3.3.5 - Camera Sensor	47
3.3.6 - Optomechanical Mounts and Holders	48
3.3.6.1 - Kinematic Mounts	48
3.3.6.2 - Lens Mount	49
3.3.6.3 - Iris Diaphragm	49
3.3.7 - Translation Rail	49
3.3.8 - Raspberry Pi	50
3.3.9 - Motor	51
3.3.9.1 - Stepper Motor	51
3.3.9.2 - Motor Driver	52
3.3.10 - Voltage Regulator	57
3.3.11 - Cooling Fans	59
3.3.12 - Microcontroller	62
3.3.13 - PCB Layout and Design	62
4.0 - Engineering Standards and Design Constraints	63
4.1 - Engineering Standards	63
4.1.1 - Optical Standards	64
4.1.2 - Electrical Standards	64
4.1.3 - Software Standards	66
4.2 - Design Constraints	67
4.2.1 - Economic Constraints	67
4.2.2 - Time Constraints	68
4.2.3 - Safety Constraints	68
4.2.4 - Manufacturability Constraints	69
4.2.5 - Sustainability Constraints	70
4.2.6 - Ethical Constraints	71
4.2.7 - Project Requirement Constraints	72
4.2.8 - Supply-Chain Constraints	73
4.2.9 - Design Constraints Comparison Table	73

5.0 - System Design Details	74
5.1 - Optical Design	74
5.2 - Hardware Design	75
5.2.1 - Printed Circuit Board	76
5.2.2 - Chassis Design	80
5.2.3 - Camera Mount	81
5.3 - Software Design	82
5.3.1 - Preparing the Raspberry Pi	84
5.3.2 - Connecting the Laser Profiler to the Network	84
5.3.3 - Server Program	84
5.3.3.1 - Server Side: Motor Driving	85
5.3.3.2 - Server-Side: Auto-Exposure	86
5.3.3.3 - Server-Side: Capturing Images	86
5.3.3.4 - Server-Side: Image Processing	87
5.3.4 Client Program	89
5.3.4.1 - Client-Side: Incoming Image Processing	89
5.3.4.2 - Client-Side: Beam-Parameter Computations, ISO	90
5.3.4.3 - Client-Side: Beam-Parameter Computations, 1/e	92
5.3.4.4 - Client-Side: Socket Thread	92
5.3.4.5 - Client-Side: User Interface	93
5.3.5 - Software Design Overview	95
6.0 - Project Testing	96
6.1 - Optical Testing	96
6.1.1 - System Alignment Testing	96
6.1.1.1 - Lens and Camera Alignment	96
6.1.2 - Beam Attenuation Testing	96
6.1.3 - M2 Beam Quality and Beam Diameter Testing	97
6.2 - Hardware Testing	99
6.2.1 - Power Testing	99
6.2.1.1 - Power Supply	99
6.2.1.2 - PCB	101
6.2.1.3 - Raspberry Pi	102
6.2.2 - PCB Testing	102
6.2.2.1 - TMC2208 (Motor Driver)	104
6.2.2.2 - Microcontroller	104
6.2.3 - Stepper Motor Testing	105
6.2.3.1 - Coils	106

6.2.3.2 - Microsteps	106
6.2.3.3 - Translation Stage	107
6.2.3.4 - Evaluation Kit	107
6.2.4 - Continuity Testing	108
6.3 - Software Testing	108
6.3.1 - Software Testing Environment	108
6.3.2 - Hardware-Software Interface Testing	108
6.3.3 - Network Testing	110
6.3.4 - Measurement Testing	110
6.3.5 - User Interface Testing	112
6.3.6 - Measurement Time Testing	112
7.0 - Project Operation	113
8.0 - Administrative Content	115
8.1 - Budget	115
8.2 - BOM (Bill of Materials)	115
8.2.1 - Optical BOM	116
8.2.2 - Hardware BOM	117
8.2.3 - PCB BOM	117
8.3 - Milestones	119
8.3.1 - Term 1 (Senior Design I)	119
8.3.1 - Term 2 (Senior Design II)	120
9.0 - Conclusion	121
10.0 - Appendices	124
10.1 - References	124

List of Figures

- Figure 1:** Overall Block Diagram for the project.
- Figure 2:** Marketing and Engineering Requirements
- Figure 3:** Diffraction-limited TEM₀₀ gaussian beam intensity distribution. $M^2 = 1$
- Figure 4:** Focused laser beam with perfect beam quality (blue) vs bad beam quality (red).
- Figure 5:** 2D propagation of a collimated beam focusing
- Figure 6:** N-BK7 optical wedge ray trace. Internal reflections represented by the dashed lines do not interfere with reflected or transmitted beams due to the wedge angle.
- Figure 7:** Two optical wedges set up to attenuate an input beam. The residual beam continues through their optical system and the second residual beam is absorbed by a beam dump. Image reproduced with permission from Gentec-EO. All credits for the image go to Gentec-EO.
- Figure 8:** Attenuation by polarization manipulation. Variable attenuation is possible with this design.
- Figure 9:** Commercial beam dump design
- Figure 10:** Buck DC-DC Converter
- Figure 11:** SEPIC
- Figure 12:** Zeta Converter
- Figure 13:** Raspberry PI 3+ B V.2
- Figure 14:** Raspberry PI Camera V2 NoIR
- Figure 15:** Wedge attenuator design used.
- Figure 16:** Transmission of uncoated N-BK7. 3 laser wavelengths are shown by the dashed lines. Transmission from 355 nm - 1100 nm is >90%.
- Figure 17:** N-BK7 reflectance percentage vs angle of incident for 3 wavelengths: 355 nm, 545 nm, and 1064 nm. Graph plotted from public domain source: refractiveindex.INFO database.
- Figure 18:** Sum of the spherical aberrations for 355 nm, 545 nm, and 1064 nm. Wavefront error is in units nm and separated by surface 1 and 2 of the lens.
- Figure 19:** Zemax ray trace for the focusing lens for 355 nm, 545 nm, and 1064 nm using a beam diameter of 12 mm. Total focal shift of 14.94 mm.
- Figure 20:** The Rayleigh length for 355 nm, 545 nm, and 1064 nm as a function of beam diameter before the focusing lens.
- Figure 21:** Graph showing the rayleigh length position along the optical axis plotted for each wavelength and beam diameter.
- Figure 22:** Beam expander diagram with convex and concave lens producing a demagnification of 10.41 times.
- Figure 23:** Plot for each wavelength of the minimum diameter and rayleigh length diameter as a function of input beam diameter before the focusing lens.
- Figure 24:** Block Diagram of TMC2208
- Figure 25:** TMC2208 Pinning Top View
- Figure 26:** Standalone STEP/DIR Driver with OTP Pre-Configuration
- Figure 27:** Filter Suggesting to Prevent ESD Events
- Figure 28:** Efficiency vs Output Current Graph

Figure 29: Input Current versus Output Current Graph
Figure 30: Maximum Change in temperature from ambient given airflow from fan(CMM)
Figure 31: Overall optical design setup using geometric optics. Consists of 3 optical wedges, 2 apertures, beam dump, 3 ND filters, lens, and a camera.
Figure 32: Full camera travel range
Figure 33: Hardware block diagram. More detail is added here from the hardware section of the main block diagram.
Figure 34: Power and Driver Schematic
Figure 35: Microcontroller Schematic
Figure 36: JLC-PCB Circuit Board Layout Final Design
Figure 37: To Scale Representation of Assembled Power System
Figure 38: Blender Model for Camera Mount
Figure 39: Commands and Parameters
Figure 40: Server Program Class Diagram
Figure 41: Image Processing Loop done by the Server0
Figure 42: Algorithm for Finding X Centroid
Figure 43: Laser Beam Profiler User Interface Home Window
Figure 44: Laser Beam Profiler User Interface Data Window
Figure 45: Overall Software Flow Chart
Figure 46: Voltage Regulator Testing
Figure 47: Test Set Up for Oscilloscope
Figure 48: Microcontroller Breadboard
Figure 49: Example Beam-Cross section with background streaking and dust
Figure 50: Beam Waist measurements plotted against the position along the ball screw travel length.
Figure 51: Knife-edge measurements in the x-axis along the focus of the lens.

List of Tables

Table 1: Core features primarily specified by sponsor CMS Laser
Table 2: Advanced features we want to include but are not requirements.
Table 3: Stretch features we could add if we have time and the budget.
Table 4: BP209 VIS Scanning-Slit Beam Profiler key specifications compared with our devices key specifications.
Table 5: M2MS M-Squared Measurement System key specifications vs our devices key specifications.
Table 6: SP932U Beam Profiling Camera key specifications vs our devices key specifications.
Table 7: Types of Primary Batteries
Table 8: Types of Secondary Batteries
Table 9: Specifications of Secondary Batteries
Table 10: Power Output and Price of Battery Packs
Table 11: NEMA-17 and NEMA-23 Comparison
Table 12: Absolute Maximum Ratings for A4988
Table 13: Power Consumptions of Various Arduino Models

Table 14: Power Consumptions of Various Raspberry PI Models
Table 15: Raspberry PI B+ 3 Idle versus CPU Load Power Consumption
Table 16: Specs for the N-BK7 wedge prisms used.
Table 17: Transmittance percentage of 3 ND filters used vs laser wavelength.
Table 18 Raspberry Pi Camera Module v2 useful specs.
Table 19: Other Pins Functions of TMC2208
Table 20: Absolute Maximum Ratings for TMC2208
Table 21: Airflow needed to maintain constant temperature at room temperature
Table 22 : PC Fan Comparison Table
Table 23: Common PEP8 Python styling recommendations
Table 24: Compares the constraints of market technologies with our device's constraints.
Table 25: Python Libraries and Descriptions
Table 26: Wedge attenuation data table
Table 27: Electrical Characteristics
Table 28: Measurement Time Test Results
Table 29: Optical BOM
Table 30: Electrical BOM for Power Supply, Motor and Related Parts
Table 31: BOM for Printed Circuit Board
Table 32: Senior Design I Goal Milestones with dates
Table 33: Senior Design II Goal Milestones with dates

1.0 - Executive Summary

Laser beam profiling is an important step in the laser machining process. Manufacturers such as CMS Laser, use lasers to precision mill parts from industrial to medical utilizations. The lasers used are calibrated to provide extremely precise cuts, engravings, and contoured shapes to a variety of materials. This allows for automation that increases production speed, and accuracy of parts to within a very small degree of error. This in turn helps reduce cost per product, material waste, and achieves uniformity in a way that is not possible with hand milled parts. Because of the precise nature of the products produced, these companies need tools to ensure uniformity of each part that is laser cut.

Due to these constraints manufacturers using laser technologies require a reliable method of measuring laser beam characteristics for optimum precision. This requires the need to measure laser beam size, shape, quality, uniformity and strength to a very precise degree. CMS Laser has provided us with a budget of \$1000 to provide a solution for measuring the characteristics of laser beams used in their machining processes. To solve this problem effectively we must develop a device that can produce accurate, and useful data for beam wavelengths in the range of 355 nm to 1064 nm. This data will need to be displayed, and formatted to show M^2 beam quality, shape, diameter, and power distribution.

Our sponsor has expressed the need for the device to be portable, small and light, while still being precision driven, with micro-adjustments performed by electrical components. This way the beam can be kept at a safe distance, which will prevent damage to the user and equipment. We will accomplish measurement of the beam properties through capturing the intensity distribution along the focus of the beam using a camera. Due to the high intensities of the laser beam used, we are not able to directly measure the beam. We will use a laser attenuator to sample the beam and reduce the power by many magnitudes. Our solution will improve CMS Lasers ability to provide quality products to their customers. We will accomplish this by producing a reliable means to measure laser beam characteristics. The improvement comes through producing a device that can reliably measure a much larger range of detectable wavelengths than any device that they currently possess.

Not only will this help to free up valuable work real estate, but they have a device that is small and portable. This will allow for the device to be taken to different locations if the need arises to measure laser beams. Therefore, they will not have to bring laser devices from sites they are working on to the main manufacturing facility for testing. Allowing for greater flexibility for field testing. This will improve production speed, ensure quality and streamline the company's laser manufacturing process.

2.0 - Project Description

This section will give an overall description of what our project entails. We will discuss the motivation for our project, the objectives and goals, the requirement specifications, overall block diagram, and marketing and engineering requirements.

2.1 - Motivation

Our motivation for this project is to create a device that will be able to measure the beam diameter of a laser beam, as well as the M^2 beam quality. The device needs to be portable and able to be used for many of our sponsor's lasers. CMS Laser is a laser-focused engineering company that supplies custom laser solutions to other companies. The quality of the laser beam being used is important when doing laser processing. This device will provide them with a way to measure important properties of their laser before they use them for processing. We will provide them with an accurate and well-designed device to suit their needs.

2.2 - Project Goals and Objectives

In this section we will go over the goals of this project and what needs to be accomplished for a successful project build. We will also mention the methods involved, as well as ways these methods could be implemented in a successful design. This will set up a means of measuring when we are heading in the right direction moving forward.

2.2.1 - Goals

The goal of this project is to have a fully-functional and portable laser beam profiler. This device should be able to be put directly in the beam path and will also include an alignment system to assure that the beam is correctly aligned through the system. The beam power will need to be reduced. The device needs to work for a large range of wavelengths (355 nm - 1064 nm). The intensity distribution cross-section of the beam will be displayed, as well as the diameter and M^2 beam quality.

2.2.2 - Objectives

The method for capturing the intensity distribution cross-section of a laser beam for this device is by using a CCD/CMOS camera. This method will allow for high resolution capturing of the intensity distribution. We need to find a camera that will work for as many of the wavelengths (355 nm - 1100 nm) as possible. This device should be able to measure laser beams up to 100 W. To achieve this, the laser beams will need to be highly attenuated before being incident on the camera sensor. To reduce the power of the laser beam, optical wedges followed by a series of ND filters will be used to bring the power low enough to capture the

beam profile. The beam will be focused to a point and measured. To measure the M^2 beam quality, 10 points along the optical axis of the focused beam will be imaged and the minimum beam waist and divergence angle measured. From the focal length of the lens used and the divergence angle, the beam diameter of the original beam can be found.

The sensor will be mounted to a translation stage that will move along the optical axis and be powered by a stepper motor. This will ensure repeatability of scan and accuracy that would not be possible if the translation was made by hand. The stepper motor will be controlled by a motor driver which will take its instructions via a 5V logic link with our microprocessor. The motor driver will take power from our power supply once voltage is stepped down through the voltage regulator. Which will ensure a constant voltage to be pumped through to the driver as the battery discharges. The driver and the microprocessor will then communicate with each other to control the stepper motors.

In order to display the intensity distribution cross-section of the beam, the sensor will be sampled using an ADC, and the data will be stored in the microcontroller. The microcontroller will talk to the client program on the PC using the interface provided by a proprietary device driver. Captures will be triggered through the client program, and the command will be sent to the device to begin a capture. Once the capture data is stored, it will be sent back to the client program for additional processing and display. The interface should be intuitive to use, and all functionality and display will be labeled accordingly.

2.3 - Requirement Specifications

Below are the specifications in a chart that will need to be achieved to reach our project's objectives and gauge our success as we work towards our goals. We have included along with this list advanced and stretch features we will strive to achieve in excess of our current primary objectives.

2.3.1 - Core Features

The core features for this device, shown in **Table 1**, are requirements specified by CMS Laser. The sponsor wants the device to work for a laser wavelength range of 355 nm to 1100 nm, with beam diameters of 3 mm to 12 mm. This device should also work for lasers up to a max average power of 100 watts. This device will need to provide a user interface for a Windows 10 PC through Bluetooth connection. In this interface the M^2 beam quality and beam diameter will be displayed as well as the intensity distribution along the optical axis of the focused beam. The device will be powered by a 12 V rechargeable battery, to keep the device portable. No specific dimensions or weight for the device were specified, so we limited the device to a 70 cm x 70 cm x 20 cm volume and under 20 lbs.

Wavelength range	355 nm - 1100 nm
Beam entrance diameter	3 mm - 12 mm
Laser power	Up to 100 W
Laser Type	CW and Pulsed
Imaging	Capture and record intensity distribution
Measurements	Beam diameter and M ² beam quality
Measurement method	Camera direct capture
Measuring accuracy	75%
Size	70 cm x 70 cm x 20 cm
Weight	< 20 lbs
Power Supply	12V Battery, SMPS
Computer interface	Windows 10 PC
Computer connection	LAN/WLAN
Budget	\$1000

Table 1: Core features primarily specified by sponsor CMS Laser

2.3.2 - Advanced Features

The advanced features of this device found in **Table 2**, include a larger range of measurable wavelengths (355 nm - 10600 nm). A possibility for an automated beam alignment through the system to help the user and save time. Some cooling fans could also be added to help regulate the temperature inside the device due to the absorption of light from the walls and the optics.

Wavelength range	355 nm - 10.6 um
Laser Type	CW and Pulsed
Automation	Beam alignment
Measuring accuracy	85%
Temp. Control	1 - 2 Cooling fan

Table 2: Advanced features we want to include but are not requirements.

2.3.3 - Stretch Features

This section contains the stretch features of the device. **Table 3** shows some of these features. If time allows, we will try to incorporate these into the device.

Measuring accuracy	95%
Laser power	5 W - 300 W
Automation	Auto-alignment for beam
Measurements	Calculate laser power
Computer Connection	USB

Table 3: Stretch features we could add if we have time and the budget.

2.4 - Overall Block Diagram

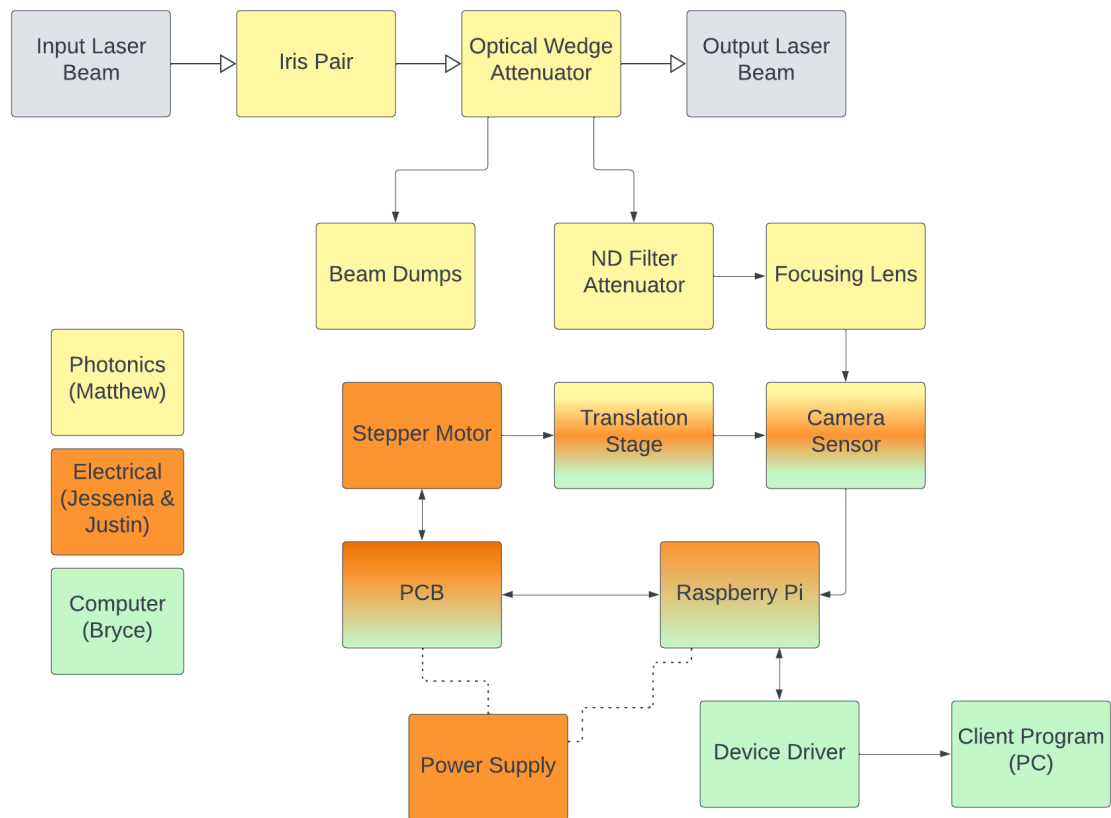


Figure 1: Overall Block Diagram for the project.

2.5 - Marketing and Engineering Requirements

This matrix compares how the marketing requirements and engineering requirements impact each other. Each requirement has either a positive or negative polarity. This means that either increasing or decreasing the requirement will increase the desirability of the product. The upward arrows show a positive correlation between what is in the x axis and the y axis. The downward arrows show a negative correlation. The more arrows the stronger the correlation.

			Engineering Requirements					
Marketing Requirements			Accuracy	Power Supply	Size/Weight	Cost	Time	Data Analysis
			+	-	-	-	+	+
	Laser Power	+	↑	↓↓	↓	↑↑	↑↓	↑
	Measurements	+	↑↑	↑	↓↓	↓	↑↑	↑↑
	Accuracy	+	↑↑	↓↓	↑	↑	↑↑	↑↑
	Cost	-	↓	↑	↑↑	↑↑	↓	↓↓
	Wavelength Range	+	↑	↓↓	↓↓	↑↑	↓↓	↑↑
	Portable	+	↓↓	↑↑	↑↑	↑↑	↑	↓↓
	Max Beam Diameter	+	↑	↓↓	↑	↑	↓	↓
Target for Engineering Requirements			75%	Battery (Portable)	Portable	<\$1000	6 months	Fast Processing
Legend: Correlation			Legend: Polarity					
↑↑	Strong Positive		+	Positive				
↑	Positive		-	Negative				
↓	Negative							
↓↓	Strong Negative							

Figure 2: Marketing and Engineering Requirements

3.0 - Research Related to Project

This section will discuss all the research done when designing our device. It includes existing products, relevant technology, and part selection.

3.1 - Existing Products

The method of profiling a beam is a well-known method and is important when using lasers for high precision processes including, laser material processing and medicine. With this being a device needed by many industries to measure important characteristics of their lasers, it is a device easily obtained with a broad range of specifications and methods.

3.1.1 - Scanning-Slit / Knife-Edge Method

The scanning-slit / knife-edge method consists of a rotating drum with narrow slits. Inside the drum is a photodetector. This method is able to profile the beam creating multiple data arrays across the beam cross-section. As the slit / knife-edge scans the beam, the fraction of the laser light is incident on the

photodetector. Only 2 axes of data are required to reconstruct a 2D representation of the beam intensity profile. The resolution of this system depends greatly on the quality of the slit / knife-edge. This technique is not good when measuring multimode beams that contain multiple peaks. This is due to laser power that is incident on the power detector is all averaged together and saved to that location.

One device that uses this method is the BP209 Scanning-Slit Beam Profiler offered by Thorlabs. The performance of this device can be seen in **Table 4**.

Key Specifications		
Device	BP209 VIS	Our Device
Wavelength	200 nm - 1100 nm	355 nm - 1100 nm
Method	Scanning Slits, Knife Edge	Camera Direct Capture
Beam Diameter Range	2.5 μ m - 9 mm	3 mm - 12 mm
Power IN Range	1 μ W - 10 W	5 W - 100 W
Resolution	0.12 μ m - 1.24 μ m	1.12 μ m
Size	Ø79.5 mm x 60 mm	70 cm x 70 cm x 20 cm
Weight	1.10 lbs	<20 lbs
Price	\$4,594.19	<\$1,000

Table 4: BP209 VIS Scanning-Slit Beam Profiler key specifications compared with our devices key specifications [36]

Our device will provide measurement for larger beam diameters than the BP209 as well as a much higher max power. These are common for laser processing as they would use a high NA and power to machine. The BP209 lacks the ability to measure the M^2 beam quality without the assistance of the M2MS M-Squared measurement System (**Table 5**). With both combined, their device will weigh more but will have a smaller size. The total price compared to our device will be >10x greater.

Key Specifications		
Device	M2MS	Our Device
Wavelength	400 nm - 2700 nm	355 nm - 1100 nm
Lens Focal Length	250 mm	250 mm
Beam Diameter Range	2.5 μ m - 9 mm	3 mm - 12 mm
Power IN Range	1 μ W - 10 W	5 W - 100 W
Resolution	0.12 μ m - 1.24 μ m	1.12 μ m
Size	300 mm x 175 mm x 109 mm	70 cm x 70 cm x 20 cm
Weight	21.34 lbs	<20 lbs
Price	\$6,366.63	<\$1,000

Table 5: M2MS M-Squared Measurement System key specifications vs our devices key specifications [37]

The M2MS M-Squared Measurement System consists of multiple focusing lenses that can be switched out followed by 2 mirrors. The mirrors are mounted to a translation stage which translates where the beam is incident on the camera. This allows for a more compact design due to the mirrors reflecting the beam by 180°. Our device will not be as compact, but has a similar design using reflections from wedges inside of the device. The focal length for our device will be the same as the one used in the M2MS. Due to ours using a single focusing lens, we are not able to measure beams with a diameter smaller than 3 mm.

3.1.2 - Camera Direct Capture

The SP932U Beam Profiling Camera uses the same method of direct capture as our device. This method is great for displaying the intensity distribution live. The key specifications can be seen in **Table 6**.

Key Specifications		
Device	SP932U	Our Device
Wavelength	190 nm - 1100 nm	355 nm - 1100 nm
Method	Camera Direct Capture	Camera Direct Capture
Beam Diameter Range	34.5 μ m - 5.3 mm	3 mm - 12 mm
Power IN Range	-	5 W - 100 W
Resolution	3.45 μ m	1.12 μ m
Size	-	70 cm x 70 cm x 20 cm
Weight	-	<20 lbs
Price	\$5,609.00	<\$1,000

Table 6: SP932U Beam Profiling Camera key specifications vs our devices key specifications [38]

The SP932U is more sensitive in the UV range. Our device makes up for that with having a larger max beam diameter and higher resolution, as well as being much more inexpensive.

While these market technologies provide a good range of specifications, what they are all missing is a large beam power range. This is due to these devices not including any beam attenuation. This is a bonus for our device because it will allow the user to directly image the beam without having to worry about attenuating the beam. The following custom device is very similar to ours and uses a camera for imaging the beam.

A portable beam profiler was made and described in the paper, “Automated translating beam profiler for *in situ* laser beam spot-size and focal position measurements” [40]. Their device, much like ours, is able to measure the beam quality by scanning the laser beam focus. Their device uses an older version of the Raspberry Pi camera, making it sensitive to 300 nm - 1100 nm with good

resolution. It also uses a commercial translation stage from Thorlabs that has a 25 mm travel distance. The focusing optics and beam attenuation are not included in their design. While this allows for the user to use any lens that they want, it does mean there is more user alignment needed. Their device is small, compact, and easy to place in the beam path. The computation for their device consists of taking axial scans, and fitting to parabolic function. They use the beam diameter definition of $1/e^2$, which is only valid when measuring gaussian single mode beams. They mention that they would eventually add the $D4\sigma$ beam width measurement to comply with the ISO 11146 standard.

Our device will unfortunately be larger than this custom device, but mainly due to the beam attenuation and focussing optics being included in the device. We will have the same resolution but a much larger travel range. Our 100 mm travel distance allows for measuring a large range of beam diameters and will only need alignment into the system to do so. Our device will also use the ISO standard beam waist and parabolic fit to find the M^2 beam quality.

3.2 - Relevant Technology

This section will cover the research done on technology that we will be using for our device. This includes M^2 and beam diameter measurements, laser beam attenuation, beam dumps, traps, and blocks, power supply, motors and drivers, microcontrollers, and PCB layout. This research will allow us to have a better understanding of what to choose for our device to have the best performance for the laser beam profiler.

3.2.1 - M^2 and Beam Diameter Measurements

This section will cover what the M^2 beam quality factor is and how we will measure it. We will also discuss components that may interfere with the M^2 factor leading to an inaccurate measurement of the beam quality.

3.2.1.1 - M^2 Measurement

The M^2 beam quality factor measures how tight a beam can be focused. As previously discussed, this is important for laser processing to ensure that the beam intensity distribution is correct and as high-quality as possible. The highest quality beam occurs when $M^2 = 1$. This is for a TEM_{00} diffraction-limited gaussian beam. **Figure 3** shows the cross section of a diffraction-limited gaussian beam with the intensity profile across the x-axis and y-axis. The beam waist for a gaussian beam profile is generally determined by the distance from the max intensity to $1/e^2$ the max value. This is around 13.5% of the max value. The area inside the red dashed circle is the beam spot size using this method.

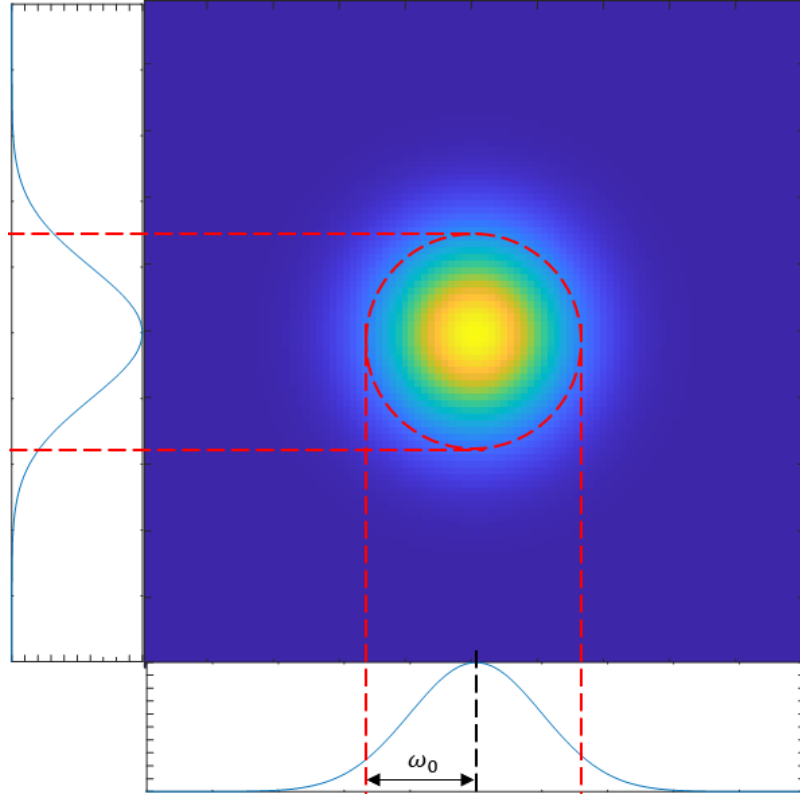


Figure 3: Diffraction-limited TEM₀₀ gaussian beam intensity distribution. $M^2 = 1$

The desired M^2 value is as close to 1 as possible since having an M^2 value of 1 is impossible. The effect of an M^2 value greater than 1 can be seen in **Figure 4**. The far-field divergence angle θ is the same for both $M^2 > 1$ and $M^2 = 1$. This is because the divergence angle depends on the focal length and beam diameter. A larger M^2 value will result in a larger spot size and longer Rayleigh length. As a result, the M^2 beam quality determines the max intensity able to be produced at the focus, as well as how tightly the laser beam can be focused.

The M^2 value using the $1/e^2$ beam waist definition is only sufficient and accurate for determining the beam quality for single mode laser beams. Single mode laser beams, like **Figure 3**, have a single peak value. Multimode laser beams may have multiple peaks making it inaccurate to calculate the waist from the max intensity to $1/e^2$ the max intensity. The ISO 11146-1 standard states that the beam width should be calculated using the second-moment width($D4\sigma$). This is defined as 4 times the standard deviation across both the x and y axis. The computation for this uses the intensity profile cross-section of the laser beam. More on this is explained in section 5.3.4.2 - *Client-Side: Beam-Parameter Computations*.

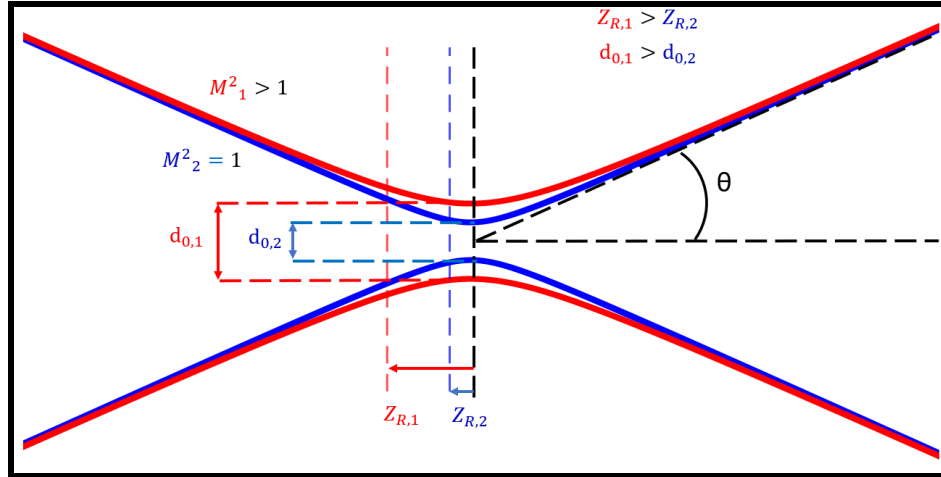


Figure 4: Focused laser beam with perfect beam quality (blue) vs bad beam quality (red).

3.2.1.2 - Beam Diameter Measurement

The beam diameter measurement is a simple measurement that can be done once the minimum waist location and size is found. The following equation can be used to find the beam diameter at any location along the focus:

$$\omega(z) = \omega_0 \sqrt{1 + \left(\frac{\lambda z}{\pi \omega_0^2} \right)^2} = \omega_0 \sqrt{1 + \left(\frac{z}{Z_R} \right)^2} \quad (1)$$

Where $\omega(z)$ is the waist as a function of axial position z , ω_0 is the minimum beam waist, λ is the wavelength, and Z_R is the Rayleigh length. The beam diameter before the focusing lens can be calculated using this equation when z is equal to the focal distance of the lens at the laser wavelength. **Figure 5** shows the beam path after exiting the lens. 3 intensity graphs are shown for 3 axial positions along the focus.

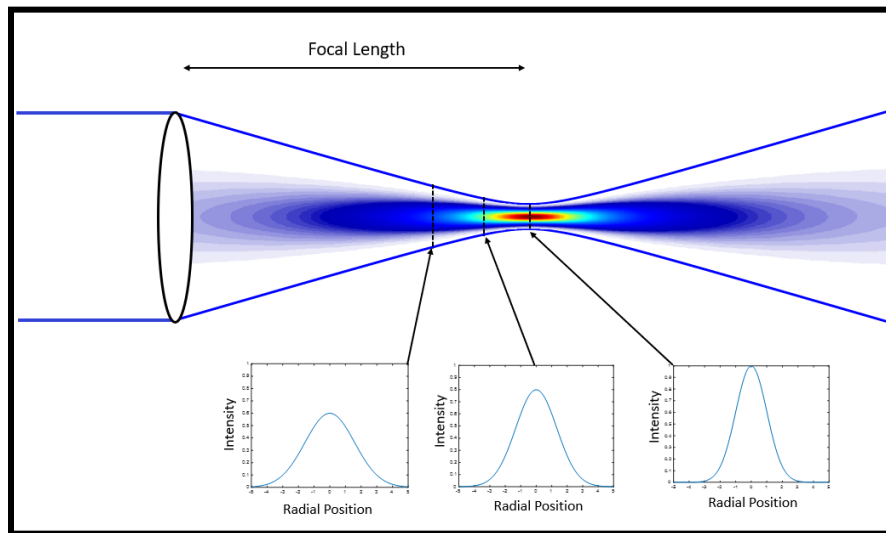


Figure 5: 2D propagation of a collimated beam focusing

3.2.2 - Laser Beam Attenuation

Laser beam attenuation is an important part when profiling a beam. Due to the sensitivity of most sensors and detectors, even laser powers of only a couple watts will saturate or permanently damage it. Some common attenuation techniques will be discussed and compared.

One popular method of laser attenuation consists of using the front reflections of an optical wedge. Both surfaces of an optical wedge are planar, with one being at a small angle. This is to ensure that the internally reflected beams do not interfere with the front surface reflection and cause a fringe pattern or optical feedback. A diagram of an optical wedge can be seen in **Figure 6**.

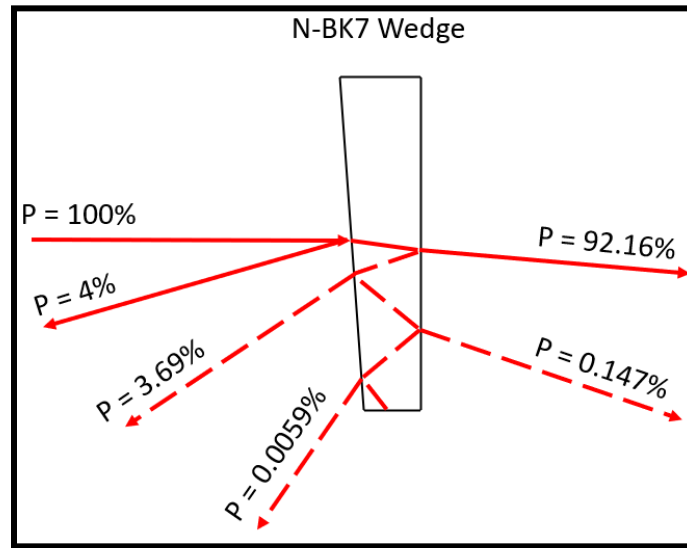


Figure 6: N-BK7 optical wedge ray trace. Internal reflections represented by the dashed lines do not interfere with reflected or transmitted beams due to the wedge angle.

The percent of the beam that is reflected depends on the refractive index difference between the 2 surfaces, angle of incidence, and polarization. The Fresnel reflection equations for s-polarization (R_s) and p-polarization (R_p) are given as:

$$R_s = \left| \frac{n_1 \cos(\theta_i) - n_2 \cos(\theta_t)}{n_1 \cos(\theta_i) + n_2 \cos(\theta_t)} \right|^2 \quad (2)$$

$$R_p = \left| \frac{n_1 \cos(\theta_t) - n_2 \cos(\theta_i)}{n_1 \cos(\theta_t) + n_2 \cos(\theta_i)} \right|^2 \quad (3)$$

Where n_1 is the refractive index of the first medium, in most cases air, n_2 is the refractive index of the wedge material, θ_i is the angle of the incident ray with respect to the plane of incidence, θ_t is the angle of the transmitted ray with

respect to the plane of incidence. When $\theta_i = \theta_t = 0$, then the reflectance can be simplified to:

$$R = \left| \frac{n_1 - n_2}{n_1 + n_2} \right|^2 \quad (4)$$

Typical wedge attenuators consist of 2 wedges for increased attenuation as well as to preserve polarization. BA16-60S by Gentec-EO is an example of a wedge attenuator. **Figure 7** shows the design of this attenuator with a beam path. If AR coatings are put on the wedges, then attenuation of approximately 0.01% can be achieved.

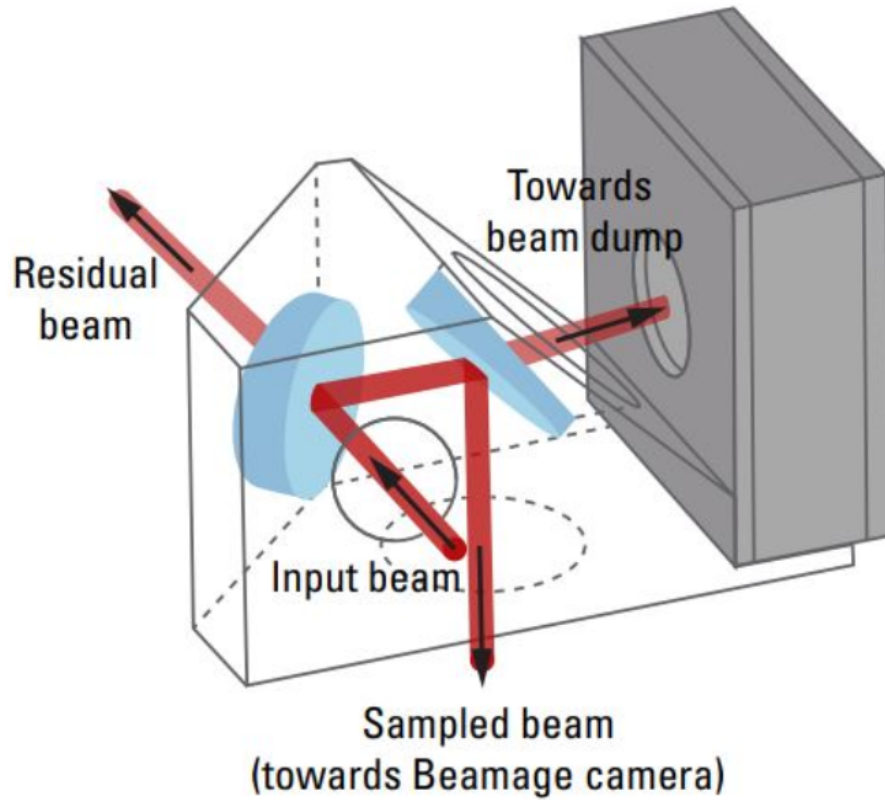


Figure 7: Two optical wedges set up to attenuate an input beam. The residual beam continues through their optical system and the second residual beam is absorbed by a beam dump. Image reproduced with permission from Gentec-EO.
All credits for the image go to Gentec-EO^[20]

One issue that wedges have that will need to be accounted for is ghost reflections off the back surface. Since one surface of a wedge is at an angle compared to the other surface, the reflections will not cause interference patterns, but instead will be reflected back out the front surface at an angle relative to the incident angle. This beam will have around 3.69% of the max power seen in **Figure 8**. This is still a large amount and will need to be

accounted for in the design by providing beam blockers and beam dumps to absorb the unwanted laser power.

Another way of laser beam attenuation is by polarization. This technique of attenuation is done by using linear polarizers, half-wave plates, and brewster windows. Since a large majority of lasers are linearly polarized, but the polarization direction may not be known, the following design is beneficial. The design consists of having the light pass through a half-wave plate to create a known polarization direction, preferably mostly s-polarized. Then to pass through a brewster window to reflect all the s-polarized light and transmit the small amount of p-polarized light. To further attenuate the beam, a linear polarizer can be placed after the brewster window or a series of ND filters. This design can be seen in **Figure 8**. Since linear polarizers work by absorbing the unwanted light, these polarizers can become damaged or even destroyed if too much power is sent through them. The brewster window would need to attenuate the laser beam well below 1% for a 100W laser beam.

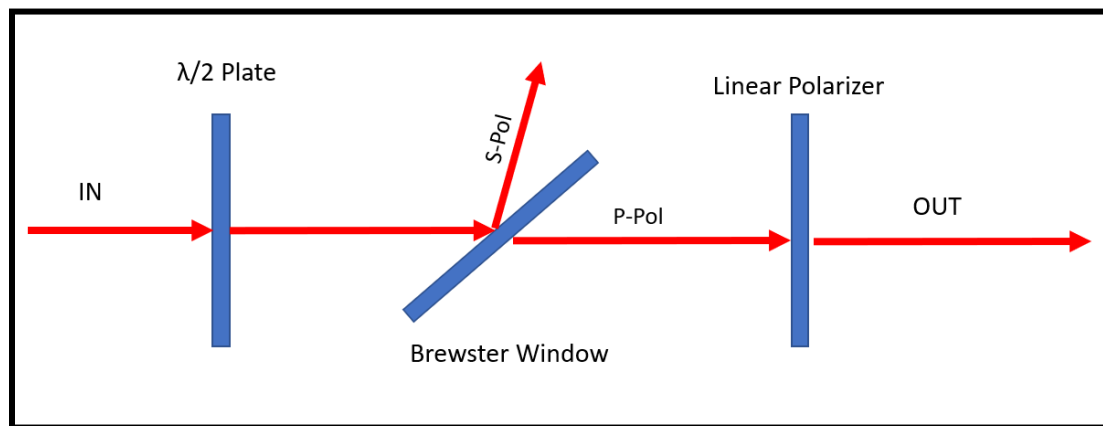


Figure 8: Attenuation by polarization manipulation. Variable attenuation is possible with this design.

Attenuation of an already low-power laser beam can be done by using neutral density (ND) filters. ND filters can be absorptive or reflective. Reflective ND filters are better for slightly higher power lasers, but will still absorb some portion of the incident light. These filters are able to be stacked in series with each other for variable attenuation ratios. The optical densities range from 0.1 ($T = 79\%$) to 6.0 ($T = 10^{-4}\%$). This allows for a large range of attenuation percentages by combining different optical density ND filters in series. The power density incident on an ND filter should not exceed 5 W/cm^2 . When too much power is absorbed by the material, thermal lensing could occur. This will cause the filter to act as a lens and change the properties of the beam.

3.2.3 - Beam Dumps, Traps, and Blocks

Beam dumps, traps, and blocks are an essential safety tool for dealing with unwanted laser light. Unwanted laser light can come in the form of reflected,

scattered, or diffused beams. There are many options for properly dealing with unwanted laser light in a safe way. One commercial used design (**Figure 9**) is by using a container with an aperture and a cone to scatter and absorb the light. The device is painted with a black paint that allows for optimal scattering and absorption. These devices also come with a heatsink to dissipate the heat.

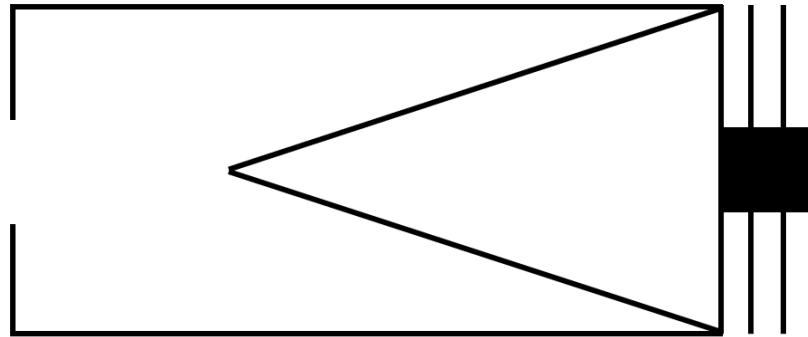


Figure 9: Commercial beam dump design

Another device for stopping unwanted laser light is a beam block. Beam blocks can consist of anything that is fireproof, can tolerate high temperatures, and absorbs and scatters light rather than reflects it. A simple beam block that can be used is a brick. Bricks do a good job at scattering and absorbing light due to the rough surface and high heat tolerance.

A solution for making reflective surfaces more efficient at scattering and absorbing light is to apply a matte black paint to them. This paint needs to be able to work with high temperatures. Some black paints that are used on grills and furnaces work great since they are rated for high temperatures. For commercial products, black acktar paint is typically used. This paint has a high absorption allowing for maximum absorption of unwanted laser light. The downside to beam blocks compared with beam dumps is that the light is scattered and not contained. While a lot of the light is absorbed in beam blocks, the rest is scattered out from the block. Beam dumps keep the scattered light contained, and in-turn, allows for more absorption from multiple surface scattering.

3.2.4 - Power Supply

This section will further detail what we have considered and researched for our power supply system. The topics that will be discussed include batteries and DC-DC converters (voltage regulators).

3.2.4.1 - Battery

We are interested in using a battery to power our device since we would like it to be portable. Due to this we did some background on batteries to determine what type of battery to use and if a rechargeable battery would be a good option. Batteries are classified into two types: primary (non-rechargeable) and secondary (rechargeable). Some of the pros of having a primary battery over a secondary

battery is that they have a lower initial cost, they are disposable, they are typically more readily available, lighter and smaller, and have a longer service per charge. **Table 7** shows the characteristics and applications of the different types of primary batteries we would consider.

Battery Type	Characteristics	Applications
Zinc-Carbon	Common, low cost, variety of sizes	Radios, Toys, Instruments
Magnesium	High capacity, long shelf life	Military and Aircraft Radios
Mercury	Very high capacity, long shelf life	Medical and Photography
Alkaline	Very popular, moderate cost, high performance	Most popular primary batteries
Silver/Zinc	Highest capacity, costly, flat discharge	Hearing aids, Photography, Pagers
Lithium/Soluble Cathode	High energy density, good performance, wide temperature range	Wide range
Lithium/Solid Cathode	High energy density, low temperature performance, long shelf life	Replacement for button and cylindrical cells
Lithium/Solid Electrolyte	Low power, extremely long shelf life	Memory circuits and Medical electronics

Table 7: Types of Primary Batteries

Some of the reasons we would like to use secondary batteries or rechargeable batteries over primary batteries is because they are more cost effective. Environmental concerns are also a driving factor for this choice as well as better energy efficiency of rechargeable batteries over disposables. Due to being reusable there will be less waste produced and while the initial cost may be higher the overall cost will be less than disposable batteries. They are more efficient because less energy is expended recharging them versus creating fresh batteries. They also have higher power densities, flatter discharge curves, higher discharge rates, and lower temperature performances.

They do, however, degrade over time with each recharge so they will have a certain life span as well and have a lower energy density than primary batteries. However, these downsides are outweighed by the upsides, particularly in our

application where portability is a concern. Since these batteries are able to be recharged, the need to bring spare batteries in case of low charge will be minimized. As the device will be able to be recharged via any standard wall outlet. **Table 8** shows the different types of secondary batteries we researched along with their characteristics and applications.

Battery Type	Characteristics	Applications
Lead-Acid	Most popular, several configurations, wide range applications, low cost, easy maintenance, low specific energy, limited cycle count, toxic	Automotive Industry, Emergency Power, Communication Systems, UPS (uninterruptible power supply)
Nickel-Cadmium	Oldest, long life, reliable, high discharge rates, wide temperature range, long shelf life, higher cost, memory effect	Portable Electronics, Industrial, Power Tools, Medical Devices, Aviation, UPS
Nickel-Hydrogen	Newer, higher energy density, less toxic than NiCd	Aerospace, Portable Electronics, Medical Instruments, Hybrid Cards, Industrial Applications, AA and AAA (consumer use)
Lithium-Ion	High energy density, high specific energy, longer cycle life, slow self-discharge, wide temperature range, protection circuit, more expensive, low maintenance	Laptops, Mobile Phones, Cameras
Lithium-Polymer	Flexible form factor, improved safety, lightweight, lower energy density, expensive, no standard sizes.	Laptops, Mobile Devices, Wireless Game Consoles, and Electronic Cigarettes

Table 8: Types of Secondary Batteries

When we were choosing a battery, we decided on a secondary battery for the reasons explained prior along with their uses being used a lot in portable devices. We wanted to have a power source that was easy to use and had a lower overall cost. We found that after comparing the types of primary batteries to secondary batteries we were able to see secondary batteries would be a great fit. From there we looked at the energy or power they contain, cycle life, efficiency, maintenance, etc as seen below in **Table 9**. Lithium polymer is not included in the list below because it has poorer conductivity due to having a polymer

electrolyte as well as not having standard sizes and being expensive, so we decided lithium ion would be a better option.

Specifications	Lead Acid	NiCd	NiMH	Li-Ion		
				Cobalt	Manganese	Phosphate
Specific Energy (Wh/kg)	30-50	45-80	60-120	150-250	100-150	90-120
Internal Resistance	Very Low	Very Low	Low	Moderate	Low	Very Low
Cycle Life (80% DoD)	200-300	1000	300-500	500-1000	500-1000	1000-2000
Charge Time (hr)	8-16	1-2	2-4	2-4	1-2	1-2
Overcharge Tolerance	High	Moderate	Low	Low	Low	Low
Self-Discharge/Month	5%	20%	30%	<5%	<5%	<5%
Cell Voltage	2V	1.2V	1.2V	3.6V	3.7V	3.2-3.3V
Maintenance Requirement	3-6 months	Full Discharge (90 days)	Full Discharge (90 days)	No maintenance	No maintenance	No maintenance
Safety Requirements	Thermally stable	Thermally stable	Thermally stable	Protection circuit	Protection circuit	Protection circuit
Toxicity	Very high	Very high	Low	Low	Low	Low
Coulombic Efficiency	90%	70-90%	70-90%	99%	99%	99%
Cost	Low	Moderate	Moderate	High	High	High

Table 9: Specifications of Secondary Batteries

After our research, we were interested in using a lithium- battery due to their lower life cycle cost with overall better performance. Lithium is the lightest of all metals, has greatest electrochemical potential, and provides the largest energy density for its weight. Some advantages of lithium-ion include that it has higher energy densities, low maintenance, no memory, no scheduled cycling, lower self-discharge, and has a variety of types when compared to other rechargeable batteries. However, some disadvantages include circuit protection, being subject to aging, transportation restrictions, expensive to manufacture, and it is not fully mature. Overall, we felt lithium-ion would be the best for our design after doing this research.

Our power supply requirements include having 6 hours of full load power on a single charge. Since all of the components in our device will not need to be running at full load throughout the test, an approximation of the power needed will need to be made. We will be able to further test these assumptions when we have the opportunity to run tests on the physical design. Since the Raspberry Pi, the NEMA 17 motor, PCB, and the high powered laser are the most power demanding devices we will be running, they will be a good place to start for sizing out batteries. At this point the budget does not allow much expense for a high-performance battery so we will need to seek budget friendly options that can still provide sufficient power.

To begin our search we looked at TalentCell, a popular brand of portable DC power supply that is widely available on amazon with fast shipping. Upon deeper inspection there is a decent range of types and sizes to choose from. The TalentCell power supplies are rechargeable 12V DC output lithium ion battery packs that come pre-packed, so no assembly will be required. The reason we chose to not build this part from scratch is due to the prevalence of affordable portable power supply units on the market. This will serve as a portable battery supply for the PCB, Raspberry Pi, stepper motors, camera, and lasers. Therefore, the amp hour rating will need to be taken into account when sizing the battery for our design.

Looking at the TalentCell rechargeable 12V/6000mAh DC lithium ion battery pack, it features a built-in overcharge protection, over-discharge protection, and short-circuit protection. This in fact comes standard in all of TalentCell's rechargeable battery pack models. This specific model contains ports for voltage draws as low as 5V unregulated but may need to utilize the 12V/3A port for our application. This is important to keep in mind as the battery pack does not contain voltage regulators to provide a constant output of 5V/2A max. We will discuss voltage regulators further in the next section of this text.

It will be possible to charge the battery pack while in use if needed. This will serve as a safeguard against the device being rendered unoperational when it is discharged. A Y-cable provided with the battery pack will allow for

charging/discharging at the same time across the same line. However, depending on the load, this means that the battery pack may charge slowly.

Also if the load takes more current than the power being supplied to charge the device additional current that is needed to be drawn will come from the battery pack through the adapter. This means there will be a net discharge, which could harm the battery. Therefore it is suggested to not run the device over 2A rated load, or the battery may be completely discharged. This means the Talentcell will need to be refurbished with new batteries.

The TalentCell rechargeable 12V/11000mAh DC lithium ion battery pack comes with a 5V/2A USB output and 12V/6A DC5521 output. To use this supply, current would need to be carefully regulated, but would work similarly to the 6000mAh version. However, now we would be able to power our electronics at full load for nearly 1.8 hours, a two-fold increase from before. However, you are going to be paying for the extra power since it is nearly double the price as well.

To compare the Talent Cell to a separate brand, the Sinkeu Portable Battery pack provides 88.8Wh on a single charge. Built as a portable battery pack for laptop computers, the bank provides a 110V/65W AC outlet. As well as two 5V/3A USB outputs. It also provides high speed charging capabilities, which allows for a full charge in less than 4 hours. The Sinkeu can even be recharged in a car with a 12V DC outlet. This unit provides up to 17760mAh of power at 5V output. **Table 10** provides a comparison of the power output and price difference between the battery pack's we have discussed.

Battery Pack	Power Output (@5Vref)	Price
TalentCell12V/6000mAh	12000mAh	\$39.99
TalentCell12V/11000mAh	22000mAh	\$53.39
Sinkeu Portable	17760mAh	\$99.99

Table 10: Power Output and Price of Battery Packs

After reviewing the price versus power output comparison, the TalentCell 12/11000mAh was chosen as our power supply for this project. It was discovered upon inspection that even though the Sinkeu Portable has more bells and whistles, such as the addition of an AC voltage output, these added significantly to the cost. Not only that, but the TalentCell11000 still had the highest mAh output advertised on its specification sheet. Since the additional features provided by the Sinkeu were not required and were outside of the specification put forth initially. The TalentCell 11000mAh was chosen for this project.

3.2.4.2 - Voltage Regulators (DC-DC Converters)

We are interested in designing a portable device that will be powered by a battery. Due to this we will need a DC-DC converter since batteries provide direct current and our device will use DC voltage as well. DC-DC converters are essentially voltage regulators which are needed since the operating voltage requirements of different electrical components can vary. Not only that but as the battery discharges, the variability of the voltage output requires correction to ensure that components downstream of the supply continue to receive power at adequate voltage to function.

We need to design a counter variable voltage/current as the battery discharges. Voltage regulators will be introduced into the PCB design in order to supply needed voltages. Such as a constant 5V to power the Raspberry PI, and +12V as well as +3.3V for the printed circuit board which includes the microcontroller and the motor driver. Keeping in mind that for the minute steps that we will be taking, this voltage range will likely not need to be at maximum. The battery's 6000mAh supply at 12V on one full charge, should last roughly 1 hour at full load.

We are also interested in the discharge for each of the individual cells of the TalentCell 11000mAh. Since we have six batteries within the pack, we'll need to multiply the discharge capacity curves by six. For example, a voltage of 10V is around 1.66V per cell. 1C correlates to capacity of the cell, which is 22000mAh per cell. If you follow the discharge curve for 1C. This discharge curve shows us the decrease in voltage as the capacity within the battery decreases. Therefore, we will have a need to correct this in all of TalentCells battery packs, through the introduction of a voltage regulator.

Although DC-DC converters allow you to use one voltage supply for multiple electrical components, there is some power loss. It is important to choose a converter with a higher efficiency to have a device that is more reliable. More efficiency allows for better quality, less noise, less heat, less ripples, and better regulation. In our application we will only need to utilize the 9V/1A output source of our DC supply to regulate. This will help to reduce all of the factors mentioned previously.

When looking at DC-DC converters we compared linear versus switching regulators because both contain buck converters which lower a higher voltage level to a regulated level. However, switching regulators are more efficient and have less power dissipation than linear regulators so we focused more on choosing a switching regulator. Different switching regulators include buck, boost, buck-boost, cuk, sepic, and zeta. We further researched buck, buck-boost, cuk, sepic, and zeta. From this we looked further into buck, sepic, and zeta since they all have lower positive output voltages.

A buck converter, as seen in **Figure 10**, is referred to as a "step-down DC-DC converter." It consists of a power transistor used as a switching element, a

rectifying diode, an inductor as energy storage, and a filter capacitor. It lowers the output voltage while increasing the output current.

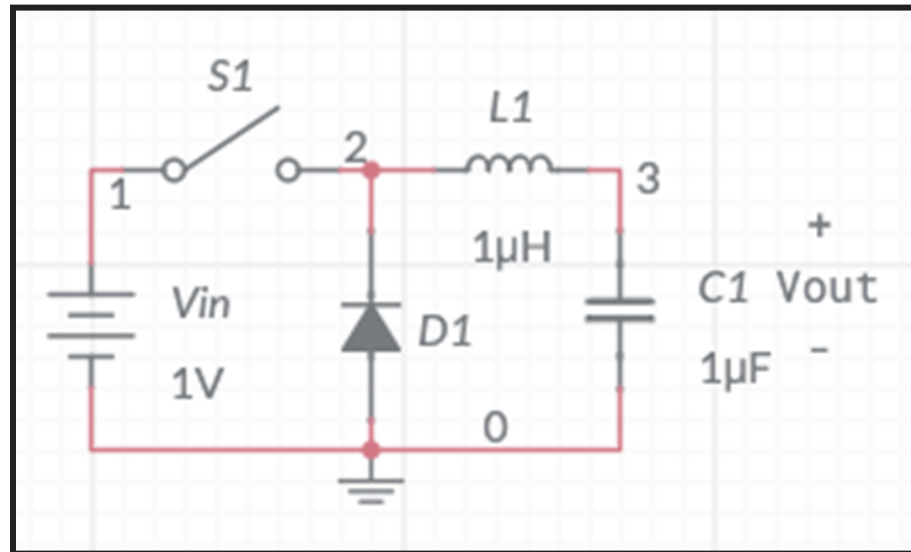


Figure 10: Buck DC-DC Converter

A single-ended primary-inductor converter (SEPIC), as seen in **Figure 11**, is similar to the buck-boost converter because the output voltage can be greater or less than the input voltage by controlling the duty cycle of the switch. It contains a boost converter with an inverted buck-boost converter. However, it has a non-inverted output due to using a series capacitor.

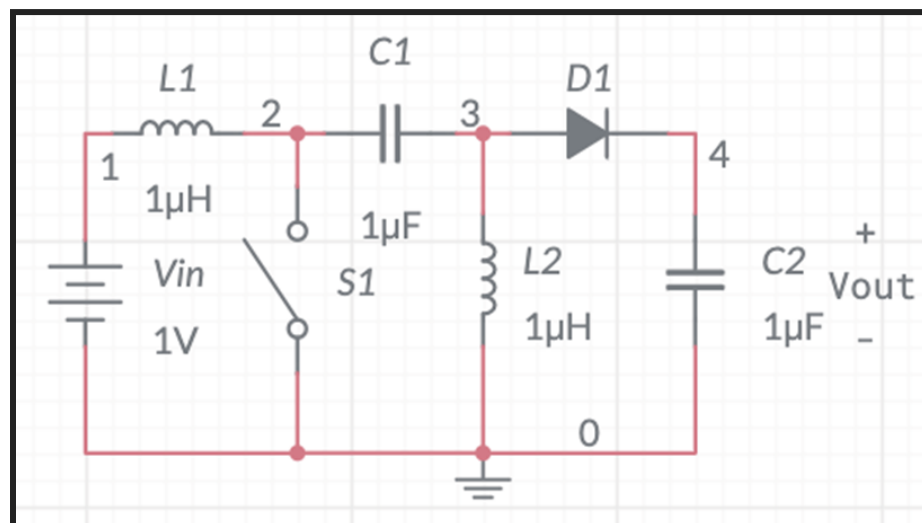


Figure 11: SEPIC

Finally, a zeta converter, as seen in **Figure 12** is configured from a buck controller that drives a high-side PMOS FET. It provides a positive output voltage by using two inductors and a series capacitor (flying capacitor).

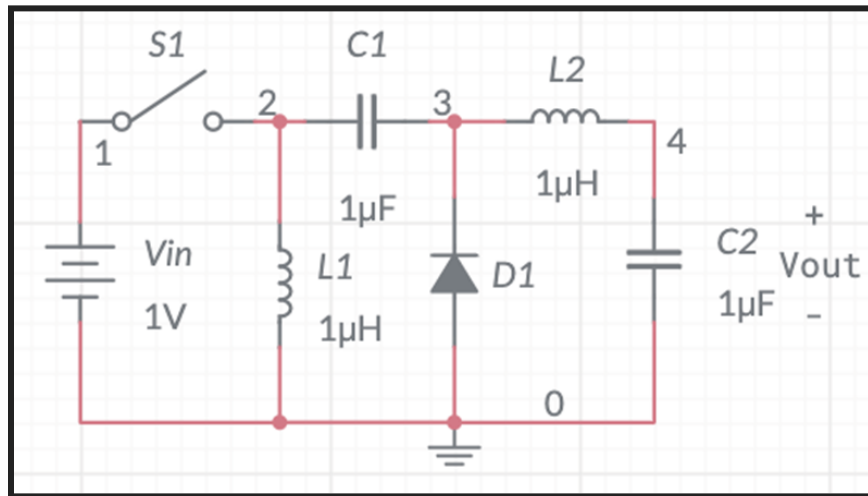


Figure 12: Zeta Converter

Along with researching the different DC-DC converters, there are also different characteristics to consider when researching data sheets including efficiency, current rating, temperature rating, ripple voltage, regulation, transient response, voltage rating, size and weight. When looking at efficiency it is important to make sure the source can provide enough power to account for inefficiency. Things to consider is to look at the maximum amount of current, voltage, and temperature the converter can handle. Also, make sure the ripple voltage, size, and weight meet our needs.

The Talentcell 11000mA has 9V/1A and 5V/2A outputs and a 12V/6A max input/output port. Powering the Pi directly through the 5V port may not be ideal as the raspberry pi version we are using requires 2.5A, which could cause the pi to crash in cases where we need to use higher current. However, since we are not using add-on shields, we should be able to get away with the lower current draw, since we will be powering the motor driver separately. For the motor drivers and motor drawing power through the 9V terminal would not be ideal, since the TMC2208 motor driver will draw up to 1.4A of continuous current. Therefore, we will design a voltage regulating circuit that will use the 12V/6A input/output port to power the PCB. We will need to regulate the 12V variable power supply output down to +3.3V. We chose to use the 12V output for the battery because it outputs up to 6A of current and the driver needs up to 1.4A of continuous current and there will be over 4A left to draw from for the microcontroller.

We decided to use WEBENCH Power Designer to create a blueprint for our regulator design. This way we could add the input/output voltage specification and WEBENCH would provide a generated solution for our needs. When narrowing our search for a PCB design we had to keep in mind the tradeoffs

associated with efficiency, size, and switching frequency. For example to reduce the size of the PCB footprint we would have to increase the switching frequency. Which in turn would reduce the efficiency of the device due to the higher switching losses that would result.

A more efficient design, would lower our energy consumption and wouldn't cause a large dissipation of excess power. This would result in an overall cooler system as well, since we would not be trying to shed excess heat. But then that would cause the PCB to be larger and expensive. For our application we chose a balanced design. This would help us to find a design which would produce a low cost, small footprint, and high efficiency device. Keeping in mind the specifications listed earlier for our project the balanced option still gave us a small sized PCB with only a 446 mm² footprint.

When we began looking into voltage regulators the LM7805 at first seemed like it was going to be a good fit. This regulator is capable of outputting a constant 5V output at 1.5A, given an input of 10V. The quiescent current at 5V gives us a value of 10mA flowing through to ground when the voltage regulator is in idle. This would have an impact on the overall battery life of our system. With a maximum input of up to 27V according to its datasheet. With a voltage drop of roughly 7V across the LM7805, we would have needed a dissipation of about 7 Watts worth of heat over several hours of operation.

The datasheet gives recommendations for capacitors to be placed on both the input and output of the voltage regulator. The capacitor at the input of the regulator is going to help with smooth out our voltage supply. Since the capacitor will hold some charge, any minute frequency disturbances to the voltage supply will be smoothed over time. The capacitor placed on the output provides a similar function of ensuring a clean output to the connected circuit on the output side.

Upon inspection of the design illustrated in the manufacturers guide, we have noticed that we should place a diode at the beginning of the voltage input into the circuit as well. This will protect the regulator circuit from accidental reverse voltage polarity. This shouldn't occur since our batteries are enclosed, but in case the batteries need to be replaced, this protection shouldn't be overlooked. shows a schematic of the voltage regulator circuit, and suggested capacitor values from the datasheet for LM7805.

With some fine tuning this regulator could have sufficed for our needs however upon further inspection we deemed it necessary to not draw our power supply from the 12V output port since it would serve to disable the ability to charge the device while in use during testing. Also the heat produced would have required additional heatsinks be added. Therefore we must rethink our design in order to utilize the 9V/1A output port of our power supply for the motor driver. We also initially did not have a microcontroller with our PCB design and needed to have more output current to be provided to this as well.

We came across a DC regulation design that used the TPSM82903 step-down DC-DC convertor module which could be used for our application. The TPSM82903 would provide much higher efficiency as compared to the LM7805 we reviewed. The module also introduces a selectable switching frequency that allows for quick transient response. It would also support a range of voltages from 3V-17V, and provide a voltage output accuracy within 1% + or - with the control topology presented in its datasheet. Unlike the LM7805, the TPSM82903 would also be able to enter a power saving mode, when the load is light. This will allow us to control power loss when the system is not in use, and helps to maintain high efficiency when not in use by the load. Unfortunately the product was not available at this time, and therefore we were not able to order this part for our final design.

We also looked at the TPS56628 synchronous buck converter with eco-mode that takes a range of 4.5-18V in and outputs a range of 0.76-5.5V with an output current of up to 6A. The quiescent current is around 950uA and it has a switching frequency of 650kHz. It comes in an 8 pin surface mount package with many features including soft start, power good, overcurrent, overvoltage, and undervoltage protection as well as thermal shutdown in the event of excessive heating. The TPS56628 was a wonderful option for incorporating safety features into our design and safeguarding the PCB components. We don't anticipate it being a problem, with an operating temperature of up to -40 to 85 degrees Celsius ambient. When designing the regulator in TI's WEBENCH, we found the efficiency to be 86.1% with a BOM count of 13. While there were other regulators that had higher efficiency, it was not available to order with JLPCB which we chose as our manufacturer while the TPS56628 was. This regulator was also chosen for it having a simpler design than other switching regulators which would help when designing the board. The TPS56628 is enabled by pulling the EN high and has a voltage divider on the VFB pin to convert to 3.3V. It also has bypass capacitors on both the input and outputs to help to further smooth out the output voltage and provide a steady 3.3V DC output for us.

3.2.5 - Motors and Drivers

3.2.5.1 - Motors

For our design we are interested in utilizing electric motors because we need precision positioning. Electric motors convert electrical energy into mechanical energy mostly in the form of rotational motion. The motor's shaft rotates by a force generated by the electric current flowing through a coil in a magnetic field. There are many types of electric motors including DC (direct current) motors, AC (alternating current) motors, and special motors. They differ in the type of current flow they use, the design of their coils, and how they generate a magnetic field. We clearly are not interested in AC motors due to our power supply being DC.

We further researched DC motors and special motors that use a DC power supply. Upon research we found DC can be split into brushed, brushless, or coreless. Brushed DC motors have brushes and commutators. They connect a stationary circuit with a rotating circuit. They do require a lot of maintenance but are simple in design and inexpensive. They include separately excited motors, self-excited motors, and permanent magnet motors.

Brushless motors as the name implies do not have brushes and commutators. The power is supplied to the stator of the motor and the rotor is a permanent magnet. It needs to have electronic commutations using switches controlled by the microcontroller. They are more expensive with more precise speed control and positioning. They also have a longer life span, are more efficient, and free of maintenance. Coreless motors help eliminate issues dealing with iron cores. It reduces the winding inductance which increases the lifetime of the motor. Since the mass of the rotor is decreased, the rate of acceleration and deceleration are increased. Overall, the brushless motors seemed to be the best option since we are more interested in having precision control.

We also further investigated special motors and were most interested in servo and stepper DC motors. Servo motors are used for pushing/pulling or lifting or rotating an object at an angle. They can run on either AC or DC. It includes a gear assembly (control over torque), controller (microcontroller), sensor, and feedback system. It is controlled through the microcontroller using pulse width modulation (PWM). Servo motors also can rotate up to 90 degrees in either direction.

On the other hand, stepper motors are brushless DC motors whose full rotation is divided into several equal steps. They have three modes which include full step mode, half step mode, and micro step mode. Going from full to micro step, for each 360-degree rotation the distinct steps increase. The more steps the better the angular resolution, however, the torque rating decreases. Stepper motors are used for accurate and precise positioning. They also have less maintenance, are reliable, and have a long lifetime. They can either be permanent magnet (PM) motors, variable reluctance (VR) motors, or hybrid motors.

Permanent magnet steppers have rotors with permanent magnets to create rotation and torque. They have low power requirements and produce a great amount of torque. Variable reluctance steppers have iron rotors with "teeth" around the circumference. They have a high angular resolution, but a smaller amount of torque. Hybrid steppers have rotors that are a combination of both PM and VR rotors. This way they have higher angular resolution and a higher amount of torque.

We had to keep in mind that our application would require precision control over speed of rotation. Due to this we decided to rule out the other DC motors and the DC servo motors and will be using a stepper motor for our design. Due to stepper

motors sacrificing speed of rotation for precision we will need to optimize for precision, as the distance we must travel to adjust our camera is very small.

Later in the design stage, we decided to introduce an adjustable actuator that would allow for micron sized step resolution directly to the camera's translation stage. This means that the motors step resolution would be greatly amplified, which may have allowed the use of a servo, however, for ease of troubleshooting any unforeseen issues we chose to stick to the stepper motor design.

As we browsed through various DC stepper motor models, we needed to keep certain questions in mind to meet the design specifications. We needed position accuracy in tandem with a micron translation stage to within 5um, a motor driver to match the pulse's required to achieve that distance on the slide we would choose, enough torque, and enough motor speed to realize this distance in a reasonable amount of time.

To start the discovery for the best stepper motor for our application we considered the budget and looked further into a 28BYJ-48 stepper motor. This stepper motor features 2048 steps/revolution resolution, which for only \$2/motor is very impressive. The delay between these steps is only 2ms, and it has a max speed of 500 steps/s. Which means that the motor would only turn about 90 degrees per second. This small delay wouldn't cause much frustration, but over long tests it may get a bit time-consuming. Therefore, due to our budget we will consider other options to look for a better stepper motor.

At the higher end, the National Electrical Manufacturers Association or NEMA line of stepper motors are great for applications such as our CNC machine. We looked specifically at the NEMA-17 and NEMA-23 for our application. These motors also come in either short, medium, or long and the following are all bipolar stepper motors. These motors were found on amazon's website and the manufacturer is Stepperonline. When you go to Stepperonline's website there are many to choose from, but these were able to get shipped quickly and were not too expensive. We compared them as seen below in **Table 11**.

	NEMA-17	NEMA-23
Frame Size	42mm x 42mm	57mm x 57mm
Body Length	20mm	113mm
Weight	140g	1.8kg
Holding Torque	16 Ncm	3 Nm
Rated Current	1.0A	4.2A
Step Angle	1.8 degrees	1.8 degrees
Steps/Revolution	200	200
Price	\$11.99	\$39.99

Table 11: NEMA-17 and NEMA-23 Comparison

The NEMA-23 has a very high holding torque for its small size. However, the NEMA-17's weight per motor is 200g, which is less than half a pound per motor. We decided to choose the NEMA-17 stepper motor, since we weren't going to need the large holding torque offered by the NEMA-23. It also made sense from a budget, power, and weight perspective to go with the smaller motor. We decided on these motors as they are widely available in the market and used in DIY projects where motion control is needed. Therefore, we chose to go ahead with the NEMA-17 for this application. Below you can see an image of the NEMA-17 stepper motor.

3.2.5.2 - Motor Drivers

A motor driver interfaces between the motor and the microcontroller. They regulate motor speed, torque, and position outputs. It takes the low current signal from the microcontroller and turns it into a higher current signal to drive the motor. Motor driver integrated circuits (ICs) use an H-bridge to connect it to the microcontroller. When selecting a motor driver, it is important to look at the voltage and current ratings and that it is compatible with your motor and microcontroller. We are interested in a stepper motor driver due to using a stepper motor. The stepper driver converts the input DC into a stepped output current.

For our PCB design we will need to consider a stepper motor driver that can handle the 0.3A-1A rated current draw out that NEMA-17 motors will require. When researching popular stepper drivers for the NEMA-17 stepper motor, the A4988 stands out as one of the cheapest. It features a 2A peak current rating with a recommended 1A current for normal operation mentioned in its datasheet which can be seen below in **Table 12**.

Characteristic	Symbol	Rating	Units
Load Supply Voltage	V _{bb}	35	V
Output Current	I _{out}	+/-2	A
Logic Input Voltage	V _{in}	-0.3 to 5.5	V
Logic Supply Voltage	V _{do}	-0.3 to 5.5	V
Motor Outputs Voltage		-2.0 to 37	V
Sense Voltage	V _{sense}	-0.5 to 0.5	V
Reference Voltage	V _{ref}	5.5	V
Operating Ambient Temperature	T _a	-20 to 85	°C
Maximum Junction	T _{j(max)}	150	°C
Storage Temperature	T _{stg}	-55 to 150	°C

Table 12. Absolute Maximum Ratings for A4988

We can make certain this current rating does not exceed by adjusting the current limiting screw. Using the equation below to adjust the current accordingly using our voltmeter and a screwdriver on the screw in the driver. The controller only provides us with a step resolution of 1/16 subdivisions on top of our 200 steps provided by the motor. Which may be adequate considering the attenuators we will be turning with the motors will add to our total resolution. Helping us achieve micrometer increments of accuracy.

$$\text{Current limit} = V_{\text{ref}} \times 2.5$$

The DRV8825 stepper motor driver is another suitable option. It can deliver 1.5A rated current, a power supply voltage range of 8.3V-45V and a microstepping resolution of 1/32 subdivisions. In conjunction with our stepper motor, this will allow us to achieve steps of 0.00125mm. Below is the formula used to calculate the arch length per step given the DRV8825 functionality.

$$.04\text{mm/step} * (1/32)\text{subdivisions} = .00125\text{mm subdivisions/step}$$

In the DRV8825 stepper motors data sheet it is possible to drive current as high as 2.5A/24V with proper heat sinking at ambient temperatures of 77°F. It is also possible to adjust the maximum current the motor will draw through the motor driver no matter the motor's current rating. This ensures that we can set the current limit to the appropriate rated current of the motor. Which can prevent overheating in some cases.

For example, if we limit our current to 1.5A on the stepper motor driver we would not be able to use a motor at max power which is rated for 2A. However, for our motor, which is rated at 1A we will need to adjust our stepper motor driver. A current of 1.5A (which the stepper driver is rated for) would cause our motor to overheat and possibly burn out.

Therefore, it is extremely important for our application that we measure the reference voltage across the potentiometer with reference to ground and adjust the max current to our motor's specifications. This will require the use of a voltmeter, in conjunction with a screwdriver to drive the potentiometer to the desired voltage level. The equation below will be used to adjust the voltage to the correct level for our application. Rcs values vary for different manufacturers and this will have to be adjusted according to the drivers we receive. (Usually Rcs= .05, .1, or .2 on DRV8825 must verify through inspection)

$$\text{Current limit} = V_{\text{ref}} / (8 \times R_{\text{cs}})$$

For comparison we will also consider the TM2208 stepper driver. This stepper driver features 1/256 subdivisions regardless of stepping resolution chosen. This allows for a much quieter operation in comparison to the DRV8825, as this many steps allows for near perfect sinusoidal control. It can deliver 1.2A continuous current with 2A peak. And features a much larger heat sink. Below is the formula used to calculate the arch length per step given the TM2208 functionality.

$$.04\text{mm/step} * (1/256)\text{subdivisions} = .156\mu\text{m subdivisions/step}$$

Overall the TM2208 is a superior stepper driver with its noise reducing features and greater step control resolution. This of course comes at a cost as the TM2208 is pricier than the DRV8825, but not by much. Due to the precision needed in our design, as well as the noiseless quality provided by the chip. We will be going with the TM2208 for this project as it will allow for much higher resolution when we use the motors to perform our laser profile scan.

3.2.6 - Microcontrollers

Microcontrollers are small, single chip devices which contain an entire computing environment accessible through pins on the outside of the package. Typical

systems within a microcontroller are ALUs, addressable memory, processing cores, and control circuitry. All microcontrollers sold include an instruction set architecture, which is the low-level machine language for programming that device. In modern times, microcontrollers are now programmable through a wide array of programming languages, while also becoming more feature robust. Modern microcontrollers can feature several-hundred megahertz to a few gigahertz in clock speeds, while utilizing 64-bit busses, and complex timing and analog-to-digital conversion circuitry. When deciding on a microcontroller for a project, it is important to take note of the project's computational requirements.

A single-package System-on-Chip (SoC) can solve a lot of computing requirements in projects, but if the project's computing requirements are simple (ex. Driving LCD character displays), much of that computing power is wasted on skipped operation cycles, and thus wasting electrical power.

Many popular microcontrollers can be ordered pre-assembled to development boards, where they may be programmed through USB or UART, and have all their GPIO pins broken out from the processor to standard pin headers on the PCB to allow for easy interfacing with peripherals. This makes the task of prototyping iterative designs much quicker.

When it came to our design constraints, we would need a controller interface that is flexible, affordable, and easy to use. Matching these criteria, we looked at both the Raspberry PI and Arduino as possible controllers for our project.

The Arduino would allow us to control electronic components with its onboard microcontroller and give us access to expansion modules available on the market that would add Bluetooth capabilities. We could also add shield modules to help drive our motors, however, due to the specifications of this project we won't be considering this functionality. Using the Arduino, we would be able to store and execute our code without needing to keep it directly connected to a computer. Coding on the Arduino is also very simple due to its simple design and use of a simplified C coding language.

Once the programming is uploaded, we could then use the device as a stand-alone system to control our electronics. Sensors and other hardware could then be directly controlled via the Arduino's I/O pins. The Arduino would also use a 5V logic power supply and 7-12 DC input voltage. With a current draw of about 100mA. This would allow us to directly power the device using our portable power supply. With such a low current draw and simple microprocessor, the Arduino uses very little power, which would significantly increase the battery life. Below in **Table 13** are max power consumptions for various Arduino models considered.

Arduino Model	Power Consumption
Nano	19mA
Pro Mini 5V	14.6mA
Pro Mini 3.3V	5.1mA
Uno	98.4mA
Mega	73.2mA

Table 13: Power Consumptions of Various Arduino Models

The Raspberry Pi, as seen below in **Figure 13**, on the other hand uses a microprocessor versus a microcontroller (like the Arduino), which requires the Raspberry Pi to have the support of various other chips on the board to help in storage and processing of code. Essentially the Raspberry Pi is a mini computer, which allows for a greater degree of complex control, but will also require more power than the Arduino.

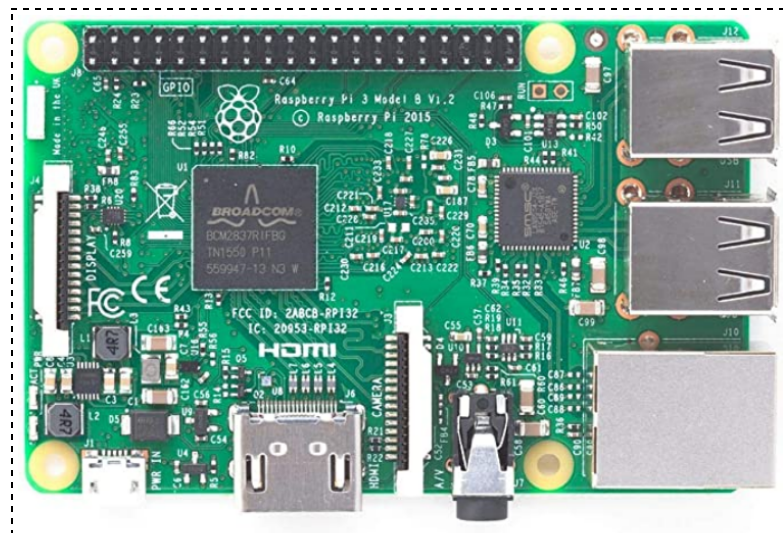


Figure 13: Raspberry Pi 3+ B V.2

The Pi will also allow for direct programming versus needing an external computer to program. To control the I/O pins on the Raspberry Pi for our sensors, and motors we will need to write code that would allow us to control the lower level software running on the board. This makes programming the Raspberry Pi a degree of difficulty greater when compared to the Arduino. **Table 14** below is a chart of the power consumption ratings of the various Raspberry Pi models.

Pi Model	Desktop Idle (No wifi)	1080p Video (No wifi)
Pi Zero W	220mA	245mA
A+	225mA	245mA
3 A+	305mA	320mA
B+	320mA	340mA
3 B+	510mA	520mA

Table 14: Power Consumptions of Various Raspberry PI Models

If we were only going to need logic functionality to control electronic components, we would have gone with the Arduino. We would do this because it requires less power to operate, is simpler to program, and offers all the functionality we would need to drive the motors. This would allow for longer battery life, less robust voltage regulation, and a simplified programming experience. It is also the cheapest option in comparison to the Raspberry Pi and is widely available on the market. Shortages at this time are causing shipping delays on nearly all versions of the Raspberry Pi or they are out of stock completely. Therefore, it is not going to be an option at this time to order a new model, as delivery times would be beyond the deadline for this project.

However, we will need to provide the functionality of a small computer if we are to use a camera sensor to detect the laser beam in real time and process the data. Therefore, a Raspberry Pi will be a necessary component to realize this project's needs. This is due to its hardware giving us all we require to drive our sensors, motors, and provide logical control. We will also have the added option of providing USB storage of the data we collect onto the Raspberry Pi for later. **Figure 14** shows an image of the Raspberry Pi camera we will be using.



Figure 14. Raspberry PI Camera V2 NoIR

As luck would have it, we have our own Raspberry Pi 3 B+ module and we will be using it for this project. It even comes with a built-in 4.1 Bluetooth module. This will allow us to add wireless data transfer if we want, to another computer or device. The Raspberry Pi 3 B+ features a 1.4GHz CPU, and the USB Ethernet controller connectivity at throughput speeds of up to 300Mb/s. At idle power draw is about 2.5watts with a max draw of about 5.7Watts under full load. This will be taken into consideration later when we choose the size of our battery. **Table 15** below shows the Raspberry Pi B+3's idle vs full load power consumption.

Raspberry Pi B+ 3	Power Consumption
Idle	2.7Watts
400% CPU Load	6.4Watts

Table 15: Raspberry Pi B+ 3 Idle versus CPU Load Power Consumption

We also need a microcontroller on our PCB to be able to communicate with the Raspberry Pi 3B+ and the TMC2208 (motor driver). The Raspberry Pi has pins that operate at 3.3V, so either we needed to have a microcontroller with pins that operate at that same voltage or have a voltage converter in between the pins. For simplicity of design, we wanted to use a microcontroller that had digital pins that operate at 3.3V. Between the team members we had TI's MSP430FR6959 and an Arduino Uno. We looked at using the microcontrollers on these boards since we already had a development board and we were more familiar with using these for our previous classes. While the MSP430FR6959 was more familiar to us we wanted to get experience with the Arduino IDE and the microcontroller on the Arduino IDE is removable while the MSP430FR6959 is not. This will be helpful when learning how to program the microcontroller. The MSP340FR6959 operates between 1.8 to 3.3V, it has multiple I/O ports, and UART and I2C serial communication. The ATMEGA328P has 32KB of program memory, read/write capabilities, 23 general purpose I/O lines, a serial programmable USART, it operates between 1.8-5.5 volts, and it comes in a through hole and surface mount package. While both have many other features, this is what we focused on mostly. We decided to choose the ATMEGA328P for our microcontroller since it offers us all of the features we want, it is removable on the Arduino Uno, we can program it with the Arduino Uno, and we can get experience with the Arduino IDE.

3.2.7 - OSI Model

As stated in Tanenbaum and Wetherhal_[4], the OSI model is a conceptual model for the hierarchical layering of the network. The OSI model has seven layers, each with a well defined function: Application, Presentation, Session, Transport, Network, Data Link, and Physical. The application layer is responsible for providing the end user an interface with communicating over the network. The presentation layer formats ingoing and outgoing data from the application layer

and session layer. The session layer maintains the connection between the two devices by controlling device ports and sessions.

The transport layer handles the data through transmission protocols such as TCP and UDP. The network layer provides the routing on the network, deciding which physical path the information will take. The data link layer defines the format, or frame that data will be sent over the line, as well as providing in circuit error detection. The physical layer is the physical medium over which the information signal is propagating. This project's design will focus primarily on the lowest four layers, but all will be developed for.

3.2.8 - Physical Link Layer In Design

The Physical Link Layer is the lowest layer OSI model, and is responsible for carrying the raw information between two connection points. In an ideal situation, transfer rates would be infinitely high, but real-life applications are plagued with noise sources that limit what is physically possible for a transmission medium.

For electromagnetic signals running through a trace in a PCB circuit, if the logic rise times are too steep, stray inductances and capacitances along with the trace resistance will behave similar to a filter circuit, putting a hard cap on the frequency of transmission through that trace. This can be mitigated partially by proper PCB design. When using high frequency signals, it is important to minimize the distance between the source and the destinations. A longer PCB trace will introduce more stray transients into the system. When routing, the high frequency trace should be placed between two ground planes. This reduces the effect of noise entering and leaving the channel. If using a board with both digital and hybrid signals, it is important to keep the high-frequency digital lines far and isolated from the analog signals.

For electromagnetic signals propagating through air, the strength of the signal is reduced due to the inverse square law, so the distance between devices must be taken into account when implementing wirelessly connected designs. Wireless channels are very susceptible to outside noise, and care should be taken that if the device will be used in a noisy environment, proper care should be taken to ensure that data transmissions are reliable.

3.2.9 - Data Link Layer in Design

The Data Link Layer sits just above the Physical Layer on the OSI model, and is responsible for handling the data stream coming in from the physical layer, and converting it to a stream of bits that is understandable by the circuits interfacing over it. Another key role of the data link layer is to verify the incoming data stream at a low level, and provide the framework for sending and receiving logical blocks of information called frames. When designing physical connections between chips, it is important to check other link parameters in order to establish

a proper connection. Many chips only communicate within a fixed interval of frequencies, and become unresponsive above or below that range.

Error correction is implemented at the physical interface in hardware for common standard protocols, and is not typically accessible by the end user, but some can still be configured in software. There are a number of error detection and correction schemes in order to verify data integrity, ranging from Cyclic-Redundancy-Checks (CRC), parity bits, and checksums calculated using hashing algorithms on the input. For the USB-to-UART bridge ICs, they were configurable to accept 3 different parity modes over UART. Most implementations however restrict this, and instead have a set list of compatible communication protocols for which the user can utilize for sending data.

3.2.10 - Communication Mediums

Choices for a suitable physical medium for communication between the Raspberry Pi 3 B+ (RPI) hosting the server connected Windows 10 computer (PC) hosting the client was researched, and the decision had to be made between connection over a USB-Interface, a bluetooth connection, or over the Local Area Network using Wi-Fi or direct connection through Ethernet

Going into this design decision, it is important to take note of the other hardware decisions in this project. The most important of which is the decision of using a RPi as compared to a MCU on PCB. The RPi hosts a full Bluetooth 4.1 implementation on the chip as well as IEEE 802.11.b/g/n/ac wireless LAN and Gigabit Ethernet over USB 2.0, with a plethora of APIs for interfacing through Linux drivers. This allows for the freeing up of ports on the board for other connectivity needs. However, since the RPi hosts a linux-based operating system, a direct USB connection would not be possible to the PC, due to both ends wanting to be the master of the bus.

Therefore, in order to implement a USB connection from the RPi to the PC, a serial-to-USB bridge IC would have to be used. This was undesirable, as these chips had high per-unit cost, along with increasing the complexity of the design. The chips that were considered for this approach were the **Silicon Labs CP2102N**_[18] and the **FTDI FT230X**_[19], priced at \$2.16 and \$2.28 respectively. These bridge ICs also had a hard cap on the maximum data rate that could be transferred from the RPi into the IC, with both the **CP2102N** and **FT230X** having a maximum baud rate of 3 megabaud.

The Bluetooth protocol RFCOMM is a simple set of transfer protocols, designed to emulate a serial port, and allows a socketed connection to be created over bluetooth using high-level language libraries available on the Linux OS. A Python library named “PyBluez” provides a high-level API to interface with the Bluetooth hardware using Python. This library allows for the creation of Bluetooth sockets, which can be read from and written to by both ends of the connection. However, bluetooth is hopelessly limited by its transfer speeds. Bluetooth 4.2 has a transfer

speed of 1 Mbps, meaning to move the 100+ megabytes of image data, it would take approximately 14 minutes at this speed, not taking into account any overhead required to facilitate this approach.

The image data could be left on the RPi, and processed there, but this would take an equally long time to do, as the RPi is limited in raw computational power. Section 5.3 - *Software Design*, is built around the premise of using network sockets over LAN, as this medium provides the highest throughput, along with not adding any additional ICs to the bill of materials.

3.2.11 - Development Environments

In order to develop our Python applications, a development environment is needed to foster the creation of the programs. The two main approaches to development environments for Python are either an Integrated Development Environment (IDE) or a standalone text editor. IDEs provide a suite of features for application development, and the ones that stand out the most for Python are PyCharm and PyDev for Eclipse. PyCharm is developed and maintained by JetBrains, who also developed IntelliJ for Java, and provides a full debugging suite, version control integration, and package management and more. PyDev for Eclipse also provides similar functionality on top of the already robust Eclipse IDE. However, this application is planned to be small, consisting of only 4 main programs, so the source text editors seem like a better option. Popular text editors for Python include Visual Studio Code and Sublime Text. These programs lack the tools that the IDEs provide, however they make up for it in their ease of use. In order to manage the Python packages, Python virtual environments will be used to maintain a consistent environment for the programs to run in. For this project, VSCode is selected as the primary development platform, as we already have experience developing within it. VSCode supports IntelliSense for Python, which allows for quicker code development using recommendations and auto-complete. Intellisense also makes understanding library functions by displaying a window with useful information about classes and functions when moused over.

3.2.12 - Version Control

As the project matures, there will be inevitable changes to the overall software design, and it is important that all versions of the software are saved for reference or backup in case a program-breaking bug is introduced. There is also the issue of maintaining concurrency between two or more developers to ensure that everyone is working on changes to the latest version of software, and allow each developer to push their changes to everyone else. Version control systems aim to solve these problems.

The most popular version control platform as this paper is being written is GitHub. GitHub allows users to push and maintain their projects on the cloud for free. In order to interface with GitHub, the end user needs to use a program to

validate their identity and access the GitHub APIs. The most popular programs to accomplish this are the GitHub CLI and git. The GitHub CLI is developed by the developers of GitHub and is specifically designed for GitHub, where git is for interfacing with many types of version control software such as BitBucket or GitLab. The CLI provides the end user with a more extensible interface and allows for complex commands to be created using extensions. For this project, the GitHub CLI will be used to interface with the project repository on Github.

3.2.13 - 3D CAD Software

This design that we have chosen requires a few parts to be 3D-Printed, and thus requires CAD software to design them. There are several options on the market at varying price points, with differing levels of functionality and support, so a decision had to be made between them. The pieces of CAD software researched were Fusion 360, Solidworks, and Blender. Fusion 360 is developed by Autodesk, the company responsible for Eagle, AutoCAD, Maya, and several other CAD and visual effects software. Fusion 360 comes with a one-year educational license that would allow us to avoid paying the \$60 a month price. This software comes with a large collection of documentation and company support if you own a license.

However, since Fusion 360 is license based, its source code is proprietary. Similar to Fusion 360, SOLIDWORKS is another proprietary CAD program. SOLIDWORKS currently costs \$60 a year for students, but can be accessed for free using the university computers. On the other hand, Blender is an entirely free modeling software that also comes with a plethora of documentation and community support. There is no license for Blender, and it can be downloaded and installed on as many systems that we require. Blender is regarded by many to have a very steep learning curve, but this is not due to bad design. Blender is packed full of functionality and menus such that ample time is required to be spent within the software to become comfortable navigating to menus that you're looking for. For this design, we chose Blender, as it is free forever, and will allow us to carry our design skills with us into the future.

3.2.14 - Tkinter Python Library

There are several user-interface libraries available for Python, such as wxPython or PyQt, but Tkinter was selected as the library of choice due to the Pygubu-Designer software. Pygubu-Designer is an open-source GUI program that allows you to view the Tkinter elements visually to ensure their positioning and styling matches what the designer has in their mind. After a user interface has been created within this program, the user is able to export the code snippet to paste wherever it is needed. This program allows for rapid prototyping of user interfaces and easy organization and stylization of UI elements, and was used extensively during the development process for both the home window and the data window. Tkinter and matplotlib are also compatible when importing the appropriate backend libraries for matplotlib which makes plotting the

measurement data straightforward and customizable. Another reason why Tkinter was selected over the other user interface libraries was the tree-like organization of user interface elements. Separate windows for the error and data windows were able to be created and properly closed easily because of this, as well as accessing any element of the user interface from anywhere in the software.

3.3 - Component and Part Selection

In this section we will elaborate on parts we used in our design. We will discuss their functions, specifications and characteristics. We will also discuss how we came to the decision to choose each of the parts that were eventually chosen. And what alternatives may have been possible to substitute in the project.

3.3.1 - Laser Attenuator

The laser attenuator section of the device is the key to being able to measure the beam properties. Since this device will be able to work with high-power lasers (5W - 100W), it needs to be able to reduce the power enough to be captured by a camera sensor. For an input laser power of 100 Watts, the device needs to attenuate the power by many magnitudes (3 to 10). The attenuation method we will use is 3 optical wedges, N-BK7 wedge prisms from Edmund Optics (**Figure 15, Table 16**).

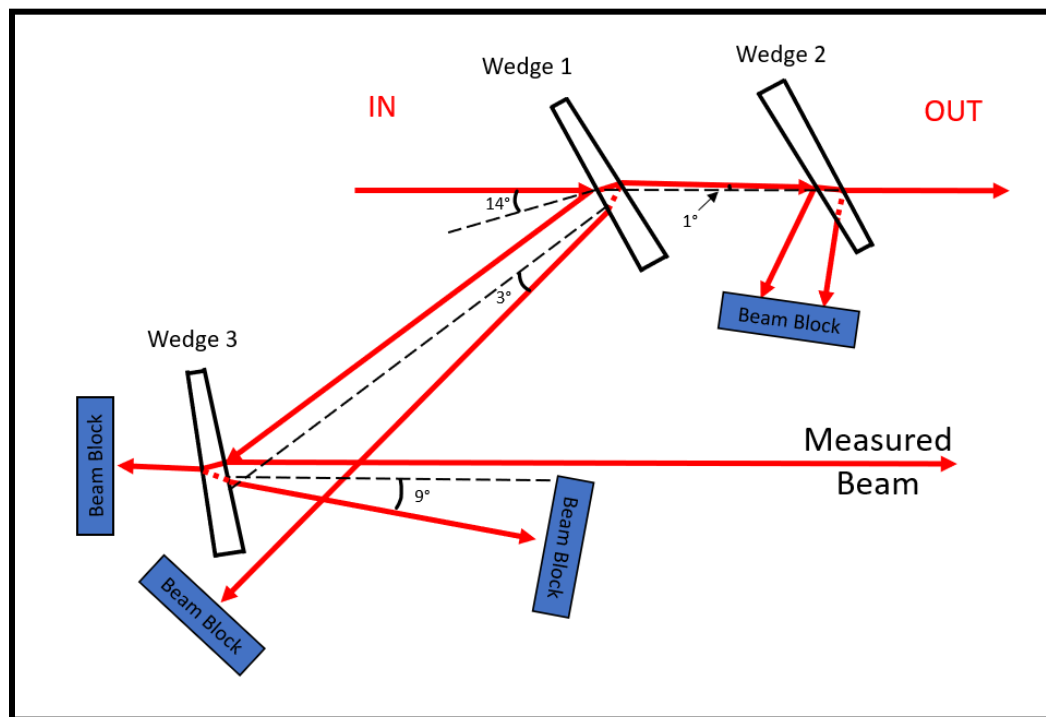


Figure 15: Original wedge attenuator design.

N-BK7 Wedge Prism Specifications (1 & 2)		N-BK7 Wedge Prism Specifications (3)	
Coating	Uncoated	Coating	Uncoated
Substrate	N-BK7	Substrate	N-BK7
Diameter	25.00 mm	Diameter	25.00 mm
Ray Deviation (@633nm)	1.00°	Ray Deviation (@633nm)	2.00°
Thickness	3.00 mm	Thickness	3.00 mm
Wavelength Range	300 nm - 2200 nm	Wavelength Range	300 nm - 2200 nm
Wedge Angle	1° 56'	Wedge Angle	3° 52'
Max Edge Thickness	3.8 mm	Max Edge Thickness	4.7 mm

Table 16: Specs for the N-BK7 wedge prisms used.

Optical wedge prisms allow for a large percent of attenuation. 2 wedge prisms allow for an attenuation percentage of 0.167% to 0.202%, depending on the wavelength and angle of incidence. A third wedge is required to realign the beam after passing through the first wedge. The beam will be deviated 1 degree after the first wedge. The compensation wedge will be placed 40 mm away from the first wedge, and will be rotated 180, thus having the same angle as the first wedge. The beam will experience dispersion in the first wedge, and will cause the beam deviation exiting the wedge to be different depending on wavelength. This deviation will be compensated for with the second wedge, with the downside of the beam being deviated perpendicular to the axis by a small amount. This orientation is the same as allowing the beam to pass through a thick piece of glass. Both surfaces are parallel to each other so the beam will still travel in the same direction but will be displaced. The second attenuation wedge (wedge 3 in **Figure 14**) was chosen to have a 3° wedge angle. This causes the ghost reflection beam to exit the front surface of the wedge at a 9° angle with reference to the front surface reflected beam. This is important in order to send it to a beam block in a short distance without getting in the way of the ND filters.

The material that the wedge prism is made from was determined based on the wavelength and laser power requirements listed in Section 2.3, **Table 1**. The wavelength requirements for this device are 355 nm - 1100 nm, and the power requirements are 5 W - 100 W. N-BK7 was chosen as the glass material because it provides a large spectrum of transmittance, and has a high damage threshold. The transmission spectrum for N-BK7 is shown in **Figure 16**.

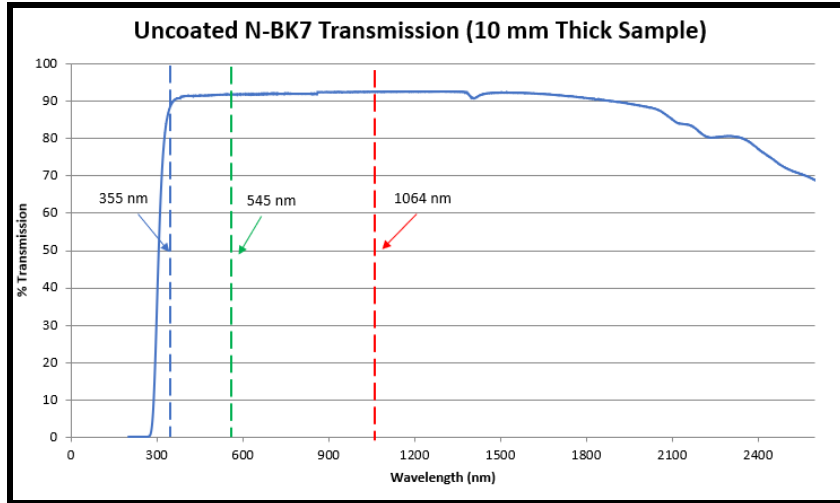


Figure 16: Transmission of uncoated N-BK7. 3 laser wavelengths are shown by the dashed lines. Transmission from 355 nm - 1100 nm is >90%.

Since the transmission is greater than 90% for the wavelength range we need, the reflectance will be low. This is desirable in order to attenuate a laser beam. For a CW laser beam, the damage threshold of N-BK7 is essentially the melting point of the optic. The absorption is very low for the wavelength range so even a 100W laser beam with a diameter of 3 mm will not cause damage to the material.

The reflectance of the wedge prism is dependent on wavelength as well. For different wavelengths, the refractive index varies. This difference is what causes the S and P polarization reflectances to vary based on wavelength. **Figure 17** demonstrates this using **Equations 2** and **3**.

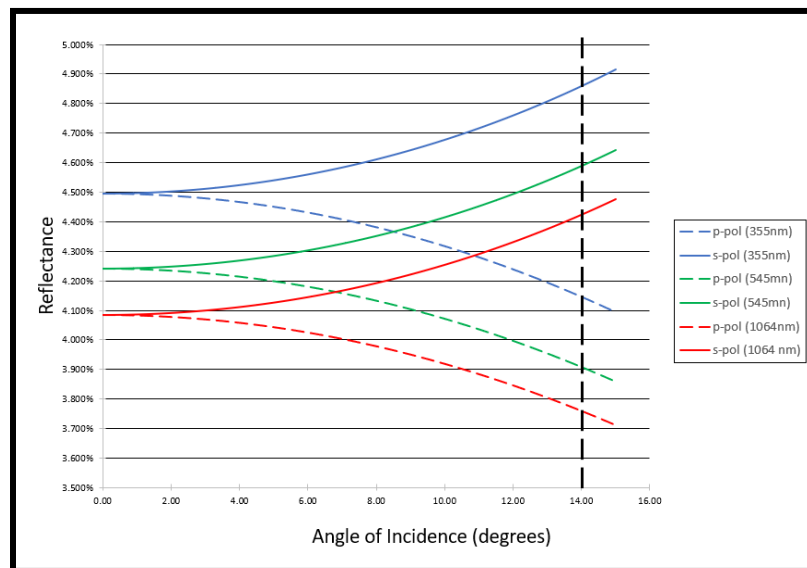


Figure 17: N-BK7 reflectance percentage vs angle of incident for 3 wavelengths: 355 nm, 545 nm, and 1064 nm. Graph plotted from public domain source: refractiveindex.INFO database.

Taking the average between the s-polarization and p-polarization, we get a reflectance for 355 nm, 545 nm, and 1064 nm, as 4.499%, 4.243%, and 4.086% respectively. These values are taken into consideration when calculating the total laser attenuation.

As stated above, this gives an attenuation range between 0.167% - 0.202%. This is still not enough in order not to damage the camera sensor when focusing the beam. Further attenuation can be achieved by placing ND filters in series before the focusing lens. The ND filters we plan to use are from Thorlabs and are the ND15A, ND30A, and ND40A. The ND filters come mounted in an SM1 threaded mount for easy assembly. Since the ND filters are wavelength dependent, the transmittance for the 3 wavelengths (355 nm, 545 nm, and 1064 nm) was calculated and shown in **Table 17**. The maximum intensity that an ND filter can take before it causes thermal lensing is $5 \frac{W}{cm^2}$. Since the power range for this device is 5W - 100W and the beam diameter range is 3 mm - 12 mm, we can calculate the maximum and minimum intensity. The maximum intensity after exiting the double wedge attenuator is $2.864 \frac{W}{cm^2}$. The minimum intensity after exiting the double wedge attenuator is $7.381 \frac{mW}{cm^2}$. This range is sufficient to not damage any ND filters placed between the wedge prism attenuator and the focusing lens.

		Transmittance		
		OD = 1.5	OD = 3.0	OD = 4.0
Wavelength	355 nm	2.6264%	0.1035%	0.000944%
	545 nm	2.7318%	0.1018%	0.000547%
	1064 nm	3.9721%	0.2330%	0.1027%

Table 17: Transmittance percentage of 3 ND filters used vs laser wavelength.

The ND filter combination used consists of an ND1.5, ND3.0, and ND4.0 arranged in that order, but can be moved in and out of the beam path to create a variable attenuation. For a wavelength of 355 nm, the ND filters provide a maximum attenuation percentage of $2.57 \times 10^{-7}\%$. For a wavelength of 545 nm, the ND filters provide a maximum attenuation percentage of $1.52 \times 10^{-7}\%$. And for a wavelength of 1064 nm, the ND filters provide a total attenuation percentage of $9.37 \times 10^{-6}\%$.

3.3.2 - Beam Dump / Block

This device requires beam dumps and blocks to help deal with the unwanted reflected beams. This section will cover how the beam dumps/blocks will be made.

A safety concern that this device has is do with the unwanted light that is reflected off of each of the 3 wedges prisms. For low power lasers this would not be an issue, but since this device will work for 100W lasers down to a 3 mm diameter, this needs to be taken into account. **Figure 6** shows the reflections from a wedge prism. The first reflection being around 4% and the next reflection out the front surface is around 3.69%. For a 100W laser beam these beams would be 4W and 3.69W. The maximum intensity incident on the beam blocks will be between $53.57 \frac{W}{cm^2}$ and $58.31 \frac{W}{cm^2}$. These reflections pose a huge safety risk if not accounted for.

To solve this issue, we will place a beam block in front of each unwanted beam to absorb and scatter the light. We will use standard masonry bricks as our beam block. These bricks are able to take temperatures of up to 1000 degrees Fahrenheit and are fireproof. The brick surface is also rough, allowing for lots of scattering of the incident light. To increase the absorbance of the brick, a black paint can be applied. The paint we plan to use is high heat tolerant spray paint in matte black. The original purpose of this paint by Rust-Oleum is for automotive engines. The combination of the brick and the paint should provide a good beam block for the unwanted laser light. This paint will also be used to coat the inside of the device enclosure. This will ensure that the metal casing does not just reflect the scattered light, but helps to absorb it more.

For our final design, square ceramic sheets were used instead of the masonry bricks. These sheets can still take the high temperatures and are much lighter and compact than bricks. Holes were drilled in the sheets to be used as apertures in the device.

3.3.3 - Focusing Lens

The lens is used for focusing the laser beam in order to calculate the M^2 beam quality. The focal length of the lens is an important parameter for other components for the device. The focal length of the lens will determine the total distance that the translation stage will need to travel in order to get an accurate M squared measurement. A large focal length lens creates a small divergence angle. This leads to the Rayleigh length to be larger. In order to get an accurate M squared measurement, we need to take beam diameter measurements at the minimum waist but also at a location at least one Rayleigh length away.

This means that the focal length of the lens determines what stage we can use, and vica versa. The translation rail will be discussed in **section 3.3.6**. The lens we chose for this device is 250 mm EFL, uncoated N-BK7 plano-convex lens

from Edmund Optics (#67-150). N-BK7 was chosen for the same reason as it was for the wedge prisms. A larger focal length was chosen for a couple reasons. A large focal length will produce a larger focal spot, **Equation 5** and **6**. This is important to reduce the risk of over exposure or damaging the camera sensor. A large focal length and a plano-convex design will help to reduce spherical aberrations. Spherical aberrations can cause the outer rays of a gaussian beam to focus at different points along the optical axis. If too much spherical aberration is introduced, it will cause the beam measurements to be inaccurate. **Figure 18** shows the spherical aberrations for the focusing lens for 355 nm, 545 nm, and 1064 nm.



Figure 18: Sum of the spherical aberrations for 355 nm, 545 nm, and 1064 nm. Wavefront error is in units nm and separated by surface 1 and 2 of the lens.

The chromatic aberration from this lens is taken into account as well. If not accounted for, chromatic aberration would cause the results to be very inaccurate. **Figure 19** shows the focal shift due to chromatic aberrations for 355 nm, 545nm, and 1064 nm.

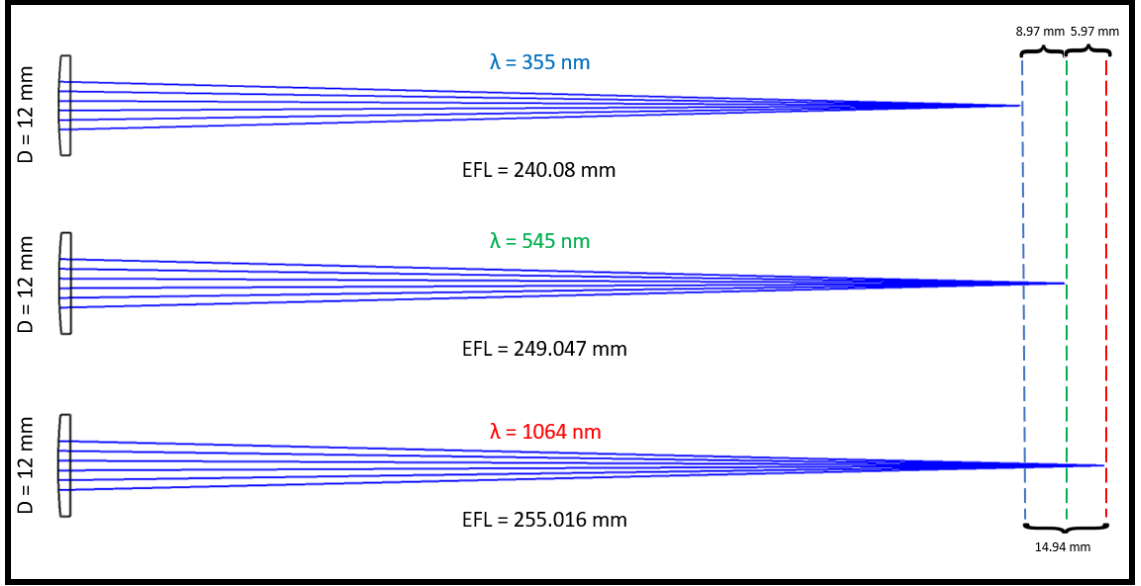


Figure 19: Zemax ray trace for the focusing lens for 355 nm, 545 nm, and 1064 nm using a beam diameter of 12 mm. Total focal shift of 14.94 mm.

Having a total focal shift range of 14.94 between 355 nm and 1064 nm, the range of the translation stage will still be able to cover and measure accurately and in compliance with ISO 11146-1 standard. The full range of the translation rail described below is 100 mm. To measure the M^2 beam quality, the beam diameter needs to be measured close to the minimum waist and at a distance of at least 2 Rayleigh lengths away. The Rayleigh length and focal spot size was calculated using this lens for each wavelength using the following equations:

$$\omega_0 = \frac{\lambda}{\pi\theta} \quad (5)$$

$$\theta = \tan^{-1} \left(\frac{0.5D_{beam}}{f_{EFL}} \right) \quad (6)$$

$$z_R = \frac{\pi\omega_0^2}{\lambda} \quad (7)$$

where ω_0 is the minimum beam waist, λ is the wavelength, θ is the far-field divergence angle, D_{beam} is the diameter of the beam at the focusing lens, f_{EFL} is the effective focal length of the focusing lens, and z_R is the Rayleigh length. Rayleigh length as a function of beam diameter before the focusing lens can be seen in **Figure 20**.

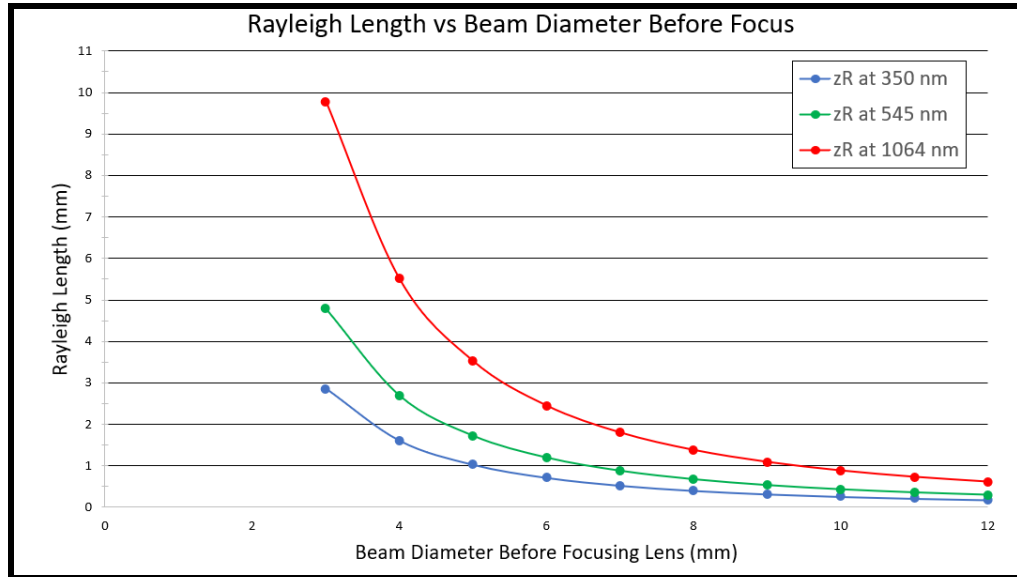


Figure 20: The Rayleigh length for 355 nm, 545 nm, and 1064 nm as a function of beam diameter before the focusing lens.

The longer wavelengths will require the camera to travel a larger distance. A similar graph is shown in **Figure 21** to show the axial position of where the camera will need to be. The solid lines show the Rayleigh length position with respect to the focusing lens, and the dashed lines represent the focal spot positions for each wavelength. The total traveling distance for one side of the beam waist for the camera is calculated to be 20.73 mm. For the full measurement this value needs to be doubled to account for both sides of the minimum beam waist being measured. The final travel distance needed for the translation stage is 41.46 mm.

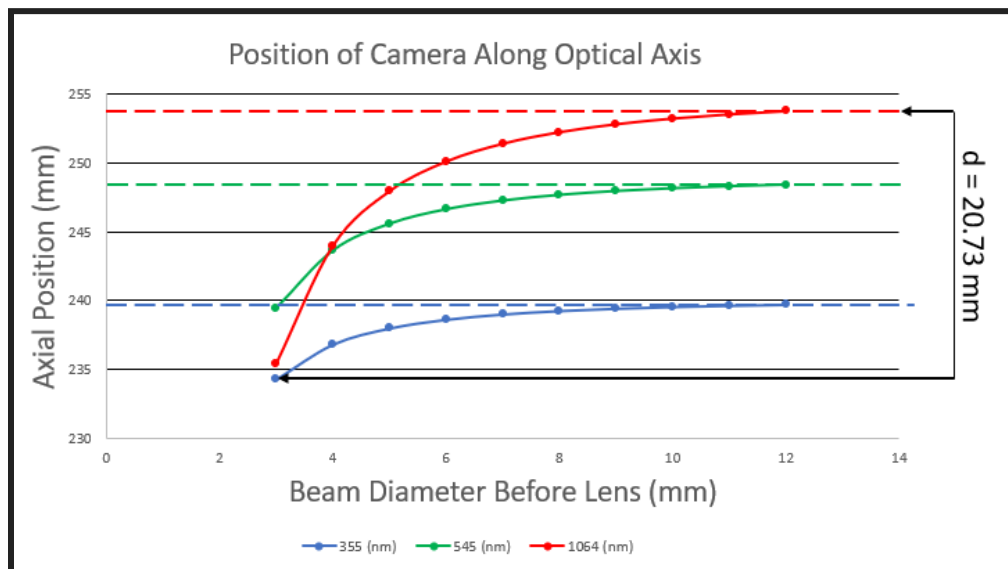


Figure 21 Graph showing the rayleigh length position along the optical axis plotted for each wavelength and beam diameter.

3.3.4 - Optical Telescope

The optical telescope is essential for imaging larger beam diameters. This is due to the small sensor size, which is covered in section 3.3.5. From this the maximum beam diameter that can be measured without the optical telescope is 2.5 mm. To reduce the beam diameter before the sensor, a secondary lens is needed after the focusing lens. An uncoated plano-concave 12 mm lens #45-016 from Edmund Optics was chosen. This lens has an effective focal length of -24 mm. This focal length was chosen to provide a magnification of -10.41 when combined with the focusing lens. **Figure 22** shows the two lenses. This optical telescope was not completed for the final device but has the option if programmed in the future.

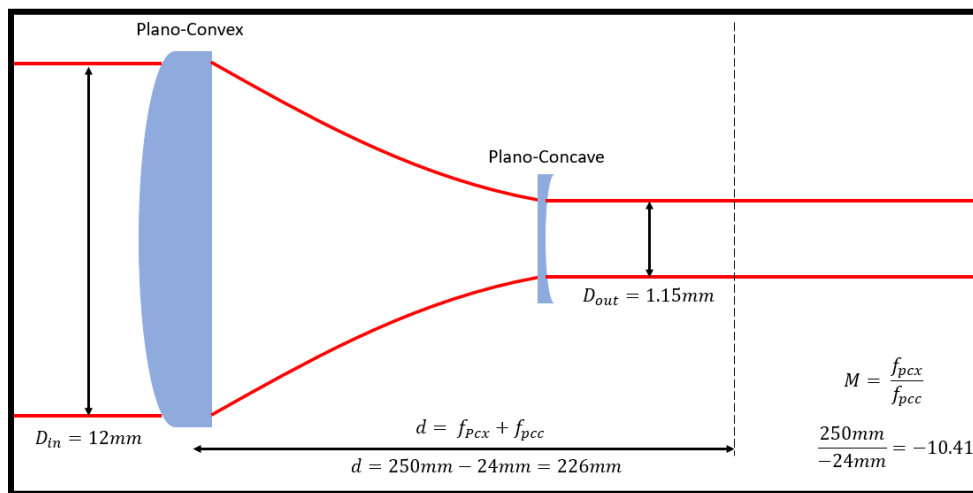


Figure 22 Beam expander diagram with convex and concave lens producing a demagnification of 10.41 times.

3.3.5 - Camera Sensor

The camera is a very important part of this device. The camera is a common point for the optical engineer, electrical engineers, and computer engineer. Picking the right camera depends on what wavelength range we are looking to capture, the max laser power or how much we need to attenuate the laser beam, and the beam diameter. A common sensor material is silicon, which has a spectral sensitivity of 300 nm - 1100 nm. This makes this type of sensor ideal for imaging in the visible and in the near-IR range of the electromagnetic spectrum. The camera that we chose for our device is the Raspberry Pi Camera Module v2. The specs for this camera are shown in **Table 18**.

This 8 megapixel camera has a pixel size of $1.12\text{ }\mu\text{m} \times 1.12\text{ }\mu\text{m}$. This is a good resolution for measuring the minimum waist of a laser beam. The pixel size should be at least 10 times smaller than the minimum waist for an accurate measurement. The sensor area is a rectangle ($3.68\text{ mm} \times 2.76\text{ mm}$), but due to laser beams being circular we will only use a square sensor area. The sensor

area we will use is a 2.76 mm x 2.76 mm area. This means that the largest waist that can be incident on the sensor cannot exceed 2.76 mm. There are two locations along the focusing beam that we will need to properly measure the M^2 beam quality: the minimum beam waist, and the beam waist two Rayleigh lengths away from the minimum beam waist. For each wavelength, the minimum waist and twice the Rayleigh length waist are plotted for the device's range of beam diameters (**Figure 23**).

Camera Module v2 Specifications	
Still Resolution	8 Megapixels
Video Modes	180p30, 720p60 and 640 x 480p60/90
Sensor	Sony IMX219
Wavelength Range	300 nm - 1100 nm
Sensor Resolution	3280 x 2464 pixels
Sensor Image Area	3.68 mm x 2.76 mm
Pixel Size	1.12 μm x 1.12 μm
Price	\$25

Table 18 Raspberry Pi Camera Module v2 useful specs [39]

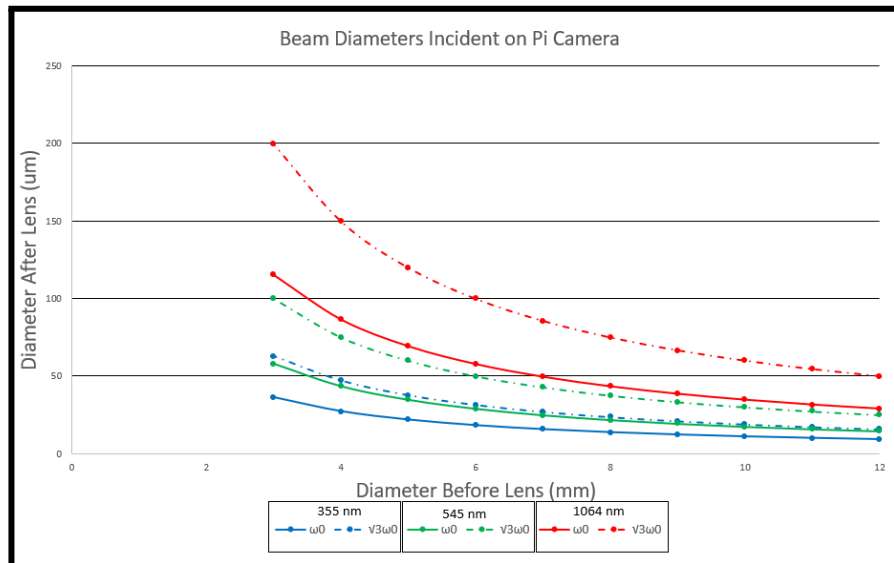


Figure 23: Plot for each wavelength of the minimum diameter and rayleigh length diameter as a function of input beam diameter before the focusing lens.

From **Figure 23** we can see that for a longer wavelength, the beam diameters are larger, and smaller for shorter wavelengths. This is important to take into account because smaller beam diameters result in higher intensities incident on the camera. A shorter wavelength laser will need a higher attenuation in order to not saturate the camera.

Another concern with the camera is the damage threshold, saturation intensity, and the SNR. Our device plans to be able to measure lasers with powers ranging from 5 W to 100 W. The previous section covers the laser attenuation to reduce the power as to not damage the sensor. The SNR can be accounted for by taking an image with no beam incident and subtracting that from the beam image. This ensures that there is no noise in the calculations.

3.3.6 - Optomechanical Mounts and Holders

The optomechanical mounts and holders are an important part of ensuring that the system is aligned and secured. Translation stages and rails are included in this section but are explained in the next section.

3.3.6.1 - Kinematic Mounts

The kinematic mounts are used to hold the wedge prisms. These mounts are the E-series kinematic mounts from Edmund Optics (#15-866). This mount can hold 25.0 mm and 25.4 mm optics and secures the optics using a securing screw. Although it allows for these diameters of optics, the clear aperture is only 23 mm. This will not cause an issue due to the largest acceptable beam diameter being 12 mm. Two axes can be adjusted and allows for 3.5 degrees of tilt. This is more than enough for alignment and maintenance. The mount has a securing screw for holding our optical wedges in place. The minimum thickness able to be secured is 2 mm. The minimum thickness for the wedge prisms used is 3 mm. This mount is able to be secured, using a M4 screw, into an optical post for height adjustments.

3.3.6.2 - Lens Mount

The lens mount in this device is the TECHSPEC thin lens mount from Edmund Optics (#13-787). This mount allows for 25.0 mm to 25.4 mm optics and secures the lens using a rationing ring. The minimum and maximum thicknesses of compatible lenses are 1 mm and 8 mm respectfully. The mount has SM1 threading which allows for additional lens tubes to be attached if needed.

3.3.6.3 - Iris Diaphragm

2 iris diaphragms will be used for this device. These irises are from Edmund Optics (#32-622). These irises will be used for alignment of a laser beam into the device. The max aperture this iris will provide is 15 mm. This aperture was chosen to ensure that a 12 mm beam will not be cut off at all. These irises are not mounted and will need to have custom mounts made to secure them in place.

3.3.7 - Translation Rail

The translation rail is an important part for measuring the M^2 beam quality. The step size of the translation rail will determine the accuracy of the device. The range of the translation rail will determine the minimum beam diameter able to be measured.

The translation rail chosen for our device is the Befenybay 100 mm length travel rail. As the name suggests it has a total travel range of 100 mm allowing for measurements on either side of the minimum beam waist for beam diameters down to 3 mm. The total length of this rail is 256 mm. This is mainly due to the rail coming with the NEMA 17 stepper motor which is 47 mm long and a motor flange length of 42 mm. Combined with the stepper motor described in section 3.3.8.1, this translation rail will have a minimum resolution of 0.025 mm. The repeat positioning is ± 0.03 mm. This low value ensures that the rail can return to the same location without causing the measurements to become inaccurate over time. With such a low repeat positioning value, the stage will not need much calibration throughout time. The mounting that this rail has is 4 M4 threaded holes positioned near the corners of the sliding rail piece. This will make the mounting tricky for the camera module. A custom camera mount will be made to fit the translation rail correctly.

3.3.8 - Raspberry Pi

In order to assess a beam for its M^2 beam quality factor using a CCD image sensor, large amounts of data would have to be moved from the sensor, managed in software, and later processed. Along with these data requirements, the software must be able to communicate with an outside system for interfacing, as well as driving a motor to move the camera sensor down the beam propagation path. The Raspberry Pi 3 B+ (RPI) was selected as the computing platform of choice for this project because it is able to fulfill all of these criteria while being accessible in the global supply chain crisis. Ideally, a Raspberry Pi 4 Compute Module would be preferred, as it would allow for a more integrated and flexible design, where most of the unused RPi functionality could be excluded.

However, due to the global supply chain crisis, all Raspberry Pi models were out of stock, or exorbitantly expensive from third-party sellers. In order to avoid increasing the cost of the bill of materials or risking not acquiring a RPi being sold at MSRP, this specific model was chosen due to it already being in possession. The RPi that we have chosen has several features and functionalities that we will not be taking advantage of. This model hosts several USB ports, an HDMI port, a 3.5mm audio jack, and a micro-usb port. In order to supply power to the RPi and have the RPi drive the stepper motor, the GPIO pin headers will be used to interface with the PCB hosting the power supply and stepper motor driver modules.

To interface with the Noir Pi Camera V2, the RPi hosts a Camera Serial Interface connector for connecting camera modules to the RPi. This interface is then made available in software through drivers accessed by the on-board Linux kernel. The image sensor data is handled by the operating system, and is made easily accessible in a development environment. The Linux-based operating system running on the RPi along with the kernel drivers allow camera image data to be manipulated easily in high-level programming languages such as C or Python.

Python was picked as the primary language for development due to the ease of interfacing with the on-board hardware in code, and providing convenient libraries for the creation and management of network sockets and handling large sets of numerical data. The RPi we have selected comes with 1 GB of DDR3 RAM, which is more than enough to handle the image data all at once. In order to facilitate the transfer of data between the computer and the RPi, the RPi hosts interfaces for both IEEE 802.11.b/g/n/ac wireless LAN and Ethernet over a CAT5e cable.

The RPi GPIO pins allow for flexibility in the design of the stepper motor and voltage regulator PCB, as well as the overall physical design of the device, as any of the GPIO pins may be used to drive the stepper motor. The RPi is able to be powered directly from the USB port coming out of the battery pack. The voltage regulator and motor driver PCB may be designed to sit right on top of the RPi, reducing the overall footprint of both boards, and allowing for a more compact device design.

3.3.9 - Motor

This section further details the selected parts for our motor design. This includes the NEMA-17 stepper motor and the TMC2208 motor driver.

3.3.9.1 - Stepper Motor

Looking further into the NEMA-17 brushless DC motor, it makes use of magnetic fields to drive a rotor through the magnetic fields produced in the stator coil windings as current passes through them. By activating the coils step by step the motor is able to attract or repel the rotor within the motor to align its magnetic field with the stator. The NEMA-17, therefore, is able to turn continuously or stop at any position. Reversal of the rotor is also possible with reversal of the current in each phase respectively. The NEMA-17 contains a rotor with 50 discrete steps surrounded by eight individual stator copper windings. These are configured in two phases which provide for four different magnetic field orientations.

This model features 200 steps with 1.8 degrees per step resolution. This allows for up to 0.04mm steps per subdivision on its own. We can achieve greater step precision through the use of a stepper motor driver. This is because the stepper driver will allow for incremental stepping of the energizing coils at different current levels which are percentages of the rated current we will be supplying the motor.

Depending on the stepper driver we choose, we can achieve various degrees of step accuracy. In order to correctly size the motors we will need equations for motor speed, position accuracy, Torque needed to turn the rail system, and pulses to rotate the actuator 1 revolution. Below are the equations we will use for that purpose with explanations.

How many pulses do we need to rotate the actuator 1 Revolution and how far does that push the actuator if 1 pulse= 1 step or 1.8 degree motor rotation? Position accuracy of distance/revolution can also be found from this equation.

$$distance \times \frac{revolution}{screw\ pitch} = \frac{pulses}{revolution}$$

Motor Speed required to move the rail system?

$$distance\ needed \times \frac{1\ revolution}{screw\ pitch} = Revolutions$$

$$\frac{Revolutions}{time\ in\ minutes} = RPM$$

How much Torque is needed? J=Inertia

$$J_{ttl} = J_{motor} + J_{attenuator} + J_{carriage}$$

Due to the torque specifications of our railing system we do not foresee an issue with sufficient power to move the translation stage. However, we will need to calibrate the device.

3.3.9.2 - Motor Driver

The driver that we are focused on using to power our stepper motor includes the TMC2208. This driver is used for two-phase bipolar stepper motors up to 2A peak. A big factor of why this specific driver was chosen versus the others we researched was due to its noiseless operation using StealthChop while still being cost-effective. Some other features include UART control, high dynamic motor control using SpreadCycle, and protection and diagnostic features. The voltage range for this driver is 4.75 to 36V with a +5V regulator and its step/dir interface has a 2, 4, 8, 18, or 32 microstep pin setting. It also uses microPlyer for a 256 microstep enhancement. The block diagram below in **Figure 24** shows the main components that make up the driver to operate the motor.

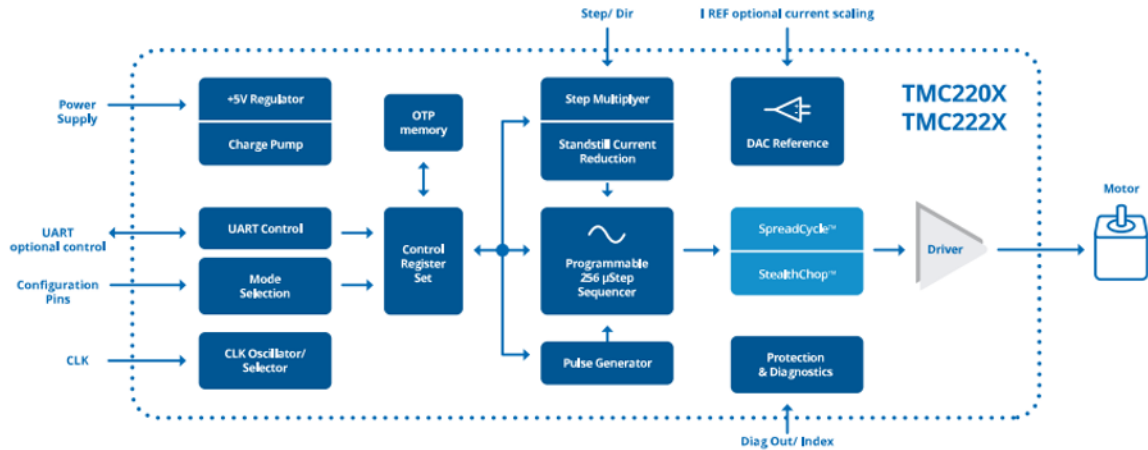


Figure 24. Block Diagram of TMC2208

These drivers' manufacturer is TRINAMICs and they do offer a universal evaluation board system as well. The dimensions of the driver IC is 5x5mm² while the eval board is 85x55mm². For our case we are more interested in purchasing the IC of the driver by itself instead of the evaluation board because we would like to create one PCB with all our components on it. However, the eval board is something we can keep in mind if we want to use their software for ease of testing our motor and changing certain settings to see what performs better prior to programming ours.

The driver works by the CPU generating step and direction signals synchronized to the motors. The TMC2208 operates the motor due to commands from CPU through configuration pins and STEP/DIR signals. The motor run current is either fixed or set by the CPU with analog input VREF. The PDN/UART will be at an automatic standstill for current reduction. The INDEX and DIAG output signals are used for feedback from the driver to the CPU. ENN pins are used to enable or disable the motor. **Figure 25** shows the layout of this IC with all of the pin configurations. **Table 19** further details the other pins not mentioned here.

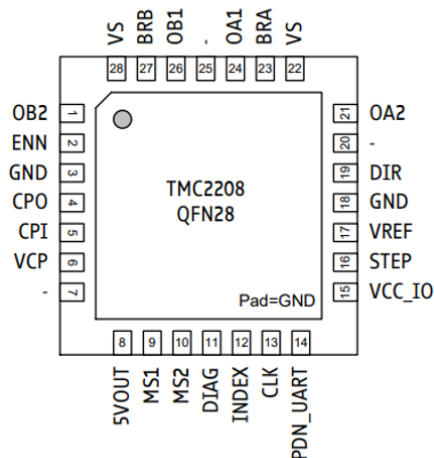


Figure 25: TMC2208 Pinning Top View

Pin	Number	Function
OB2	1	Motor coil B output 2
GND	3, 18	Connect to GND plane near pin
CPO	4	Charge pump capacitor output
CPI	5	Charge pump capacitor input, tie to CPO using 22nF 50V capacitor
VCP	6	Charge pump voltage, tie to VS using 100nF capacitor
N.C.	7, 20, 25	Unused pin, leave open or connect to GND
5VOUT	8	Output of internal 5V regulator, attach 2.2uF and 4.7uF ceramic capacitor to GND near pin for best performance
MS1	9	Microstep resolution configuration
MS2	10	Microstep resolution configuration
CLK	13	CLK input, tie to GND using short wire for internal clock or supply external clock
VCC_IO	15	3.3V to 5V IO supply voltage for all digital pins
VS	22, 28	Motor supply voltage, provide filtering capacity near pin with shortest possible loop to GND pad
OA2	21	Motor coil A output 2
BRA	23	Sense resistor connection for coil A, place sense resistor to GND near pin
OA1	24	Motor coil A output 1
OB1	25	Motor coil B output 1
BRB	27	Sense resistor connection for coil B, place sense resistor to GND near pin
Exposed Die Pad	-	Connect to GND plane for heat transfer for power drivers and analogue circuitry

Table 19: Other Pins Functions of TMC2208

The driver has three modes of operation including standalone STEP/DIR driver (legacy mode), standalone STEP/DIR driver with OTP configuration, and

STEP/DIR driver with full diagnostics and control. The first option listed works as described above with no UART connection. The second option allows for pre-programming OTP memory. This allows you to tune the chopper to the application, cost reduction by switching the driver to internal sense resistor mode, and adapting automatic power down level for better efficiency.

It is noted that the CPU we will be using will be the Raspberry Pi 3+ B and then from there use Bluetooth to communicate to a PC. This configuration can be seen below in the block diagram for **Figure 26**. The third option is similar to option 2, however, PDN_UART pin is connected to the CPU UART interface. This allows for detailed diagnostics and thermal management, passive braking and freewheeling for low power stop modes, more options for microstep resolution, and software controlled motor current settings. There are only three signals required for full control which include STEP, DIR, and PDN_UART instead of using ENN, DIAG, INDEX, MS1, MS2, and VREF.

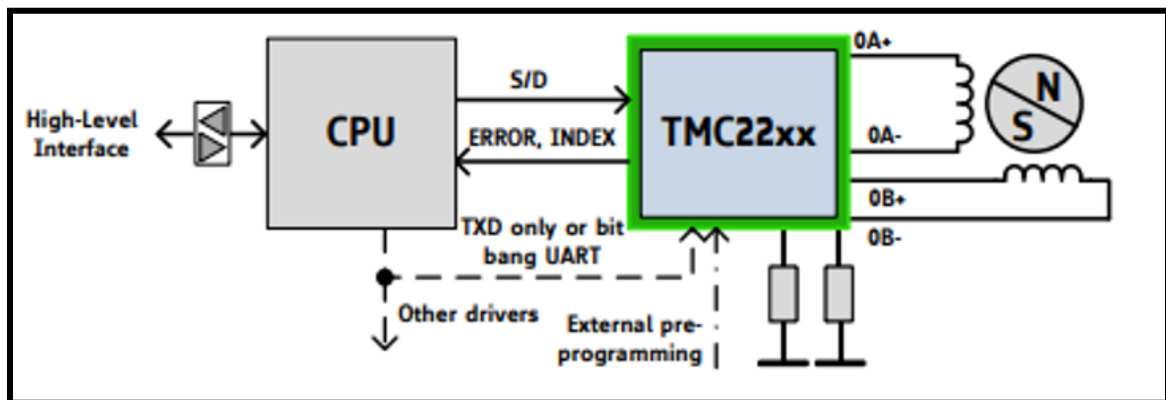


Figure 26: Standalone STEP/DIR Driver with OTP Pre-Configuration

The UART interface allows for both uni- and bi-directional operation for full control and diagnostics. A standard microcontroller UART can drive it with baud rates from 9600 to 500k or higher if use an external clock. There is a CRC checksum that allows data transmission over longer distances if needed. This is important since we are interested in using Bluetooth as our option for transmission to the PC. It also further details how to calculate CRC and if there are multiple slaves (which we will not have to worry about for our configuration). It also explains further how to program the UART interface through write and read.

There is also more explanation for moving and controlling the motor with the STEP/DIR interface, internal step pulse generator, StealthChop2 and SpreadCycle drivers, and precise clock generator. The STEP/DIR interface controls the motor through rising edges of both rising and falling edges controlled by the DEDGE bit for the STEP input. Each step can either be a fullstep or microstep ranging from 2 to 256 microsteps per fullstep. The internal step pulse generator is important to note because it is used when there is not a requirement for precise motion. However, we will need precision for our product, so this will

not be used. StealthChop 2 and SpreadCycle drivers however will be useful in programming our driver to reduce noise and have high motor dynamics including higher velocity. The internal clock generator allows for precise chopper frequency and performance. If quartz precision is required, we could also use an external clock. The clock also has timeout detection and automatic standstill power down to reduce power dissipation.

The datasheet further details sample circuits which include the standard application circuit, internal RDSon sensing, 5V output supply, configuration pins, high motor current, and driver protection and EME circuitry. The circuit we would be most interested in further detailing is the standard application which uses the minimum set of additional components including two sense resistors, low ESR capacitors for filtering the power supply, and minimum capacity of 100uF near the driver. It also can either be supplied by a 5V output regulator or an external regulator. We will most likely use the internal regulator.

It is also important to keep in mind thermal properties, it is recommended to not go above 1A RMS for motor current for increased periods of time. The other sample circuit to keep in mind was the driver protection and EME circuitry. These further details how to avoid ESD events by connecting all conductive parts to PCB ground. Capacitors can be placed at driver outputs or even introducing LC filters to decouple the driver outputs from the motor connector as seen below in **Figure 27**.

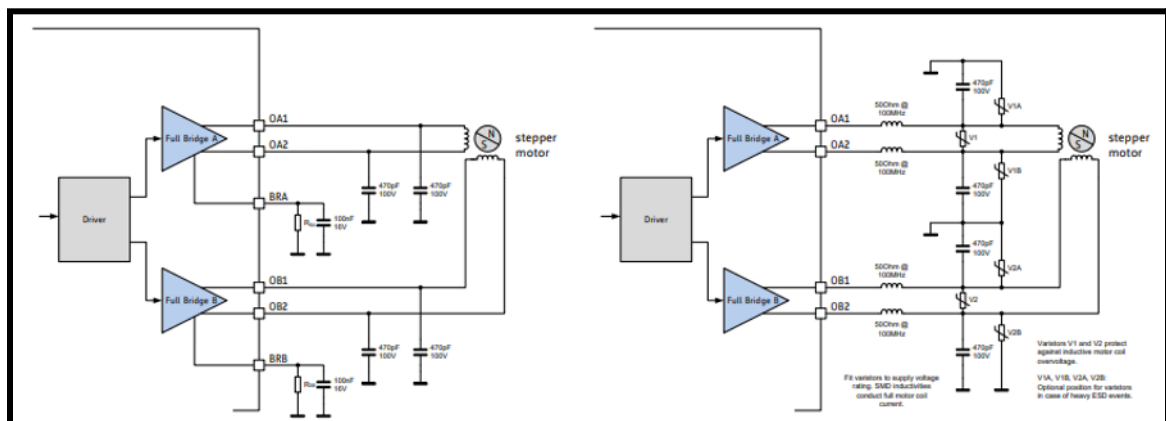


Figure 27: Filter Suggesting to Prevent ESD Events

One critical part of the datasheet is the register map. This outlines the complete register set which includes the general configuration registers, velocity dependent driver feature control registers, and chopper registers. The general ones include global configuration, global status flags, OTP read access and programming, and interface configuration. The others further detail driver current control, setting thresholds, optimizing StealthChop2 and SpreadCycle, driver diagnostics, and driver enable/disable. The table lists each register including its read/write function, address, name, bit, and description.

The datasheet further explains how to program StealthChop, motor current control, STEP/DIR interface, internal step pulse generator, driver diagnostic flags, and clock oscillations. It also lists the internal sense resistors to use. Another important factor to look at with our PCB design include the absolute maximum ratings, electrical characteristics, and the layout considerations. The absolute maximum ratings are what the circuit can max operate at and should not be exceeded. Some of them are listed below in **Table 20**. It is important to test for these when we are building and using our motor. The recommended considerations include the exposed die pad, wiring GND, and supply filtering. An example layout configuration including the schematic and PCB placement is given and will be further detailed in the design section.

Parameter	Symbol	Min	Max	Unit
Supply Voltage Operating with Inductive Load	Vvs	-0.5	39	V
I/O Supply Voltage	Vvio	-0.5	5.5	V
Logic Input Voltage	Vi	-0.5	Vvio + 0.5	V
VREF Input Voltage	Vvref	-0.5	6	V
Max Current to and from digital pins	lio		+/-10	mA
5V Regulator Output Current	I5vout		25	mA
5V Regulator Continuous Power Dissipation	P5vout		0.5	W
Junction Temperature	Tj	-50	150	°C
Storage Temperature	Tstg	-55	150	°C
ESD Protection for Interface Pins	Vesdap		4	kV
ESD Protection for Handling	Vesd		1	kV

Table 20: Absolute Maximum Ratings for TMC2208

The recommended schematic and layout designs for the TMC2208 are provided in the datasheet. These are helpful to consider when creating our schematic because they have been researched and tested already. These designs keep in mind supply filtering, wiring to GND, and the exposed die pad. We will use these as a reference for our design.

3.3.10 - Voltage Regulator

Due to our unregulated DC voltage source coming from the battery pack, we will need to introduce a voltage regulator circuit into our printed circuit board design along with our stepper driver, microcontroller, and raspberry pi. This is because most of the electrical components chosen for this project are rated for voltage at 5V to 3.3V continuous DC. This means we will run the risk of burning them out if we try to power them above their voltage rating. Our battery will provide variable voltage as it loses charge, which could cause unexpected problems in the system. This is extremely important to remedy, since without access to another Raspberry PI caution should be taken to ensure it continues to function for this project.

We were interested in using the TPS56628 from Texas Instruments for various reasons. It is a synchronous step-down converter which takes in a range of voltages from 4.5-18V and outputs a range of voltage outputs from 0.76-5.5V. It has a max continuous output current of 6A.

The device also features built in short-circuit protection, and thermal shutdown in the event of excess heat collection. With an operating temperature of up to 100 degrees celsius we don't foresee that being an issue for us. Within the overall design schematic the TPS56628 serves to step down the voltage in conjunction with two 100nF, 10uF, and 47uF capacitors on the inputs and outputs. This will serve to smooth out the output voltage even further to give us a constant 3.3V DC output. Below in **Figure 28**, we can see that this particular circuit design produces high efficiency according to the webench design report at about 86.1% for a 12V/6A source.

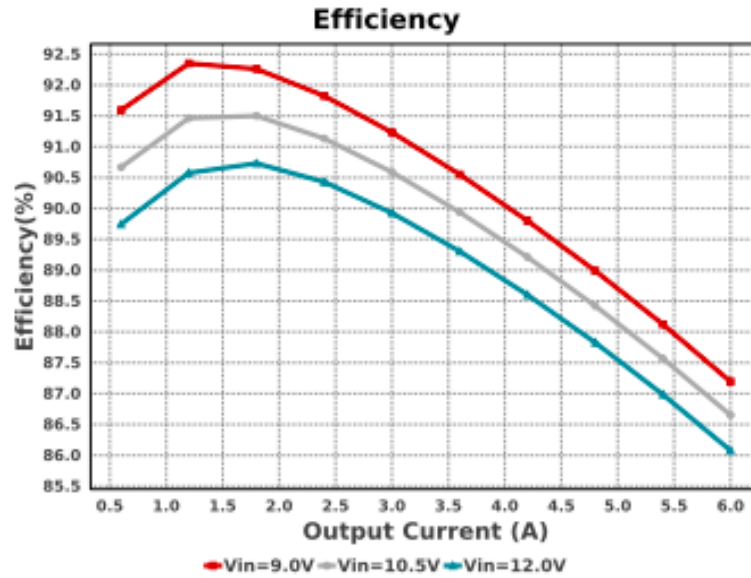


Figure 28: Efficiency vs Output Current Graph

For the input/output current graph however we do not see a very significant increase in the current output. This seems to be in line with the hand calculations that we performed earlier to determine the current that would need to be pushed through the voltage regulator to the microcontroller. We will need to test these output graphs against our final PCB to ensure accuracy. This can be easily accomplished through testing the positive and negative output leads of the switching circuit once we have the physical design produced. Below in **Figure 29** you can see the output current as a function of the input current within our voltage range.

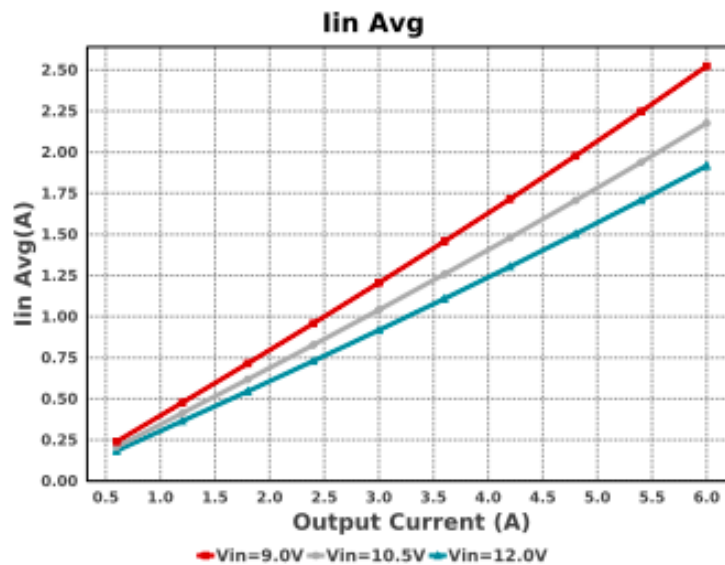


Figure 29: Input Current versus Output Current Graph

3.3.11 - Cooling Fans

For our design it is important to manage the thermal output of the electronic components to prevent overheating and system failure. In order to determine the appropriate fan for our application we will first need to assess our current design and answer which of the components are the most critical to keep cool. We can gain insight into the maximum ambient temperature ranges for our electronic components through analysis of the datasheets for each device. Once we know these values we can make calculations on the power dissipation for all components in the design overall.

In our design the raspberry pi, and the PCB are the primary components that are critical to keep cool for proper system operation. The motor running at rated current, won't generate enough heat to be of concern under the conditions we will be using it. Since the motor will not be continuously running for long periods of time in ambient temperatures above room temperature. Therefore we will be looking at the max temperature ranges for the Raspberry Pi 3+ B and the TMC2208.

Analyzing the Raspberry Pi 3+ B's datasheet we can see that the maximum temperature threshold is 85 degrees celsius. However, the Pi was designed to be run without a heatsink under normal conditions. This is because the processor will adjust the clock speed to keep temperatures within a safe range. However, since we plan to mount the TMC2208 PCB on the top of the Raspberry Pi we will need to be certain that excess heat will not cause the Pi to force a shutdown.

Using the simplified equation below we will plug in the maximum permissible temperature, and heat dissipated by the components. This will give us the heat generated or in other words, the heat to be dissipated(q) in Watts. We will then use the result to determine the airflow(Q) required in kg/second to dissipate the heat generated. The values in the table below were calculated based on an ambient temperature value assumption of 26 degrees celsius.

$$Q = .05 \times \frac{q}{\Delta T}; \text{ Airflow in Cubic Meters per Minute}$$

Using these figures, we will now plot a graph showing the Airflow needed over time as ambient temperature rises around the device. Assuming that the heat dissipated will continue increasing the temperature surrounding the device until the level will stop efficient heat removal. The inverse relationship between the change in temperature and airflow becomes apparent as we perform this analysis. In order to protect the system, we will take the smallest possible change in temperature, and multiply the airflow by a factor of 20%. This should keep the temperature from climbing. **Figure 30** shows the maximum change in temperature each component can attain at a given air flow. And **Table 21** gives the airflow required to cool each individual component to keep temperature constant under room temperature conditions.

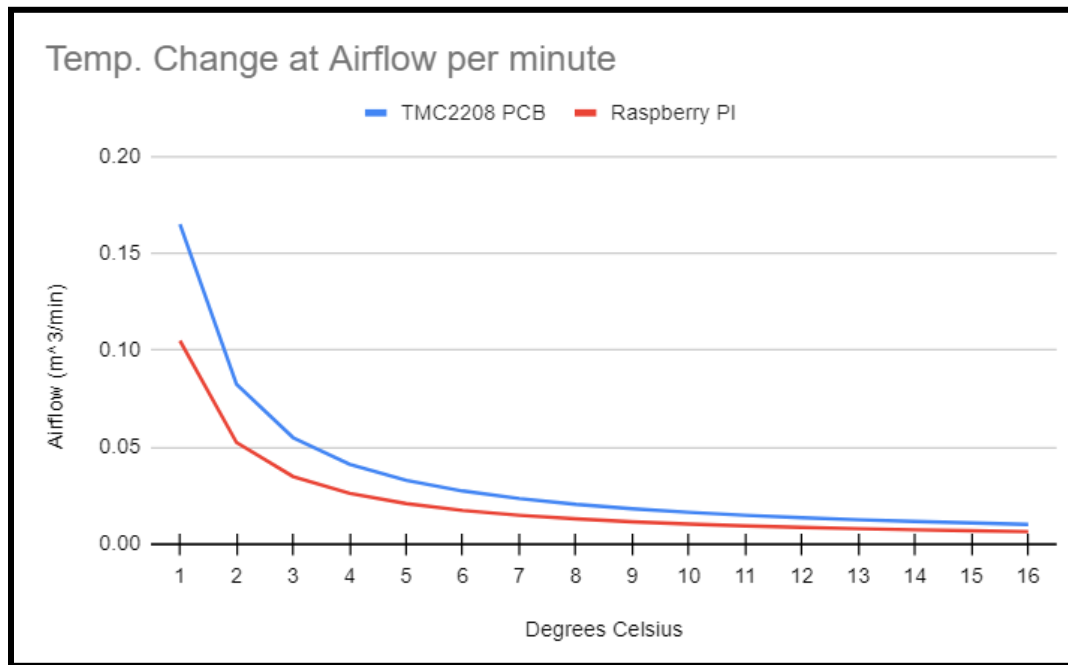


Figure 30: Maximum Change in temperature from ambient given airflow from fan(CMM)

Device	Max Temp.(C)	Heat Dissipation (Watts)	AirFlow Needed(CMM)
Raspberry Pi 3+B	85	2.1	.10*120%=.12
TMC2208	125	3.3	.165*120%=.198

Table 21: Airflow needed to maintain constant temperature at room temperature

Therefore we will use .198 m³/min of airflow minimum to size our fan, treating the system as an enclosed unit. This should ensure that we keep the components well below the maximum operating temperature. The Calculated Airflow needed can now be compared against the specifications for various PC fans. Which will take into consideration the airflow in cubic meters per minute needed and static pressure(of air). We have access to 3 separate sizes of PC fans currently in our possession, so we won't need to wait for shipping or include them in our BOM.

When sizing our fans we will need a way to compare CMM(cubic meter per minute) maximums of air displaced. Therefore we will need to rely on the following equation for calculating the CMM, which relies on MPM (meters per minute) and area of the fans surface. As well as MPM which takes into account the rpm produced by the fan, and D(diameter).

$$CMM = MPM * Area \quad MPM = \pi(rpm)(D)(\frac{1}{60})$$

First we will look at the AD0812HS-A70 PC fan that is rated at 12V and .25A. The fan will also produce a noise level of 33dB when running at full speed. The fan is a medium size compared to the others with a diameter of .08m. Observing the CFM(cubic feet per meter) we see that it has a rating of 29.9 or .83m³/min when converted to CMM. [28] This is well above the .198 m³/min needed for our design.

Second we looked at the N5010B1 12V/.17A PC fan. The noise level at full speed was only 30dB with a diameter of .05m. With a CFM of 8 converted to CMM, we have .2265 m³/min. Compared to the AD0812HS-A70, the N5010B1 is preferred since it has a lower noise level and will use less power, while still providing the cooling we need.[29] Third we looked at the Cooler Master 12V/.37A PC fan. This fan came with a range of 15-35dB with a size of .12m in diameter. The CMM(max) we found to be .277m³/min, which is still within the specifications for our design.[30] **Table 22** below compares the dB range, size, CMM and Power ratings for each PC fan noted above.

<u>PC Fan</u>	<u>Noise(dB)</u>	<u>Power-Watts</u>	<u>CMM</u>	<u>Size(m)</u>
Cooler Master	15-35	4.44	.277	.12
N5010B1	30(max)	2.04	.227	.05
AD0812HS	33	3	.85	.08

Table 22 : PC Fan Comparison Table

All fans that we have in stock meet the CMM specifications we solved for above. Since power, and size should be minimized as much as possible for our design we will go with the N5010B1. We should be able to calibrate the fan to run at CMM of .198m³/min which should bring the noise level to around 26dB. Which is less than the dB rating of 30 for a human whisper.

3.3.12 - Microcontroller

When considering the microcontroller for our printed circuit board we looked at what we were familiar with which included TIs MSP430 microcontrollers. We had development boards for these, but we were also interested in looking at the microcontrollers that the Arduino development boards had as well. We came down to choosing between the MSP430FR5969 and the ATMEGA328P. The MSP430FR5969 operates with an input voltage of 1.8 to 3.6V with up to 64KB of nonvolatile memory. It had different serial communications including UART, SPI, and I2C. It also has 40 I/O pins. The ATMEGA328P has 32KB of flash memory, 23 programmable I/O pins, and an operating voltage of 2.7-5.5V. It also can communicate serially with UART, SPI, and I2C. While the MSP430FR5969 has

more memory and more digital pins for our purposes we did not need something this complex. Our microcontroller will need to read in inputs and write out outputs coming from the Raspberry Pi. Due to the operating voltage of the digital pins of the Raspberry Pi being at 3.3V and that the ATMEGA328P on the Arduino Uno is removable we decided to go with the ATMEGA328P. We also chose it because it still offers us all of the features we want, we can program it with the Arduino Uno, and we can gain experience with the Arduino IDE as well.

3.3.13 - PCB Layout and Design

When considering the design of our PCB we looked at thermal management, ground and power planes, decoupling and bypass capacitors, EMI filtering, and frequency response for power. Important parameters for our power supply include voltage, max current, voltage ripple, and heat loss at maximum current. It is important to use switching power supplies because they improve efficiency of converting voltages, reduce PCB footprint of power supplies, and reduce power ripple. We need to have a good PCB layout and efficient power distribution for our design.

Certain things to consider include choosing the right regulator which have previously been discussed. We are interested in using a switching regulator which uses MOSFET technology. If using a switching regulator, the output can be adjusted by altering the duty cycle of the Pulse Width Modulation (PWM). Efficiency also depends on the heat dissipation of the circuit. PWM switching causes noise or ripple in the output. It is important to keep the power isolated from critical signals because the switching currents can crosstalk. They also emit electromagnetic interference (EMI) noise. We can reduce EMI by filtering, reducing current loops, ground planes, and shielding.

We also need to keep in mind the thermal management of our power supply. The heat emitted comes from the component's power level, characteristics, and impedance. Switching regulators help reduce heat dissipation. We can also look at proper cooling methods including using fans or a heat sink. The heat dissipation is not even on the board, it is important to use thermal vias to help with this as well.

Ground and power planes offer low impedance paths for the power to transmit. The ground planes offer power to be distributed, reduce EMI, minimize crosstalk, and reduce voltage drops. Power planes however are used to transmit the power to areas on the PCB.

The importance of decoupling and bypass capacitors is to reduce impedance between power and ground. The power can cause ringing in the power rail which could lead to voltage drops so it is important to add these near the power pins. They help reduce noise in the system and must be connected between power and ground. We can use short traces as well as ground vias to minimize inductance. When choosing a capacitor, we should look at the capacitor value,

dielectric material, geometry, and location relative to the IC. It is important to note that when the board is powered from no load to full load we can get oscillations until the voltage settles to normal. We can tune the capacitors to help limit these oscillations.

4.0 - Engineering Standards and Design Constraints

This section will cover some engineering standards and design constraints for our beam profiler design. Standards include protocols that establish characteristics of product, process, or service. Constraints include economic, environmental, social, political, ethical, health and safety, manufacturability, sustainability, and many more.

4.1 - Engineering Standards

Standards are published documents that establish specifications and procedures. They include protocols that are universally used for product development. They help keep people safe, create reliable products, and maximize functionality. The Standards Development Organizations (SDOs) oversee the process of developing standards. Some SDOs include IEEE, IEC, and ISO. They are each run independently and maintain their own policies and processes. According to IEEE's website, IEEE standards follow a well-defined path from concept to completion.

They have five principles which include due process, openness, consensus, balance, and right of appeal. ISO is an independent, non-governmental international organization that brings experts to share knowledge and develop voluntary, consensus-based, market relevant standards. These help in the reduction of product failure through the implementation of quality control. IEC is a global, not-for-profit membership organization whose standards provides a framework for governments to build companies that buy and sell safe and reliable products. In this section we will discuss certain standards in place that our device must comply with.

4.1.1 - Optical Standards

ISO 11146-1:2021 Lasers and Laser-Related Equipment

This standard is for all products that measure beam waist, divergence angle, and beam propagation ratios. This standard only covers part 1, which consists of descriptions and test methods for stigmatic and astigmatic beams. Preparation consists of ensuring that the beam power losses are less than 1% of the total power, the lens is mounted to ensure no symmetric errors occur. Other precautions include reflections, interference effects, external ambient light, thermal radiation, and air draughts. The environment should also be monitored to not affect the beam power distribution. Pixel based devices should have a spatial

resolution of less than 1/20th the beam diameter, and accuracy of these devices depends on the SNR of the detector. Beam measurements should be fit to a hyperbola and calculated values found using equations in Clause 9.

ANSI Z136.7-2007 American National Standard for Safe use of Lasers

This standard mainly involves safe use with laser systems. This includes safe use in healthcare facilities, outdoors, in educational institutions, in research, development and testing, in manufacturing environments, and in entertainment, displays and exhibitions. It also covers safety protective gear. Safety protective gear includes goggles, barriers, and blocking curtains. For use with our device we recommend that the user follows this safety standard when operating the laser that will be measured using this device.

4.1.2 - Electrical Standards

IEEE 1625-2004 Rechargeable Batteries for Portable Computing

It is a standard that explains how to design, test, and evaluate a battery cell or battery pack. This standard provides guidance when it comes to designing and manufacturing lithium-ion and lithium-ion polymer rechargeable battery packs. These rechargeable battery packs can be used for portable computing. This standard helps to mitigate battery system failure for the user. It includes subsystem interface design responsibilities as well as education and communication materials for the end user.

ANSI C18.2M Part 1-2019 Portable Rechargeable Batteries Specifications

Defines standard performance tests and provides guidance to users who will be dealing with portable rechargeable battery products. Minimizes the proliferation of cell and battery types. It includes battery dimensions, terminals, standard charge method, polarity, and test conditions and procedures. Secondary batteries must be designated by a number (voltage of battery), followed by a letter (I for lithium-ion), followed by a sequential number (assigned according to physical dimensions, terminal configurations, and performance variables).

IEC 62281:2019 Safety of Primary and Secondary Lithium Cells and Batteries during Transport

This standard specifies test methods and requirements for both primary and secondary lithium batteries during transport for safety. It includes many types of testing methods including type, misuse, and transport. These tests include sampling, altitude, thermal cycling, vibration, shock, impact, short-circuit, and overcharge protection. Safety is also discussed and further details packaging handling cartons, transport (including air, sea, and land), and storage. This standard is not meant to be used for recycling or disposal requirements, only for transport.

IEC 61960-4:2020 Secondary Batteries Containing Alkaline or Other Non-Acid Electrolytes for Portable Applications

This standard specifies performance tests, designations, markings, dimensions, and other requirements for coin secondary lithium cells and batteries for portable applications and backup power supply. This allows users of these batteries to have a set of criteria of the performance of them. Multiple characteristics of secondary lithium batteries are discussed including voltage range, nominal voltage, end-of-discharge voltage. It also includes the cell designation and marking of batteries and electrical tests that can be performed.

IEC 61000 Series Electromagnetic Compatibility (EMC)

These standards include a series of documents over the years to explain further the importance and characteristics of electromagnetic compatibility when creating your product. One document from 2013 further explains the limits of voltage changes, fluctuations, and flicker for low-voltage supply systems. Another document from 2011 details the assessment of low frequency electromagnetic immunity and emission requirements in low voltage networks. A document from 1997 outlines emission levels, frequency bands, and electromagnetic disturbance levels for low voltage systems. These documents are helpful to look at when designing our PCB. The goal is to have the most ideal layout for our board design to have the least amount of electromagnetic emissions. We also want harmony between the different components. This can include placing things on different layers of the PCB, having a ground layer, spreading out certain components on the same layer, and increasing or decreasing the width of the signal routes.

IEC TR 62188:2003 Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes Design and Manufacturing Recommendations for Portable Batteries

This standard is more important for manufacturers when designing the batteries but can come in use for us to better understand how they are manufactured. It includes information on marking, dimensions, electrical characteristics, parameters for design, and recommendations of use of the battery. Basic parameters include number of cells, installation of overcharge protection, design, wiring, materials, and reversed batteries. The recommendations for use include charge, discharge, and handling precautions.

IEC 60445:2021 Basic and Safety Principles for Non-Machine Interface, Marking and Identification

This standard specifies identification and marking of terminals of electrical equipment. This applies to resistors, fuses, relays, contactors, transformers, rotating machines, conductors, and assemblies. The color and alphanumeric labels are further detailed to identify certain conductors including cables, cores, busbars, electrical equipment, and installations. These colors include single and bi-color combinations. While this document is mainly used by technical committees it is helpful for us to see how this process works since we are creating a product.

4.1.3 - Software Standards

PEP 8 – Style Guide for Python Code is a styling guide for python programs which follows the mentality that code is read more than it is written, and aims to provide better readability in order to improve programming efficiency and maintenance. The styling guide goes in depth on how long lines should be divided into multiple lines, while lining up similar values with another to improve readability, the naming conventions of functions and variables, and much more. **Table 23** outlines the styling guide's recommendation for common syntax and programming paradigms:

Topic	PEP 8 - Style Guide Recommendation
Line Length	• Limit all lines to a maximum of 79 characters • Docstrings or comments should be limited to 72 characters
Indentation	• No tabs should be used; 4 spaces for each indentation level
Blank Lines	• Surround top-level function and class definitions with two blank lines. • Method definitions inside a class are surrounded by a single blank line.
Imports	• Imports should be on separate lines, and are order by way of: 1. Standard Library imports 2. Related third party imports 3. Local application/library specific imports

Table 23: Common PEP8 Python styling recommendations

4.2 - Design Constraints

Design constraints include any limitations to our design. These limitations can include ones we can and cannot control. It is important to look at these constraints because they can affect the design, part selection, requirements, and user manual of our product. The constraints we detailed include economic, time, safety, manufacturability, sustainability, ethical, project requirements, and supply-chain.

4.2.1 - Economic Constraints

Economic constraints include the limitations of production, distribution, and consumption of goods and services. For our device certain constraints include the cost, availability, and quality of certain parts for our design. These all go hand

in hand since typically the higher the cost the better the quality. Availability also plays a role because even if you find the right part, it may not be available.

Availability of parts can play a role when it comes to economic constraints because if you need a certain part for your product and you are not able to get it then you will either find it somewhere else or change your design. The worst-case scenario is that you must change your design. This could affect the production, distribution, and consumption of your product. Changes to the design can cause production to take longer than originally anticipated along with limited part availability. We will need to keep this in mind since we have a deadline with this project.

The price of laser beam profilers can get very expensive. Just a beam profiler for M² beam quality and diameter can be around \$10,000, and that is not including all the attachments that allow for the ability to measure more powerful and larger beam diameters. The quality of the optical, electrical, and mechanical parts plays a large role in the price. For most market products, the quality of the device is seen by the price of that device. Our device will meet the gap between accurate/quality measurements, and an affordable price. We have a \$1,000 budget from CMS Laser to create this device. Our part selection needs to be very specific to meet the requirements from CMS Laser. This may mean using lower quality optics compared to market technologies, in order to stay within the budget.

4.2.2 - Time Constraints

Time constraints can occur in many ways including the amount of time we have for the project. The goal of this project is to give us an experience similar to the real world of engineering. It will allow us to prepare for our future career. Part of engineering is project deadlines. They do sometimes get moved around, however, they are set there to give some organization to the project along with having specific reasons for it being set. Since we are taking this class in the Summer of 2022, we only have 12 weeks compared to 16 weeks available in the Fall or Spring. We need to keep track of the deadlines that we set for ourselves to not get behind.

In the milestones section we have laid out the major deadlines we need to keep in mind. We also have set up weekly meetings to make sure we are staying on track with these deadlines. Keeping in mind that we have certain deadlines to meet, we also need to make sure the items we purchase are in stock and can arrive on time. Ways to tackle this constraint is to have backup parts available if something is not in stock or shipping will take too long. We also have chosen manufacturers with faster shipping times. We can also order multiple of the same part in case something happens to the part when we are building the device.

We have a goal and are on track to order all required parts by the end of Senior Design 1. We want to make sure that we will have enough time to build and

troubleshoot the device. Another thing to note is that building the design is not going to be perfect and as expected. We need to make sure we have enough time to work on building the device. The parts of this device that will require the most time is aligning and calibrating the device. We will also be testing things separately before placing them all together, including the motor and actuator, the raspberry pi, the laser set up, the camera, the programming of the raspberry pi, etc. This will be more detailed in the testing section.

The time for a full measurement is also a constraint for this device. The minimum time for a complete M^2 measurement is 12 minutes. This time could be improved if the image processing could be done all on the PC and if the image data could be sent faster.

4.2.3 - Safety Constraints

Safety constraints always play a big role in any engineering production. There are many safety standards available that have been created specifically to minimize hazards with parts and design. A lot of these safety protocols can be found in the standards section and include SDOs like IEEE, IEC, ISO, and ANSI.

When working with high power lasers, both CW and pulsed, user safety is our top priority. We need to take into consideration where the beam is traveling inside the device and ensure that there is no potential for damaging or injuring the user or user's equipment. One possible hazard is the scattering of laser light inside the system due to unwanted reflections. Our method to minimize and eliminate this safety hazard is to fully enclose the device, with only a laser IN aperture and laser OUT aperture. This will ensure that no scattered light will escape the device. Safety goggles rated for the laser being measured should still be worn during operation to remove any possible risks. We will also keep track of where the unwanted reflections are and ensure that they are not a risk to the user. To comply with the *ANSI Z136.7-2007* American National Standard for Safe use of Lasers standard, the device, although not containing a laser, will strongly recommend the user to use the correct safety goggles for the laser being tested.

Power supply is another aspect of safety, we need to look at specifications when it comes to rechargeable batteries being used for portable devices. The batteries must also provide sufficient power to sensors, motors, and processors. The power supply should be able to last for multiple tests, at least 6 hours of continuous use. This can include transportation, recycling, environment, testing, and labeling. Along with power supply includes voltage and current levels. We want to make sure we are reading datasheets of our electrical components, so they are powered properly. We do not want to short our circuit and overheat and risk anyone getting hurt. This is not only for our safety, but also will ruin our electronics.

4.2.4 - Manufacturability Constraints

Manufacturability constraints focus on the design of the product itself and limitations that occur with manufacturing. For our overall design, we will have constraints with manufacturing due to our budget limit of \$1000. Since this product is influenced by our sponsor, we have a certain budget to attain. If they would like to manufacture this product for commercial use, their cost for the laser beam profiler will have to be more than \$1000 due to manufacturing cost to be taken care of. We must keep in mind that even though we have a set budget, we should do our best to create a device that is cost effective as well. The good thing about our budget is that when we compared it to other existing products, we found the cost to manufacture our product was much less.

Along with cost being a manufacturability constraint, we also considered part availability. Due to COVID-19 concerns, manufacturing was highly impacted due to job loss and infection rates. Due to this, part availability is much slimmer and there will have to be changes in our part selection and possibly design which we did keep in mind. Shipping has also been delayed, sometimes up to an entire year or more to receive a part from the manufacturer. The manufacturers we choose may also discontinue certain parts or no longer be in business in the future. To help with these constraints we looked at multiple sites including digikey, mouser, amazon, etc to make sure we found a part we needed. When we created our bill of materials, we also had multiple backups for the parts we selected in case there were any changes to availability or shipping when we went to order them.

We also kept in mind that our device must be light and small enough to be portable. This means when choosing our device, we would like to make sure we are not choosing anything that is too large or too heavy. For example, when looking at the NEMA stepper motors, we found that they had different sizes for their motors including short, medium, and long. While the long ones can be more powerful, the short ones allow us to achieve what we need and are smaller, more compact, and lighter.

For our optical part of the design, we need to make sure the device can be aligned by hand, with micro-adjustments being done by precision stages. The optics required for this device can be easily found online and require no custom optics. This will ensure that the device will be easy to manufacture in large quantities if needed as long as there is availability for the parts.

If this device is going to be recreated to have multiple laser beam profilers for our sponsor, it is more convenient for our parts to not be outsourced by multiple manufacturers. We kept in mind our sponsors' connections with other manufacturers, so they are able to order directly from them as well as receive discounts for cheaper parts.

It is important to look at the certain things like climate and elevation for example temperature and humidity constraints that the parts have. Our sponsor is located in Winter Park, Florida which is a pretty humid environment, hot temperatures, and low elevation. While this device will be kept and used in doors, it is still important to make sure the user is aware of these constraints when storing and using this product.

4.2.5 - Sustainability Constraints

Sustainability constraints include any limitations when dealing with maintaining our product. Certain parts may need maintenance throughout the life of the part. It was important when building our bill of materials to choose parts with less maintenance and to outline what maintenance would be required.

The power supply was highly researched due to this being a big part of maintenance when considering electrical components. The power supply can also influence the other devices it is powering. One of the reasons we choose a power supply with a lithium-ion battery is because they have lower maintenance compared to other rechargeable batteries. Stepper motors may also have to be repaired or maintained after a certain amount of time which could include alignment checking, cleaning, and testing. It is important to look at these when we receive the motors so the user is aware as well.

While all optical parts should be secured, the device may need minor alignment maintenance. This is achieved by using a two-axis kinematic mount. All optics are mounted using optical posts and post holders that allow for vertical alignment and are secured to the base using clamping forks. All alignment procedures should be done using a low power laser, as to not risk damaging the maintenance personnel. A class 3R or lower laser should be used when the lid is open and maintenance is in progress. A class 3R laser will have a max power of 5 mW, which is sufficient when aligning the optical wedges, and will not damage the camera sensor. This laser power is still hazardous and if direct eye contact occurs, serious damage to the eye will be caused. The reflections and diffuse reflections are not enough to cause damage with this laser. It is also important to consider the calibration of the translation stage and camera position will require a laser with a known wavelength. This is so the camera can have a reference point along the optical axis to use for other wavelengths.

We are currently limited in looking further into details of sustainability constraints because we do not have our parts yet, they are still in the process of being ordered. Most parts come with an instruction for use that will better outline any maintenance requirements. For now, we are doing our best to have an idea of maintenance required through research of these different components.

4.2.6 - Ethical Constraints

While our specific device is not going to be used for commercial purposes, it is still important that ethical constraints are taken into consideration. Ethical constraints that we should consider during this project include honesty, integrity, promise-keeping, fairness, respect for others, responsibility, and accountability. Working in a group environment can be a challenge because there are many different types of people with different cultures, socioeconomic backgrounds, genders, sexual orientation, work habits. Something that may be normal to one person may not be for another. It is important that we work together as a cohesive team and look beyond things that we may not agree with when it comes to someone's personal life.

The goal of this project is to work in a team environment to produce a design that has different components of engineering. We are essentially simulating what we will be doing in our future careers. We also all come from different disciplines and may know more about a certain field than someone else would. Instead of seeing this as a bad thing we need to work together and find out what everyone's strength is and work with that while improving our skills that are not as great! Our group consists of two electrical engineer students, one optics and photonics engineer student, and one computer engineer student.

We will all have to work together to achieve a working product. The optics and photonics engineer is focusing on the laser beam set up along with calculations for the measurements. The electrical engineers will focus more on hardware for the rest of the system and the computer engineer will focus on software to program our device. Since we do not have a mechanical engineer, we will all be working together to design the look of our product.

Whenever someone is stuck or needs help it is important that they speak up and get support from their other team members. Sometimes personal matters occur and we will need to be understanding and continue on to keep up with the deadlines. One example is the electrical engineers were getting stuck on the design of the PCB. The computer engineer in the group however has more experience with PCB design and was able to help them with what they were struggling with. This is how we will grow as engineers is to remember we are always a student and to not be afraid to ask for help.

One of the biggest factors is accountability and making sure everyone is doing their part of the project. We are working through this by setting up weekly meetings along with having deadlines. If certain team members are not doing their part then we will not be able to succeed. Not only will this mean that other members will have to step up more, but it may be something they are not familiar with which means they will not be able to execute it as well. Our weekly meetings are set to make sure we are all contributing equally and to answer any questions that occur.

We will also have other meetings that we set up as needed if the one weekly meeting is not enough. The most important part is to remember this is a team effort not an individual effort. We will all be bringing together our expertise to create a complex measuring device. And in the workforce we will always be working with other engineers, so this is great practice for us to start understanding how important it is to keep up with our work.

4.2.7 - Project Requirement Constraints

Our constraints with project requirements include requirements given to us by our sponsor and requirements we have given for our design as well. It is important to set these limitations because they provide us with a guideline and map of where to start. These requirements help when we research, choose parts, and design the product.

Due to our project being sponsored by CMS Laser, there are certain requirements that they would like to see for our device. Overall, we have free range to approach the design however we want from an optical, hardware, and software point of view. The only design aspect that was requested was that it be light and small enough to be portable, preferably no larger than the size of a microwave. However, when it comes to what the device is measuring and how many measurements we need is where these requirements come from. This includes wavelength range, laser power, imaging, beam diameter measurements, M^2 beam quality measurements, measuring accuracy, and a budget of \$1000.

We have also set certain requirements for our device along with stretch and advanced requirements that we would like to achieve. This includes the measurement method, size, weight, power supply, computer interface, and computer connection. Our advanced and stretch features are included to give us goals if we have time to achieve to improve the device. We have included a couple of tables with these features we would like to achieve.

4.2.8 - Supply-Chain Constraints

Supply-chain constraints have changed drastically over the past few years. Due to global economic factors, semiconductor products have become scarcer and harder to obtain. Reliable manufacturers that we once went to are now struggling to keep up with the demands of their buyers due to manufacturing issues that arise when a global pandemic occurs. We saw this as a big limitation with both our optical parts and electrical parts.

The design is restricted to a Raspberry Pi 3 B+, due to all Raspberry Pi models being out of stock, and this is the one that was on-hand prior to this design. The plans to use a Raspberry Pi compute module on the same board as the other systems for a single PCB design had to be scrapped. Many industry-popular microcontrollers have also become impossible to order, such as the Microchip ATMEGA328P-P, as well as other common integrated circuit components such

as voltage regulators and interface ICs. We also looked at other sites like Amazon to get a lot of our electrical parts and were able to use prime for free and faster shipping.

4.2.9 - Design Constraints Comparison Table

The following table displays the economical, safety, manufacturability, and sustainability constraints of the market devices explained in **Section 3.1**. **Table 24** shows that our device should be an improvement in 3 out of 4 of the constraints.

Constraints	Market Technologies				Comments
	BP209 VIS	M2MS	SP932U	Our Device	
Economical	↓↓	↓↓	↓↓	↑↑	cheaper than market products
Safety	↓	↑↓	↓	↑	Fully inclosed device with added beam dump to decrease scattered light.
Manufacturability	↑	↑↓	↑	↓	Our device will take longer and more materials to build compared to market technologies
Sustainability	↓	↑↓	↓	↑	We will build a durable product and supply in-person maintenance

Table 24: Compares the constraints of market technologies with our device's constraints.

5.0 - System Design Details

This section will detail the optical, hardware, and software designs along with calculations needed for the optical design. We also have included an overall diagram of our intended design and look of the entire system.

5.1 - Optical Design

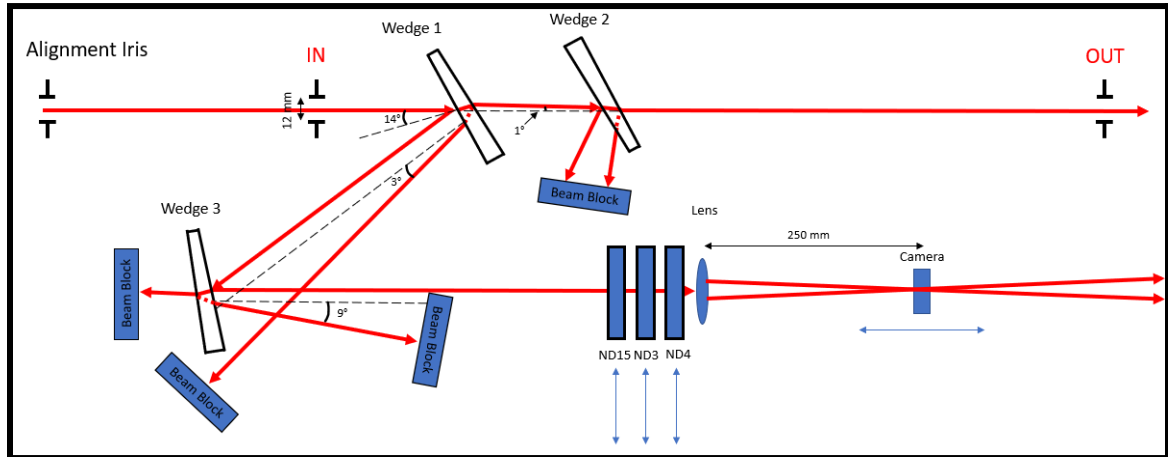


Figure 31: Overall optical design setup using geometric optics. Consists of 3 optical wedges, 2 apertures, beam dump, 3 ND filters, lens, and a camera.

The optical design will have an aperture hole that the input laser beam will enter the system, as well as an exterior alignment iris. The purpose of these two irises is to ensure that the beam is aligned into the system correctly. This beam then becomes incident on the flat side of a 1° wedge prism. This prism is rotated such that the angle of incidence (AOI) is 14° . The front fresnel reflection of this wedge is what is kept, the back surface reflected beam is sent to a beam block to be absorbed. The transmitted beam is sent out of the wedge at a 1° deviation where it is incident with another 1° wedge. This wedge is identical to the first wedge except that it is rotated 180 degrees. This is done to realign the beam so it is parallel to the beam that entered the device. The reflected beams from the second wedge will go to a beam block. The third wedge is used to further attenuate the beam and to realign it into the focusing lens. This wedge is a 3° wedge which causes the back reflected beam to exit the front surface at 9° compared to the front reflection. This is done to deviate the beam enough to block it without it hitting the ND filters. The transmitted beam is also sent to a beam block. The two front surface reflections are kept because they allow for low attenuation and they will not distort the beam before it is measured. This beam will then pass through several ND filters in series and attenuate the light to a level that will not damage the camera sensor. The ND filters will be mounted to manual slides that will allow for a variable attenuation adjustment by the user. The beam will be focused by a plano-convex lens. The beam will hit the convex surface first, then the plano surface to share the power between both surfaces. The translation rail will be mounted under the camera module. The rail will translate the camera to locations close to the focus, as well as 2 Rayleigh lengths away on either side of the focus. **Figure 32** shows the full range that the camera will need to travel using 3 laser wavelengths, 355 nm, 545 nm, and 1064 nm. The solid colored lines represent the focus of each wavelength, and the colored dashed lines represent the distance of 2 Rayleigh lengths away. The camera will

need to travel a full range of 40.34 mm. This means that there is still much unused rail that would allow for smaller beam diameters to be measured. The current limitation for smaller beam diameters is the laser input power.

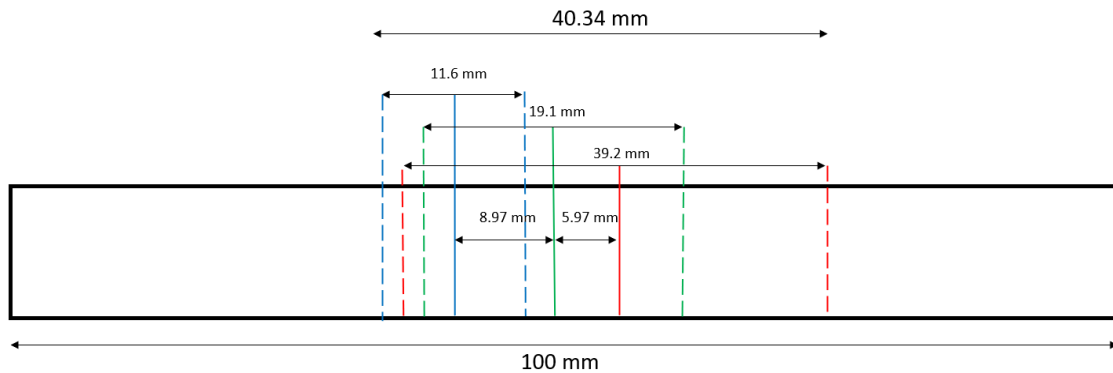


Figure 32: Full camera travel range

The whole optical system will be enclosed in a metal casing. The inside of this casing will be coated with a black, high temperature paint to absorb scattered light and reduce internal scattering. The base of this device will be a half-inch thick metal sheet where the optics will be mounted on. Aluminum is the choice of metal due to it being lightweight and able to dissipate heat relatively quickly. There will be a divider between the electronics and the optics. This is so the scattered laser light does not damage the electronics in any way.

5.2 - Hardware Design

The overall hardware system includes a Raspberry Pi 3B+ communicating with an ATMEGA328P (microcontroller). The ATMEGA328P will then communicate to the TMC2208 and then the TMC2208 will communicate with the NEMA-17. The NEMA-17 will rotate to control the movement of the camera on top of the translation rail. The Raspberry Pi and ATMEGA328P communicate via digital pins and serial communication to send signals to control the TMC2208. The TMC2208 then controls the motor to control the movement of the rail. The Raspberry Pi is powered by the 5V output of the battery, the ATMEGA328P is powered by the output of the regulator, and the TMC2208 is powered by the output of the regulator as well as the output of the battery. **Figure 33** outlines how the signals move between each component and will be further discussed in detail in the following sections.

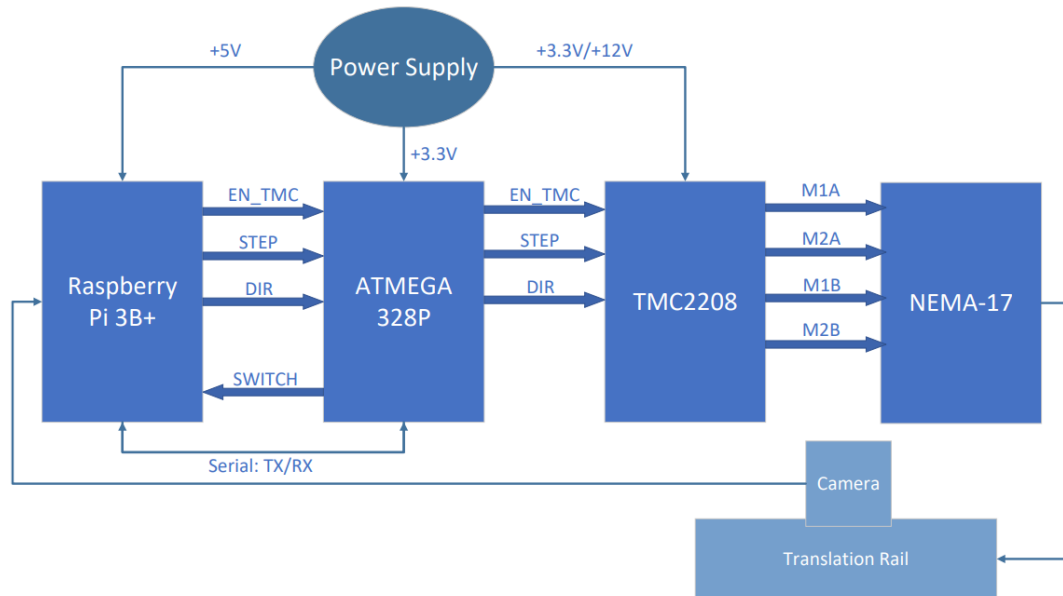


Figure 33: Hardware block diagram. More detail is added here from the hardware section of the main block diagram.

5.2.1 - Printed Circuit Board

For our input terminals we needed a voltage connector that would allow us to access the male DC terminal. For this reason we will use a DC power barrel jack for the input to the PCB. We will choose to use jumper wires for testing purposes initially. This way we can test the device before making any changes that could likely harm the PCB. This way we can avoid unnecessary mistakes that could cost us time along the design process. We can solder permanent wires later on to give the design a more professional appearance and reduce accidental pulling of the wires that feed the microcontroller.

After this the power goes to the regulator and the regulator outputs 3.3V. The regulator has bypass capacitors on the inputs and outputs. There is a voltage divider consisting of 73.1kohms and 22.1kohms to output the 3.3V. The output is calculated by taking the ratio of $R_{fbt}/R_{fbb} + 1$ and multiplying it by 0.765. The motor driver has STEP and DIR inputs that come from the ATMEGA328P to control the motor. It also has 4 pin header outputs to connect to the two coils of the motor. We also pulled out DIAG and INDEX to be connected to the ATMEGA328P to be read in if needed to diagnose the driver. To enable the driver it needs to be pulled low so this is also connected to a digital pin of the microcontroller to be controlled to turn the driver on and off. The driver needs 3.3V to operate while the VS pin for the motor needs 12V. There are bypass capacitors added to the input of the 12V for the VS pin. The CPO and CPI pins are connected together with a 22nF in between. These pins are used for the charge pump capacitors. The charge pump voltage is connected with a capacitor to the VS pin as well. The 5VOUT pin is connected to GND and is not used in this application. MS1 and MS2 can be used to have greater microsteps

depending on how they are connected. To achieve 8 microsteps which is what we are interested in we need to connect both to GND. The rest of the pins are connected to GND because they are not used or need to be grounded. Below in **Figure 34** shows the schematic for the power jack, regulator, motor driver, and header of the motor.

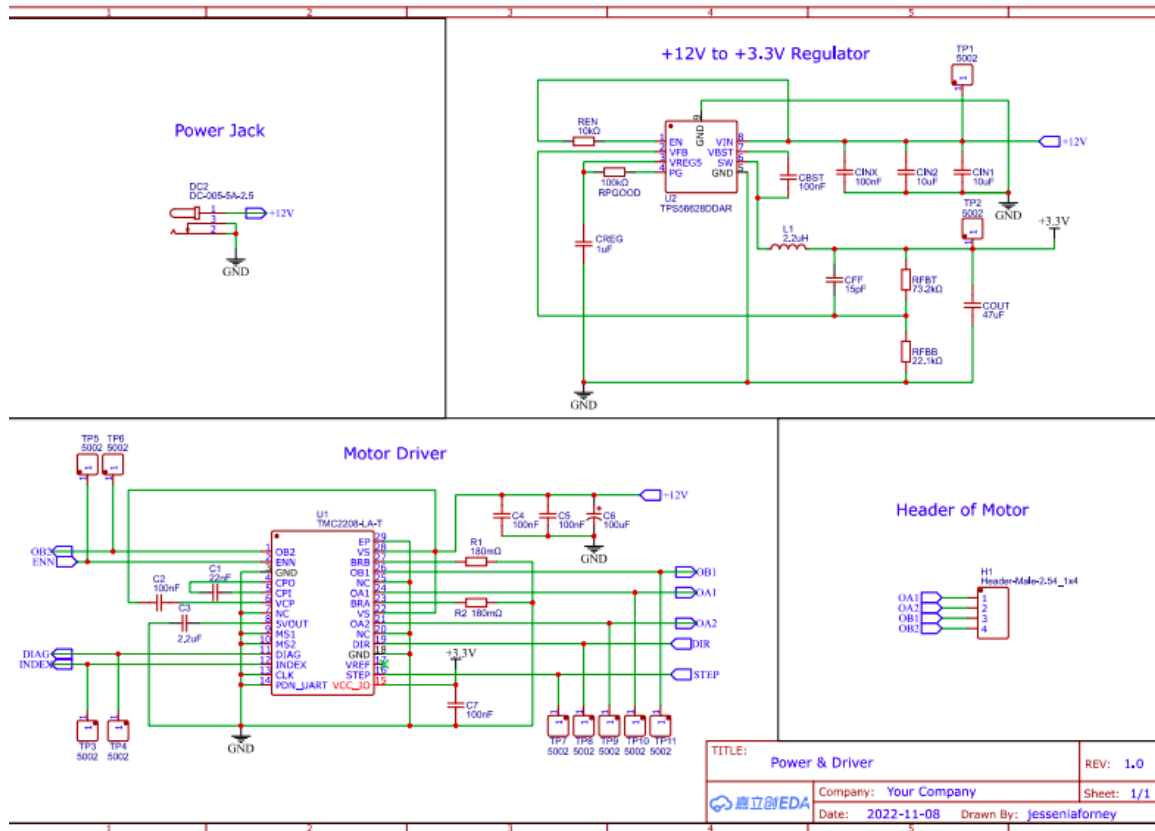


Figure 34: Power and Driver Schematic

The second page of the schematic seen in **Figure 35** shows the ATMEGA328P with all of the signal lines to and from the Raspberry Pi and the TMC2208. The signals coming from the Raspberry Pi include STEP, DIR, and EN. These will be read in as inputs to the ATMEGA328P and then written as outputs to the TMC2208. Enable when pulled low will enable the TMC2208. STEP and DIR will be used to control the motor. There is also a microswitch attached that when switched on it will send a signal from the ATMEGA328P to the Raspberry Pi to reset the position of the camera to the zeroth position. The microcontroller also uses a 16MHz external crystal with two 22pF capacitors connecting it to ground. While the ATMEGA328P does have an internal 8MHz crystal we wanted to use the 16MHz crystal for our purposes. A push button is also attached to the reset pin of the ATMEGA to be used to reset it. We also pulled out a few more digital pins to pin headers to be used for different purposes. TX and RX pins are used as a serial connection and can be connected to the Raspberry Pi as well as the Arduino Uno for programming. The RPI_EN pin is used as an extra pin if needed. SCK, MISO, and MOSI are pulled out to be used to burn the bootloader to

program the microcontroller. DIAG and INDEX are also used if needed to read the diagnostics from the TMC2208.

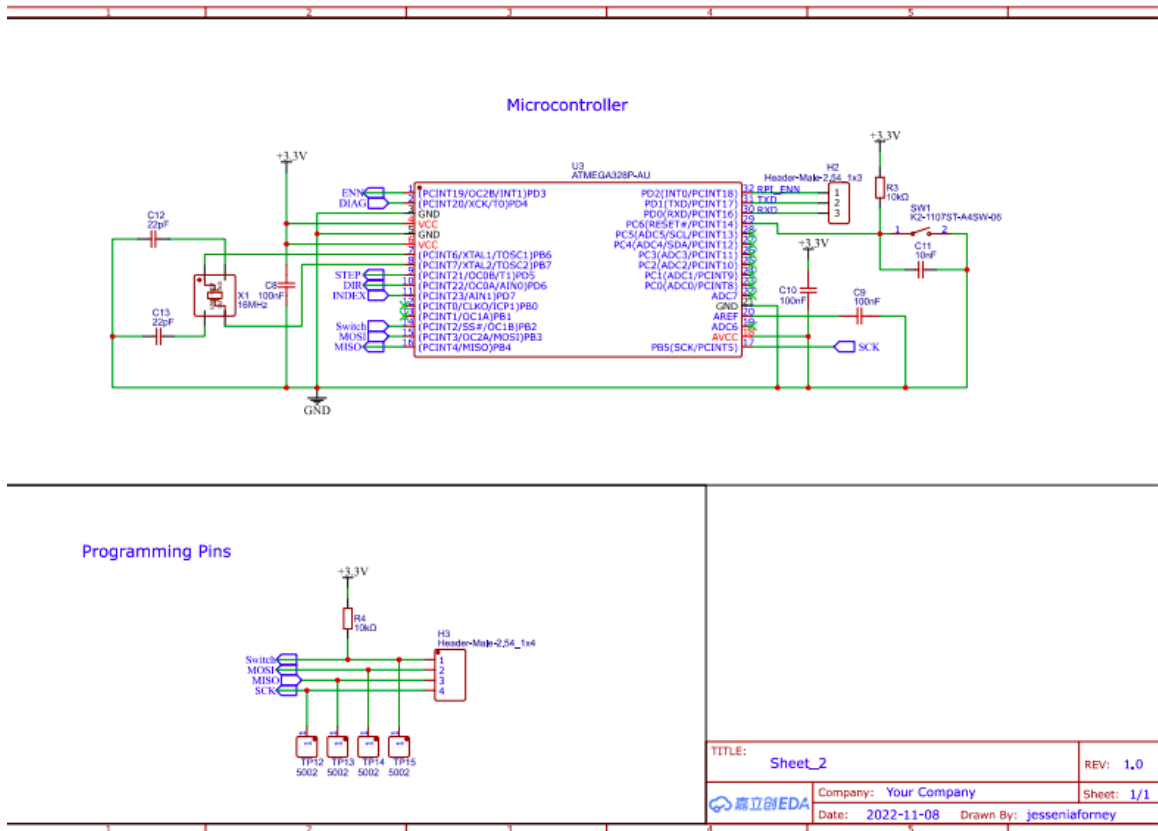


Figure 35: Microcontroller Schematic

To create the board layout, we export the schematic design and drag the parts to the appropriate areas. Connecting the components together and laying the traces. Paying special attention to the width of the traces, we see that the default is 0.25mm which will only allow for .88A of current to flow. We change this to .3mm just to be on the safe side. **Figure 36** shows an image of the 3D board layout of the final circuit board that we will receive from JLCPCB, once we have submitted the Gerber files.

After making sure all the traces are routed and we have shrunk the total size of the board to contain the components we have selected we now have our circuit board. After we have finished with the bottom layer of the board, we select the top layer of the board and lay down a copper grounding layer. This will serve as the ground for all open pins. We can now export the Gerber files, pick-and-place file, and BOM with LCSC parts numbers. This will be sent out to be manufactured with an estimated return time of about 2 weeks.

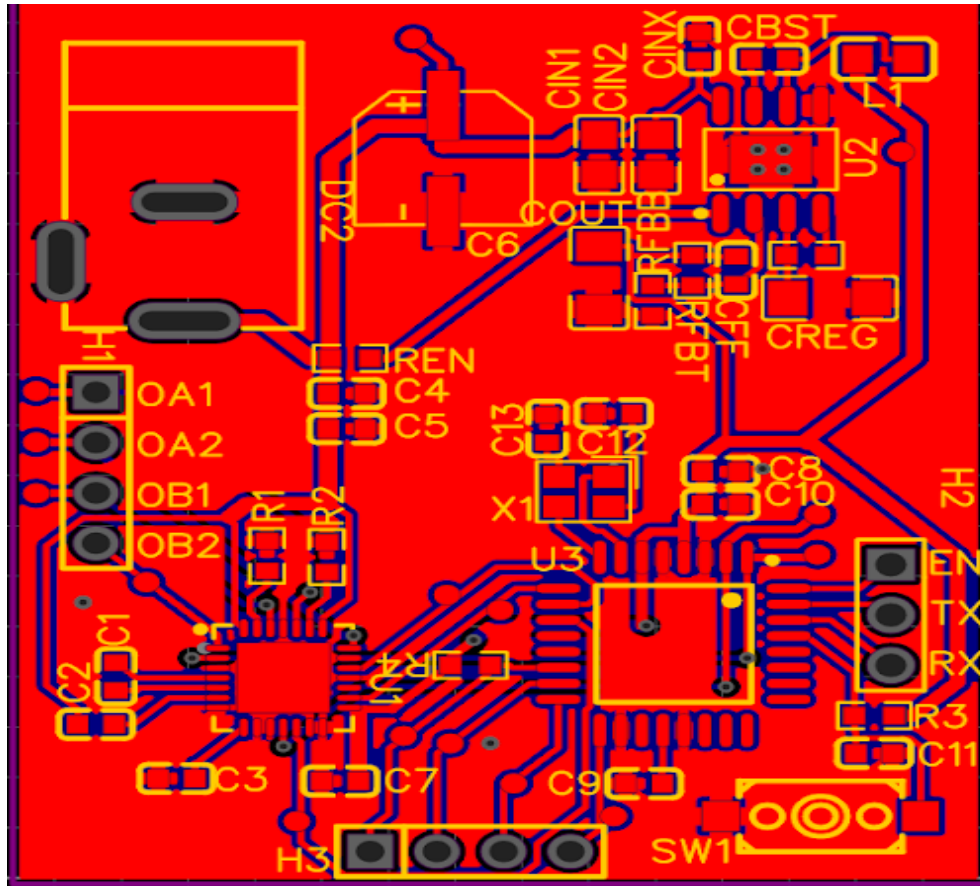


Figure 36: JLCPCB Circuit Board Layout Final Design

We would like to rely entirely on assembly through JLCPCB. For testing purposes we will also purchase a through hole package of the ATMEGA328P and the TMC2208, but have surface mount packages for our printed circuit board. We have checked that all of the parts on our PCB are in stock with JLCPCB which will allow JLCPCB to place the parts on our board for us. We will be ordering 5 PCB's to minimize the risk of ruining the board during testing. However, due to the prices of ordering all of the parts we decided to only have two boards with parts already placed and the other three will be available to solder ourselves.

If we need to solder any components onto the PCB we will need to observe that the contact's are properly placed. This will be accomplished using a multimeter set to measure resistance and running a continuity test. When we run the continuity test we will need to ensure that we have a good connection on each of the pins. This will verify that the soldered parts are fused correctly and make sure none of the pins are accidentally making contact with each other. After connecting the power supply to the PCB, we are ready to test between the PCB and NEMA 17 motor. We will look for a voltage output of 3.3V and 12V and that the current is sufficient to run our motor at rated current. This test will not only verify that we have enough current to run our motor, but also we are not needlessly exceeding the motors current rating. Which could lead to overheating

and damage to the motor. This can be done using the oscilloscope within the senior design lab. Finally the PCB will be on top of the Raspberry Pi module as shown in **Figure 37**.

After attaching the Raspberry Pi to the PCB, again using the oscilloscope provided in the lab, we will examine the mA output over time. We are expecting the TMC2208 to produce a near perfect sinusoidal current response as we step through the 8 subdivisions offered by the stepper driver. If for whatever reason the current is not what we expect, we must begin testing the power supply and work our way back towards the motor. These results should be verified against the waveform chart in the TMC2208's datasheet. This will require running multiple test programs through the Raspberry Pi.

Once we are satisfied that the waveform current does not exceed that rating of our NEMA-17 stepper motor, and we are receiving enough current to operate our system we must perform a heat test. We will monitor the heat generated by the PCB by pointing a laser thermometer at the TMC2208. Over the course of our first full functionality test of the device we will be looking for a temperature increase that should not exceed 85 degrees celsius. Which is the maximum rated heat that the raspberry pi can withstand before shutdown. Ideally, the temperature of the PCB should not increase more than 5 degree's C above the ambient temperature of the room. Below is a sketch drawn to scale representing the power system once it is assembled. The fan that will sit at top of the PCB has been left out deliberately to show detail.

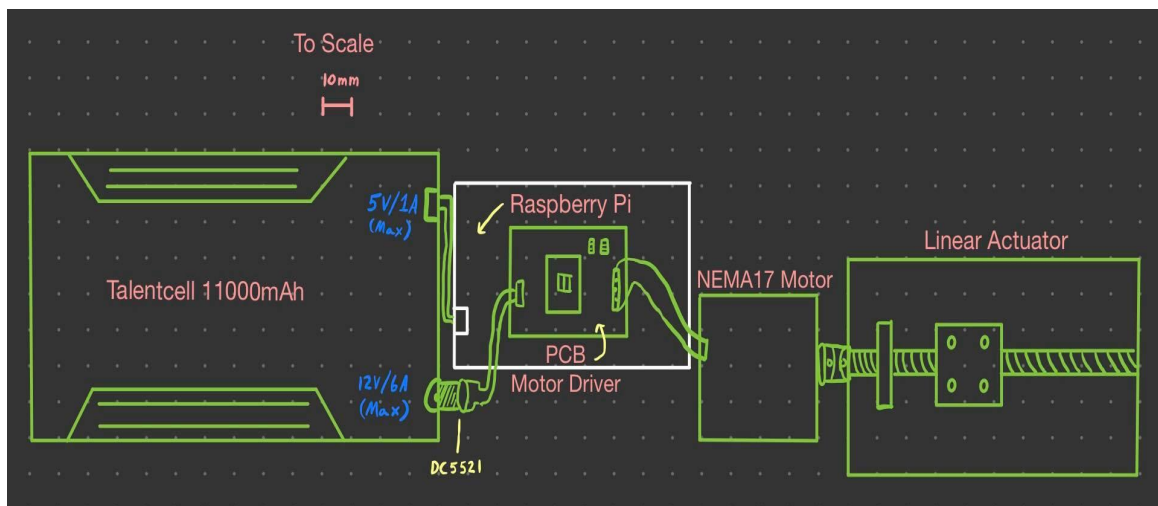


Figure 37: To Scale Representation of Assembled Power System

5.2.2 - Chassis Design

The chassis of the device will come in two parts, the base plate and the cover. The material of choice for both the base plate and the cover is aluminum, as this material is cost effective, durable, and easily obtainable through big-box hardware stores such as Home Depot and Lowes, or online retailers such as

Amazon. aluminum's low density also makes it ideal for portable devices such as ours. The device overall will be 70cm x 70 cm x 20cm. The base plate will be 1/2 inch thick to ensure that it will be rigid enough to withstand being moved around without significant flex, as flexing in the base would affect our optical alignment and potentially damage the fragile wedges and lenses. The base will have holes drilled into and through it to hold the linear rail and optical post holders. The linear rail will be mounted using M4 threading, and the optical post holders utilize M6 threading. These drill locations are dictated mostly by the optical design, and will be realized once the ordered parts come in. The cover will be cut from sheets of 1/8 inch aluminum sheets, and will be secured together using M4 screws. The cover will have dividers for the optical portion and the electrical portion, as the electrical portion will generate heat that will need to be removed by a fan, and the optical section can not be exposed to external optical noise. To facilitate the removal of heat, the cover above the electrical section will have a grill cut for ventilation, along with 50mm x 50mm mounting holes for a fan. The cover will have circular holes drilled through it on opposing sides to allow entry and exit of the beam under test. More holes will be needed to cut for the external 12V DC barrel jack, ethernet port, and power switch. In order to assist with dissipating the heat generated by stray reflections from the wedges, the insides of the cover and the top of the base plate will be painted using a matte-finish high temperature grill paint, which can be acquired from a hardware store for low cost.

5.2.3 - Camera Mount

To set the camera sensor on the beam propagation axis, a mount was designed in Blender to secure to the top of the stage of the linear stage actuator. To secure the camera mount to the stage of the ball screw, the camera mount is fastened using four M4 socket-head bolts in a 30x30mm square arrangement. The center camera sensor is currently designed to sit 85mm above the base of the device (bottom of the ball screw). The cable will run over the top of the mount to the Raspberry Pi to the side of the stage. If adjustments need to be made, the 3D CAD model's vertices can be adjusted at any time in Blender. This part was 3D printed using PETG filament in the CREOL Senior Design Lab. **Figure 38** below shows the model in Blender.

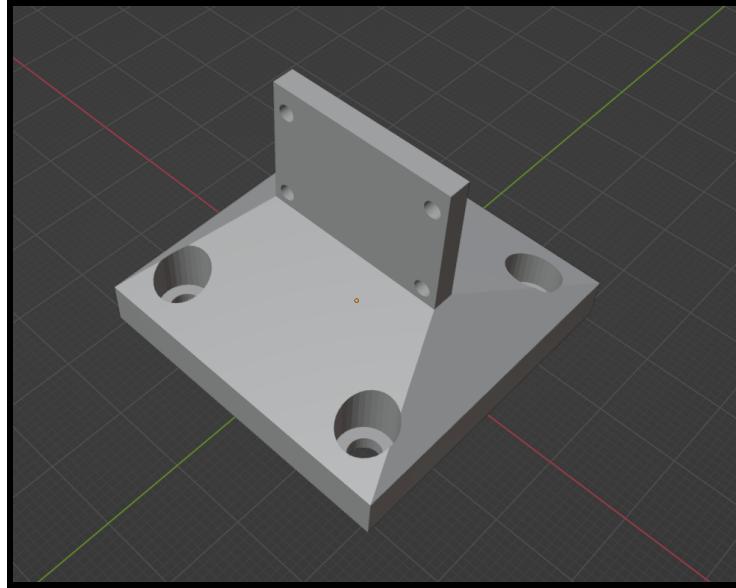


Figure 38: Blender Model for Camera Mount

5.3 - Software Design

The software for this project is broken up into two programs, the Server Program, and the Client Program. The client is a Python program developed for a PC running a Linux-based operating system that supports a Python 3.10.8 ecosystem, and is responsible for the majority of the image processing for measuring beam waists, performing curve-fitting to extract beam parameters as well as providing a graphical user interface to allow the user to interact with the scanner hardware through a network socket. The GUI of the application includes fields to adjust measurement parameters such as beam wavelength, images per measurement, measurement type, etc,. Below these measurement options are fields for moving the camera to a specific location on the ball screw, or scanning a range of regions, as well as capturing images. a button to call or halt a scan, a window to display beam measurement data, and a plot for visualization of the beam-waist measurements at each location along the beam propagation path. The Raspberry Pi 3 B+ (Henceforth referred to as “RPI”), running a server program written in Python, is responsible for receiving and parsing the commands from the client in order to drive the stepper motor and interface with the camera sensor. The client will send requests to the server in the form of TCP packets with pre-defined datagrams.

Beneath the GUI, the client program takes on the task of calculating the beam diameter in the form of an $1/e^2$ measurement, as well as calculating the “second-moment beam waists” and M^2 quality values as stated in *ISO 11146-1*^[35]. The client will request a capture or movement from the server, and the server will begin by driving the motor to the locations defined by the client. The server can then capture images at those locations, and transmit the images to the client for processing. After 10 measurements of the beam at 10 different locations,

curve-fitting is performed on the waists to extract the beam parameters of M^2 , Z_0 , and W_0 . After the beam minimum waist is known, its Rayleigh length is able to be calculated. After 5 measurements are conducted within one Rayleigh length of the minimum waist on either side of the minimum waist location, and 5 outside 2 Rayleigh lengths, curve-fitting is attempted to extract the beam parameters. This is required for a proper M^2 value measurement.

Library Name	Description
PyBluez	PyBluez provides an interface for interacting with the RPi over a bluetooth connection from the client for configuration purposes
PiCamera	PiCamera provides an API for interfacing with the camera module from Python
Numpy	Numpy provides several logical data structures used by PiCamera to store sensor data, namely <code>numpy.array</code> . Numpy will also provide the serialization functionality to package arrays to be sent over LAN. Numpy's other mathematical functions will be used to modify arrays when needed
SciPy	SciPy is a library designed for solving optimization problems, as well as algebraic and differential equations. For this use case, the <i>curve_fit()</i> function will be used to find the M^2 values provided the second-moment beam waist measurements
RPi.GPIO	RPi.GPIO will be used to enable control of the GPIO pins using Python.
Tkinter	Tkinter will be used to generate the graphical user interface
matplotlib	Matplotlib will be used to generate plots based on computed sensor data for beam radius and displaying intensity maps of the beam cross-section.
socket	Standard Python library for utilizing network sockets

Table 25: Python Libraries and Descriptions

The client-server approach, along with the large quantity of raw image data and the corresponding computations required to calculate beam waist and quality demanded a high-level language. Although the implementation could be developed in a low-level language such as C, or a high-level language like C++, the time constraints of the project narrowed the decision to use Python as the

primary language of development, due to its extensive catalog of libraries for matrix/multi-dimensional array computations and user interfaces. The LAN network was picked to be the medium of communication from the server to the client, due to the hundreds of megabytes worth of image data that would have to be moved from the server to the client. In order to configure the server to connect to the LAN, either a direct connection using a CAT5e cable to the RPi can be used, or the device can be configured to connect over bluetooth. However, due to security concerns, it is advised to forego configuring over bluetooth and instead connect the internal RPi to an external monitor and keyboard to configure the wireless interface through the Desktop. **Table 25** lists the libraries used and a brief description of their purpose in the design.

5.3.1 - Preparing the Raspberry Pi

In order for the RPi to properly interface with the camera, the legacy camera stack must be enabled. This is due to the fact that in the “bullseye” and later versions of the Raspbian operating system, they have changed the camera stack to libcamera. This is an issue however, as the *PiCamera* library is built around the old camera stack. However, this can be fixed by enabling the legacy camera API in the raspi-config program. The RPi should also be updated using the apt update and upgrade commands to ensure that all firmware is running on the latest version. All of the program files for this application will be stored in a folder in the directory “/Desktop/snr_dsn/”. In order to start the server program on boot, a systemctl service file is created to start the server program on system boot.

5.3.2 - Connecting the Laser Profiler to the Network

Although earlier iterations of this design use the bluetooth server to accept configuration information from a client program over a simulated serial port, that functionality was removed from the latest version due to security concerns. Instead, the current solution is to connect the internal RPi to an external monitor and keyboard and configure the wireless network through the Desktop

5.3.3 - Server Program

The Server Program will be hosted on the RPi, and will communicate over a TCP socket with the Client Program using the *socket* library. The Server Program will start only after a network connection is established, and a valid IP address is issued for the RPi. The server and client communicate using TCP sockets in order to ensure that data transmitted is received on the other end, as lost packets would break the program and/or image data.

Each command is constructed out of a predetermined datagram, and a conversation-based approach is used to ensure that image data is transmitted and received when expected by both sides. Each command is headed by 4 bytes used to send the message length in ‘little-endian’ form. The first command that is typically sent from the client is ‘config’, which sets the parameters of

exposure-type (auto exposure or manual set), beam wavelength, and images per capture. The 'background' command instructs the server to capture a full resolution background image to subtract from all future images to reduce the effect of noise on waist measurements. Although these commands could be conveyed only using a byte, the full english command is passed as ascii to improve readability. For example, the first 'word' of the message for the 'config' command would be "CONFIG ". Each word of the message is separated by a space character to simplify communication and readability. Following the first word of a command is the command parameters, which vary from command to command. The full table of commands and their parameters are shown in **Figure 39**.

Command Word	Parameters
CONFIG	exposure_time, wavelength, images_per_capture
BACKGROUND	-
MOVE	index
CAPTURE	-
SCAN	number_locations, location(s)...

Figure 39: Commands and Parameters

The server is primarily responsible for the capture and transmission of image data. The image data is stored in the server memory as a numpy.array class, which needs to be serially transmitted by the socket and reconstructed by the client. In order to accomplish this, the numpy array is first converted into a Python string object using the numpy.array class function *tostring()*. This Python string is then encoded into a utf-8 bytes array using the string class function *encode()*. This bytes array is then transmitted over the socket to be reconstructed on the other end by the client. Before the server sends any image, the server transmits "IMAGE" followed by the exposure time used to capture the image. The image data immediately follows this message. After the image is successfully read and stored by the client, the client messages "DONE" and waits for a "STANDBY" from the server.

5.3.3.1 - Server Side: Motor Driving

In order to drive the stepper motor, the RPi will have GPIO pins connected to the input pins of a stepper motor driver IC. The pins used to drive the stepper motor will be *STEP*, *DIR*, and *ENABLE*. When the *ENABLE* pin is pulled LOW, the stepper motor driver IC enables the motor-side outputs. *DIR* determines the direction of rotation, where LOW drives the stage towards the focal lens, and HIGH drives it away, and *STEP* will step the motor one step in the selected direction when presented with a rising edge (logical LOW followed by logical HIGH). The control loop for driving the stepper motor is trivial, as each step is known to move the stage at a constant distance, and the number of steps

required to travel to a point is the distance of that point minus the current location of the stage, divided by the constant `MM_PER_STEP` of 0.003125

The GPIO pin assignments are as follows: DIR is connected to GPIO3, STEP connected to GPIO4, ENABLE connected to GPIO 17, and ZERO connected to GPIO2. In order to allow the RPi to know the “zero” location of the motor, a micro-switch connected to the RPi is placed in the travel path of the translation stage near the end of the ball screw nearest to the focal lens. When the switch is closed, the software will know that the translation stage is zeroed. This zeroing step will occur every time that the server powers on, as well as after a successful connection from the client. The positions of where the stepper motor stops to take an image based off of this zero point are used for calculating the M^2 values.

5.3.3.2 - Server-Side: Auto-Exposure

For the image captures to be useful, a suitable exposure time must be found such that no pixel of the sensor is driven into saturation. Before the server program captures any image, a suitable exposure time will be found and used using the auto exposure subroutine if a manual exposure time was not declared by the client. Since the sensor data is 10-bits, the maximum intensity value of a given pixel is 1024, so a reasonable threshold is somewhere near this upper bound. It will be hard programmed into the server as 850 to 950, but this is able to be adjusted at any time in order to account for edge cases such as very dim lasers. In the auto exposure function, there is a loop that captures an image in the form of a regular 8-bit RGB array, and finds the maximum intensity value of that image. This is not an accurate representation of the beam-under test, but it is much quicker than attempting to capture the Bayer data for each image and using it to determine saturation. If the maximum value is below the minimum threshold, the server increases the camera exposure time and repeats the process. Else if the server determines that the maximum value is above the maximum threshold, the server decreases the exposure time and loops. If neither condition is met, then the captured image’s maximum pixel value is between the minimum and maximum threshold, and a suitable exposure time has been found. Although using RGB arrays does not accurately reflect the values of the Bayer array data, as long as the values are large enough (on the order of 500 or greater), after background subtraction, the client program will still process it accurately.

5.3.3.3 - Server-Side: Capturing Images

The *PiCamera* Python library is used to interface with the Pi NoIR Camera V2, allowing for easy retrieval of image data from the sensor. For this application, the raw Bayer pixel data is retrieved from the sensor. An alternative to the PiCamera library is OpenCV, which would have been preferred if not for the extensive beginner documentation and examples for PiCamera, which made it easy to become accustomed to that ecosystem. If the device had to be redone however, OpenCV would be preferred as it provides a larger suite of tools and most importantly, is still updated. Bayer data is captured to allow more flexibility in the

design, as well as avoid some post processing effects such as auto-white balance. These camera parameters are disabled when the camera object is declared in code. Before any beam-image captures are done, a background image must be taken while the beam blocker is lifted on the chassis in order to subtract the background noise from the following images of the beam cross section. After the stepper motor drives the sensor to a capture location, the sensor data is grabbed by the RPi, and the stepper motor immediately moves to the next location. A class diagram for the server program is shown in **Figure 40**. If requested by the user, the server will take multiple images per stop depending on the parameter passed by the user in the CONFIG command. Taking multiple images at a location and averaging their intensity values can help improve the SNR and increase the accuracy of the instrument. This averaging process is described in further detail in the next section.

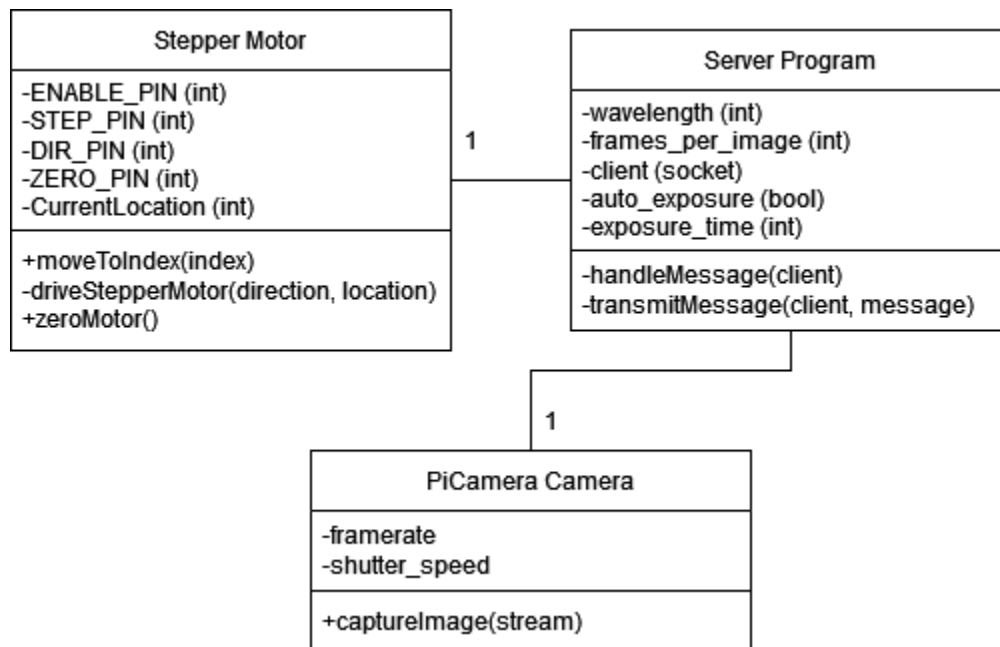


Figure 40: Server Program Class Diagram

5.3.3.4 - Server-Side: Image Processing

Image processing is purposefully minimized on the server, as the RPi processor is most likely out-matched in raw computational power by the machine hosting the client. And thus, the image data needs to be sent to the client for processing. However, due to the large amount of image data that is required to be moved, there may be a noticeable delay in when the images are ready for transmission and when they appear on the client. In order to remedy this, the GUI will display the transmission status so the user knows how far along the image transmission is. For each scan location, if multiple images were requested by the user, the server will create a mean image by averaging all the pixel values across each of

these captures. After a mean image is created, or if only one capture was done at each location, the image data will be subtracted by a background image on the client after transmission. The background image is a capture of the background optical and electrical noise that affects the sensor. After an image has been serialized and transmitted to the client, the client will subtract background noise out of it.

Each pixel value is stored as a 16-bit integer, meaning storing a single image takes approximately *16.2 megabytes* of space. A minimum of 10 images need to be captured, so the minimum amount of image data per scan will be approximately *162 megabytes*. The images will be stored in a *piBayerArray* class from *PiCamera*, which utilizes *Numpy* arrays to store the data. The numpy arrays are serially encoded as utf-8 byte array strings using the numpy.array class function *tostring()* followed by the Python string class function *encode()*. When it is time to send the image data, the server will send the client “IMAGE” followed by the exposure time that the image was captured at. Immediately after, the server will message chunks of each image data serially as utf-8 encoded bytes for reassembly in the client. Once the image data is transmitted, it will be reassembled for further processing. Transmission of all the image data may create a noticeable delay by the user, so the text log of the graphical user interface will update as new packets arrive and are processed. The algorithm is made trivial due to the high-level nature of Python, as the transmission is handled reliably with the TCP socket, and all that is required is a ***sendall()*** function call from the server. **Figure 41** illustrates the image capture process per location.

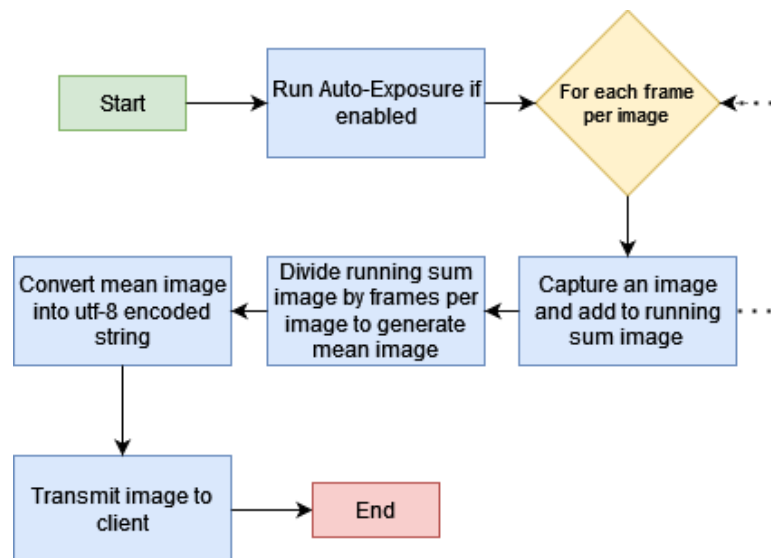


Figure 41: Image Processing Loop done by the Server0

5.3.4 Client Program

Between the RPi and the PC, the PC is much more likely to have raw computational power at its disposal. Desktop processors from several generations ago will run circles around the modest 1.4 GHz quad-core processor hosted on the RPi. Because of this, it is desirable to do the bulk of image processing on the PC rather than the RPi. The client program will provide the user with a graphical interface to conduct laser beam captures, as well as view scan data as it is being computed. In the background, the client program will take on the task of calculating the beam waists in both principle axes, as well as the curve-fitting to extract beam parameters. Before the client is able to connect to the server, the server must be on the same LAN as the computer running the client. If wireless connectivity is preferred, the client must configure the server over bluetooth or through an external monitor connected to the RPi to connect to the wireless LAN. This can be done by an auxiliary program that will run if the server and client are being run for the first time.

5.3.4.1 - Client-Side: Incoming Image Processing

The image data incoming from the server is the serialized version of the *Numpy* arrays of the image data. To begin, this data needs to be deserialized, restoring them to *numpy array* objects. To accomplish this, the utf-8 byte arrays are converted using the *numpy.array* class function *frombuffer()* to generate an array out of the bytes. Once restored, these objects will undergo an interpolation step provided by the *colour_demosaic* library function for bilinear demosaicing, which converts the Bayer sensor data into a 3-D RGB array. The color array selected is the one that most closely matches the wavelength of the beam under test. After each image is demosaiced, if a background image was captured, it will be subtracted from the incoming image

5.3.4.2 - Client-Side: Beam-Parameter Computations, ISO

The client program is responsible for handling the computations for calculating a beam's first order moments and second order moments of a power density distribution in accordance with **ISO 11146-1**_[35] for both the x and y axes, along with using the aforementioned values to calculate the beam widths ($d_o(z)$). The first step to finding the M^2 value is to find the location of the centroids of the beam. The equations for the centroids of the beam (Equations 1, 2) are defined as:

$$\bar{x}(z) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x,y,z)x \, dx \, dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x,y,z)dx \, dy} \quad \bar{y}(z) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x,y,z)y \, dx \, dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x,y,z)dx \, dy}$$

Where $E(x, y, z)$ is the energy distribution cross section at location z along the optical axis. The values $\bar{x}(z)$ and $\bar{y}(z)$ represent the x and y coordinate of the maximum value of the intensity distribution. Accompanying the centroids, the beam second order moments (Equations 3, 4, 5) are defined as:

$$\sigma_x^2(z) = \langle x^2 \rangle = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) (x - \bar{x})^2 dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) dx dy} \quad \sigma_y^2(z) = \langle y^2 \rangle = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) (y - \bar{y})^2 dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) dx dy}$$

$$\sigma_{xy}^2(z) = \langle xy \rangle = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) (x - \bar{x})(y - \bar{y}) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) dx dy}$$

Since we are doing these computations on a computer whose data is all discrete, we may discretize these equations by changing the bounds of integration to a region of integration (ROI), and replacing the integrals with summations. According to the standard, the ROI must be centered to the beam centroid, and sized three times the beam widths. Since these values are originally unknown, the procedure starts with an initial guess of the integration area. The new values generated from this initial guess will be used to recalculate the integration area, and shall be repeated until convergence of the result is obtained. For this device, the height of the camera sensor will be used as the initial guess of the integration area (2480 x 2480 pixel area), and convergence will be defined as less than 1 percent change between results. The discrete time form of equations 1 and 5 are shown below as an example:

$$\bar{x}(z) = \frac{\sum_{y_0}^{y_f} \sum_{x_0}^{x_f} I(x, y, z) \cdot x}{\sum_{y_0}^{y_f} \sum_{x_0}^{x_f} I(x, y, z)}$$

$$\sigma_{xy}^2(z) = \frac{\sum_{y_0}^{y_f} \sum_{x_0}^{x_f} I(x, y, z) \cdot (x - \bar{x}) \cdot (y - \bar{y})}{\sum_{y_0}^{y_f} \sum_{x_0}^{x_f} I(x, y, z)}$$

Where the summations are carried out over the X and Y axes of an array of image data. y_0 , y_f , x_0 , x_f denote the bounds of the ROI in the image data, x and y describe the current pixel in question, $\bar{x}$ and $\bar{y}$ are the location of the beam centroid, and $I(x, y, z)$ is the intensity value of that given pixel of image data. The bounds of summation will be shrunk iteratively until the extracted values for both the centroid and the second order moment vary by less than 1 percent to reduce the effect of noise in the system. **Figure 42** Below details the algorithm for finding the x centroid. Note that all the equations follow a nested for-loop approach, and

the only difference between the equations is the Numerator. For example, the xy second-order moment numerator will be:

$$\text{Numerator} += \text{image}[x][y] \cdot (x - \bar{x}) \cdot (y - \bar{y})$$

Each of these for-loops will be implemented separately and sequentially, as the centroids are needed to find the second-order moments.

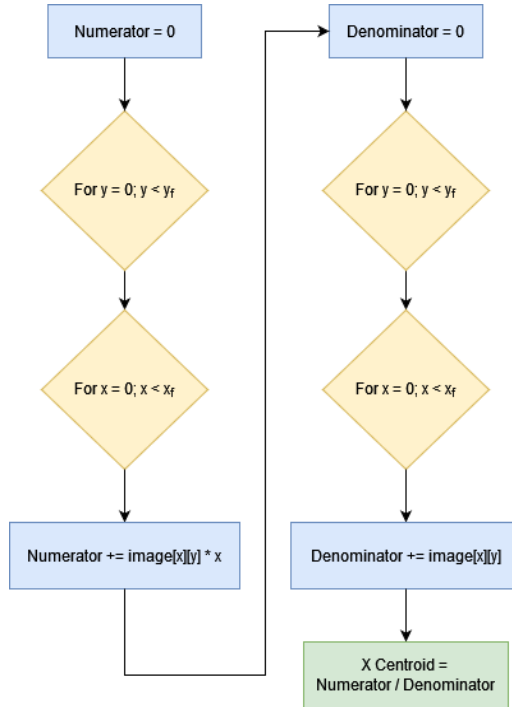


Figure 42: Algorithm for Finding X Centroid

After the centroids and the second-order moments have been calculated, the process will repeat, adjusting the ROI as follows:

$$\begin{aligned} x_0 &= \bar{x} - \frac{3}{2}\sigma_x^2 & x_f &= \bar{x} + \frac{3}{2}\sigma_x^2 \\ y_0 &= \bar{y} - \frac{3}{2}\sigma_y^2 & y_f &= \bar{y} + \frac{3}{2}\sigma_y^2 \end{aligned}$$

This scaling will guarantee that the ROI is always sized three times the beam widths in the x and y directions, complying with the standard. The deciding variable for the convergence loop will be σ_{xy}^2 . After this value converges, the loop will break, and move on to the next step, which is curve fitting the waist data to extract the beam parameters.

The ISO method, although rigorous and recommended for beam profiling, suffers from severe sensitivity to noise, especially on the edges of the images. A

common problem encountered when taking these measurements was the presence of dust on the edges of the camera sensor being illuminated slightly by stray reflections from the laser, causing noise to be present on the captured image. This is not solvable using the background subtraction, as the dust particles are only illuminated when the laser is hitting the camera sensor. Whenever taking a measurement using the ISO method, be sure to visually inspect the capture frame for the presence of noise and verify the waist measurements generated.

5.3.4.3 - Client-Side: Beam-Parameter Computations, $1/e^2$

In order to calculate the beam waists using the $1/e^2$ method, intensity distributions of the x and y axes are generated by taking the integral across each row for each column and vice versa. A visualization of this is shown in **Figure 3** where the dashed lines represent the approximate location of two standard deviations away from the center of the distribution.

After these intensity distributions are generated, they are curve-fitted to the standard two-dimensional gaussian function to extract the standard deviation. Two times the standard deviation of the intensity distribution is the length of the waist of the beam. The returned value is multiplied by 1.12 to convert the value from pixels to micrometers, then divided by 1000 to return the value in mm for curve fitting to extract beam parameters. This curve fitting also generates some uncertainties for the waist measurement, which are also returned and passed onto the final curve fitting. Although this method is less-susceptible to noise introduced by stray reflections and dust, a background subtraction is still recommended to improve the quality and accuracy of the measurements.

5.3.4.4 - Client-Side: Socket Thread

In order to develop a responsive user interface, all of the server communication and image processing is done on a second thread henceforth referred to as the “socket thread”. The socket thread is responsible for transmitting the commands and command parameters to the server as well as receiving and processing the image data. The socket thread receives the commands and parameters after the GUI thread verifies the entries and passes the command through a queue. Once a command has been detected in the queue, the socket thread wakes up and initiates a conversation with the server. When the socket thread receives image data, it first performs a demosaic on the Bayer data to restore it to a full-resolution RGB array. A background subtraction is then performed on this array to remove the background noise from the image. The beam within the image then has its waists measured using the measurement definition defined by the user in the GUI. After the beam waists have been measured, if 10 measurements have occurred, the socket thread will attempt a curve-fit to extract the beam parameters. The returned beam parameters are displayed for the user in the bottom left of the data window of the GUI.

5.3.4.5 - Client-Side: User Interface

The client program is responsible for providing the user a clean, simple to use interface for taking beam measurements. It is designed to make most of the device's functionality easily accessible to the user. The user interface has been divided into two sections, the home window and the data window. The home window contains the fields and buttons relating to taking captures and moving the stage, as well as displaying the latest capture for the user. The data window displays all of the measurements of the beam across all locations, as well as plotting this data for the user. Beneath the user interface is the socket thread, which updates all of the user interface elements as image data becomes available.

Before the client accepts any input from the user, it must check to see if it can make a connection to the server. The server must be on the same LAN as the client, with its IP known to the operator if the client is being run for the first time. The client program will check to see if it has previously connected to the server before to attempt to load known settings, but in the case that no previous settings were saved, a message will prompt the operator that a connection was not successful. If a connection fails, the operator will be notified through the console that a connection was not able to be established, and asked to try again. Once a connection is successful, the graphical user interface will enter into a standby state, and allow the operator to interact with it.

In order to preserve thread safety, the GUI thread is only responsible for reading the entry fields, and the socket thread is responsible for writing to these fields and updating the canvases when new information arrives. On a button press, the GUI thread reads the information in the entry fields pertaining to that button press, and verifies that the entries are numeric and within the bounds of the device. If the entry fields are valid, the GUI thread packages these parameters into a message constructed from a python string, and passes it to the socket thread below through a queue to initiate communication with the server. As images are received by the socket thread, they will be displayed in a window on the left side of the home screen using matplotlib's Tkinter integration. Below this image, the image location and waist information will be displayed in text entries filled with the measurement information. To the bottom right of the image are buttons to crop the image around the beam, or to toggle a grid overlay on top of the image. On the data window, the table on the right will be updated with the latest image measurement information, and the plot will be updated to include the new point. If enough points are captured, curve fitting will be attempted and the results will be displayed on both the plot and in the entries on the bottom left of the window. **Figure 43** displays the home window of the user interface. The data window of user interface is a sub window containing all the beam waist measurements and the curve-fit parameters, and is shown in **Figure 44**

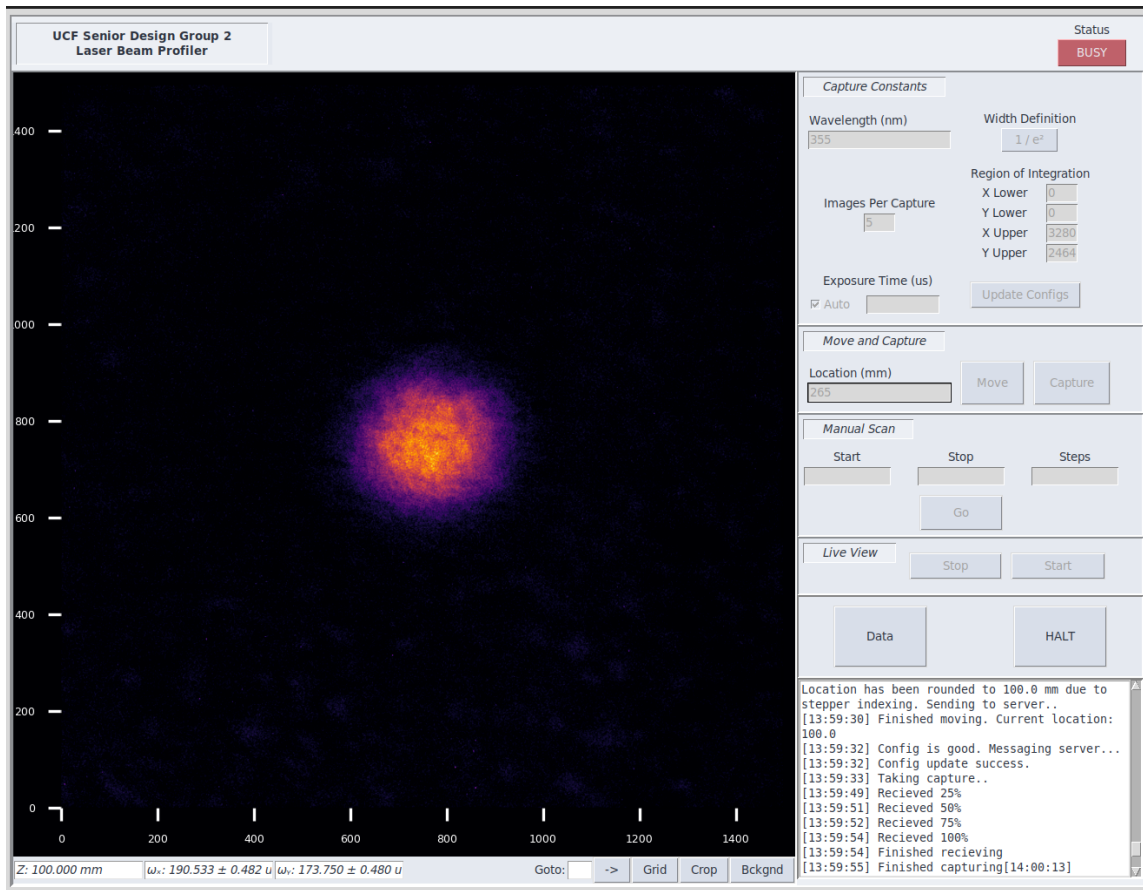


Figure 43: Laser Beam Profiler User Interface Home Window

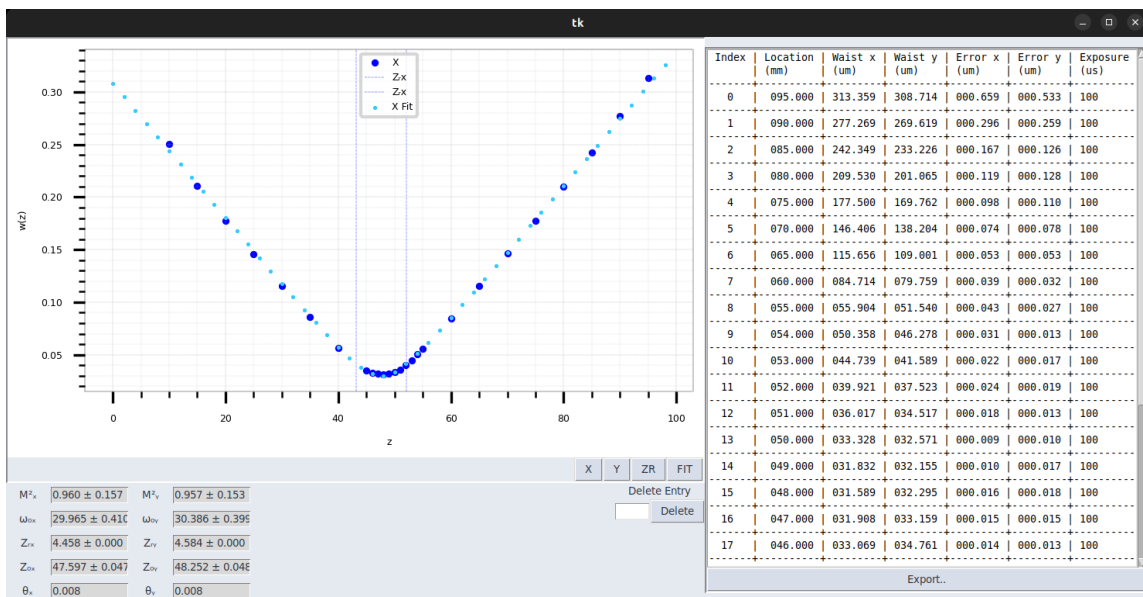


Figure 44: Laser Beam Profiler User Interface Data Window

5.3.5 - Software Design Overview

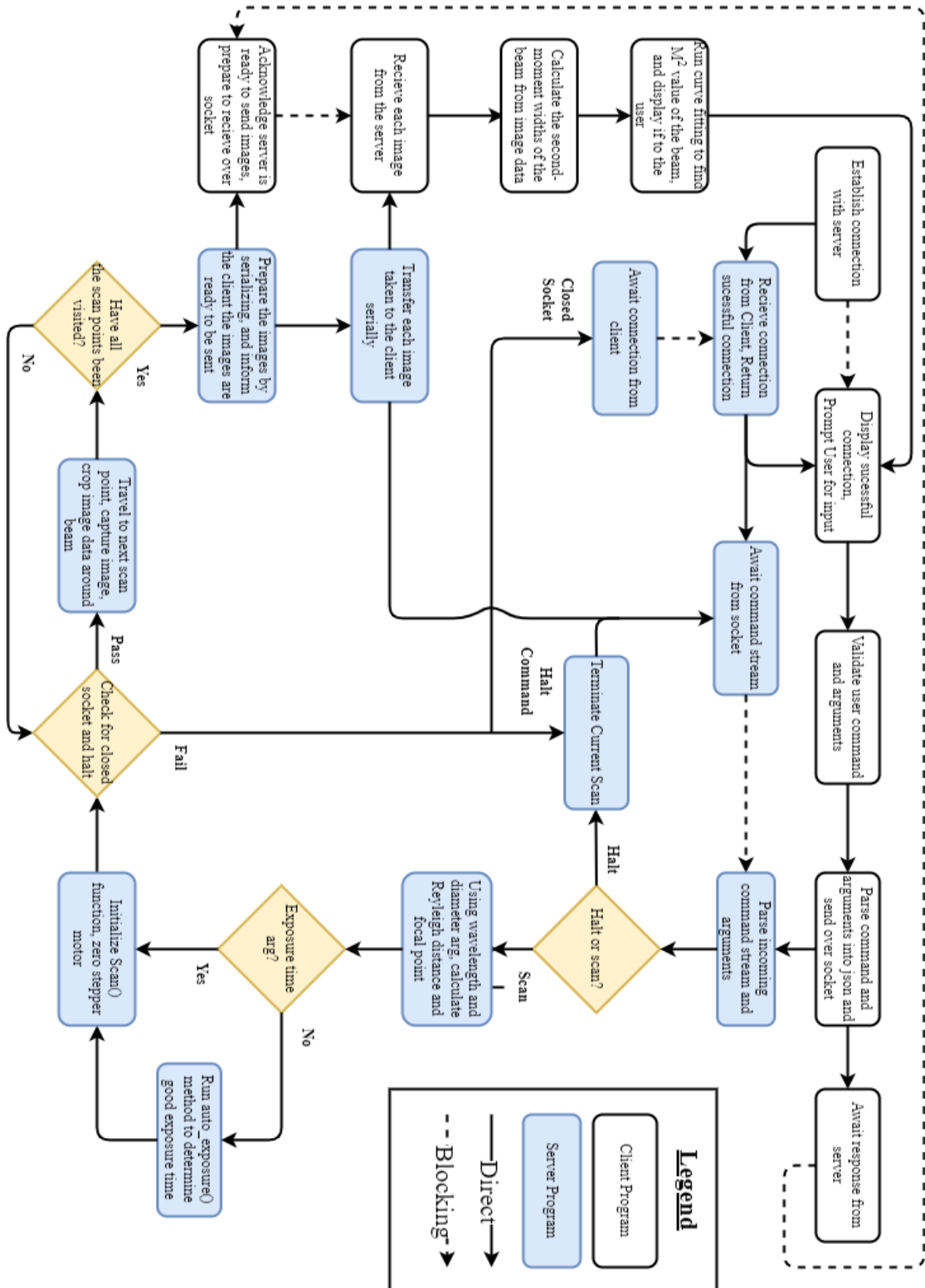


Figure 45: Overall Software Flow Chart

6.0 - Project Testing

This section will discuss all proposed testing that will be done for the optical, hardware, and software components of our design.

6.1 - Optical Testing

Testing of the optics will consist of aligning a laser through the system and accounting for any unwanted reflections, measuring the total attenuation from each device in the system, and imaging the focused beam. With these tests we will be able to compare the measured values with the theoretical calculated values.

6.1.1 - System Alignment Testing

The purpose of these tests is to see if the device is portable and able to be placed in a beam path and have the beam aligned into the system in under 5 minutes. For these tests to be conducted the internal components of the system need to be properly aligned and secured to an optical breadboard provided by our Senior Design lab.

6.1.1.1 - Lens and Camera Alignment

These tests were conducted after the optical wedge alignment had been done on the final device. The third optical wedge was mounted in a kinematic mount allowing for front reflection beam alignment into the focusing lens. The beam needed to be aligned very precisely into the focusing lens to not add any defocusing or increased spherical aberrations, as these will greatly reduce the accuracy of the device. Before placing the lens, the beam was aligned parallel to the device base. The translation rail will only be able to travel parallel to the device base. The translation rail was placed in the beam path and the camera was translated over the full rail distance. This is done to ensure that the camera is aligned along the optical axis. Images along the travel distance will be taken and compared to make sure there are no horizontal deviations. To align the beam along the camera travel axis, the two attenuation wedges were used. The camera was changed from the closest to furthest position on the rail, and the beam was aligned to the center of the camera. This was done many times until the beam was in the center of the camera for the whole travel distance. When these procedures were done the translation rail was secured to the device base.

6.1.2 - Beam Attenuation Testing

Beam attenuation tests are important to make sure that the camera will not be saturated or get damaged when imaging the beam profile. A more thorough approach to this is to measure the attenuation after each surface. This ensures that we know where the power is going through the whole system. For these tests we used the same green, 5mW, 532 nm laser. The average power was measured before attenuation. The power meter used is the 846-R from Newport,

and the detector is the 818-UV/DB model from Newport. This detector has a spectral range from 300 nm to 1100 nm. This power meter has a built in power attenuator to measure up to 1.5 W and 30 W/cm². It was important to use a laser that had a steady output power. All the data was displayed in **Table 26**. The same setup as **Figure 16** was used for these tests.

		532	
		Measured	Calculated
Wedge 1	Average Power IN (mW)	0.968	0.968
	S1 Reflection (mW)	0.0342	0.0411
	Attenuation %	3.533%	4.25%
	S1 Reflection (mW)	0.00255	0.0018
Wedge 2	Attenuation %	7.456%	4.25%

Table 26: Wedge attenuation data table

6.2 - Hardware Testing

The hardware testing section details all the testing that we will do including power supply, motor driver, stepper motor, and continuity testing for our PCB, Raspberry Pi, motor, and power supply. We will be conducting most of our testing at UCF's senior design laboratory located in the Engineering I building, room 456. The equipment provided include the following:

- Keithley 2230-30-1 Triple Channel DC Power Supply
- Rohde and Schwarz HMC 8012 Digital Multimeter
- Rohde and Schwarz RTM3004 Oscilloscope
- Tektronix AFG 3022B Dual Channel Arbitrary/Function Generator

Testing of the hardware should be conducted to ensure proper operation of the stepper motor and the stepper motor driver IC with each other and with the microprocessor. The hardware testing is divided into power testing, motor driver testing, stepper motor testing, and continuity testing.

6.2.1 - Power Testing

It is important to test the power of our devices to see if there is any limitation in the performance or potential damage to the electronics. We will be testing the power using the HMC 8012 digital multimeter that can be found in the senior design lab. We will test the voltage using the Rohde & Schwarz HMC 8012 digital multimeter that can be found in the senior design lab.

6.2.1.1 - Power Supply

Our power supply has three outputs including a max rating of 12V/6A, 5V/2A, and 9V/1A. We can use the multimeter to measure the voltage and current draws coming from these outputs to check that we are getting the power supply we need for our PCB and Raspberry Pi. Our test set up will include the Raspberry Pi, PCB, motor, and power supply. The design section can be referenced to see how these will be connected. We will use the probes connected to the COM and V connectors on the bottom right of the multimeter to measure these values. The COM connector is used for the negative silicon lead which will be placed on the ground pin and the V connector is used for the positive silicon lead which will be placed on the input pin. To measure the voltage, the DC V button is turned on which is located on the front panel. The current is measured in a similar way, with slight changes. The A connector is used for the positive terminal while the COM connector is still used for the negative terminal. The DC I button on the front panel is used instead as well to measure the DC current. It is also possible to measure these at the same time to make sure our results stay consistent by either using two multimeters, two channels, or using the second function on the menu to select a different function to use. We can then compare these values to what we expect to see and see if they are similar. Of course, we expect them to be similar, but this is important to check because we want to make sure we have enough voltage and current supplying our Raspberry Pi and our PCB.

From these measurements we can find the battery life of our power supply while it is at full load. According to the battery's pack's documentation the capacity of the 12V/6A will be 11000mAh, the 9V/1A will be 14500mAh, and the 5V/2A will be 26400mAh. We can calculate the load current or amp draw with the multimeter as explained above. We can find the battery life in hours by dividing the battery capacity by the load current and then compare all of the values to find how long our battery will last when fully charged at full load.

When we received our PCB we did not have a proper way of testing the power on the board. While breadboarded we were able to see that the battery provided the correct power for and the regulator converted to 3.3V to power the microcontroller and motor driver. What we needed was to add 0 ohm resistors to the signal lines. That way we could isolate sections of our board to test without the other sections being affected. We could do this by removing the 0 ohm resistor from the board then placing it back to connect the traces. Since we did not have a way to do this we plugged in the PCB with a 12V power supply and found that the entire board was reading 12V. We do have test points on every

signal line so we were able to see that something was not done correctly on our board. Upon further examination we found that the voltage divider resistors on the regulator had two 84ohm resistors instead of the 73.1kohm and 22.1kohm resistor. This means our regulator was not going to output 3.3V. Due to finding this out too late we did not have time to adjust the resistors for our PCB for the final project review.

For our purposes we had the LM338 linear regulator on hand that we used with our new board design which was a solder board created. We tested it using a 5.7kOhm and a 470ohm voltage divider at the adjust pin that is connected to the output and the ground. This was tested and seen below to output 3.273V DC in **Figure 46**.



Figure 46. Voltage Regulator Testing

6.2.1.2 - PCB

It is important to keep in mind that all the components of our PCB should be measured to make sure there is enough voltage and current being supplied while not overheating our board. We can go through the board from start to finish to see how the power moves through it. It is important to note that when handling the PCB we should wear a ground strap so we are grounded and not interfering with the board as well as the PCB being placed on an ESD mat. We will have already tested the outputs of the power supply to make sure they are at a max of 5V/2A, 12V/6A, and 9V/1A.

The supply voltage to the driver will be coming from the 12V/6A output of the power supply. It will then output 12V/3.3A regulated voltage to the stepper motor via routing through the driver itself. We can check the input connection the same way as described in the power supply testing section to make sure the input is rated at 12V/3.3A max for the driver. When testing pins of our motor driver we can reference the datasheet to see what the max current ratings and electrical characteristics are. The list below in **Table 27** shows the ranges for different pins we will be testing using the multimeter.

Parameter	Min	Max
Supply Voltage	5.5V	36V
I/O Supply Voltage	3V	5.25V
RMS Motor Coil Current		1.4A
Peak Output Current per motor Coil Output		2A
Logic Input Voltage	-0.5V	I/O Supply Voltage + 0.5V
Max Current from Digital Pins		+/-10mA
5VOUT Supply Voltage	-0.5V	5.5V
5VOUT Current		25mA

Table 27: Electrical Characteristics

First we breadboarded our PCB with the microcontroller, voltage regulator, and motor driver. After inputting 12V to the regulator we were able to test the pins of the microcontroller and motor driver to see if they were 3.3V, which they were. The pins going to the motor, however, are 12V. The next step was to test the PCB. As discussed earlier, due to the PCB not being powered properly we placed all of our PCB parts on a solder board. We tested the pins of the microcontroller and the TMC2208 of our solder board to make sure they read 3.3V and the motor pins read 12V.

6.2.1.3 - Raspberry Pi

We will look at supplying the microprocessor or in our case the Raspberry Pi with the 5V/2A output from the power supply. The Raspberry Pi operates with a 5V/2.5A DC power input. We need to measure the voltage and current coming into the microprocessor from the power supply to make sure it is receiving enough voltage and current to support it. This is important to note that our power supply is rated at 2A and not 2.5A. As discussed previously since we are not using add-on shields, the lower current draw might be fine moving forward with the design. Therefore, we need to test that the board is getting enough power to run properly. If it is then we will not need to change our design, however, if it is not then we will need to rethink how we are supplying power and look at using a regulator or a different power supply.

We used the test points located throughout the board which are labeled as TP with a number on the Raspberry Pi to check the power being supplied to it. The schematic of the raspberry Pi can be used to find all the test point locations. We made sure the voltage is around 5V with about 2.5A at each location. This was helpful when troubleshooting for power issues as well.

6.2.2 - PCB Testing

The R&S RTM oscilloscope features a power analysis option. This allows us to measure the current harmonics for different standards including EN61000-3-2 class A, B, C, D, MIL-STD-1399, and RTCA DO-160. It also measures the input including inrush current, power quality, and power consumption. We can look at the switching components as well by using the power converter control measurement. This includes modulation analysis, slew rate, and dynamic on-resistance. The power path measurement can analyze the safe operating area, turn on/off, switching loss, and power efficiency. The output measurement measures the output ripple, transient response, and output spectrum.

We want to improve efficiency, increase power density and reliability, comply with EMC regulations, prove power rail integrity, and decrease thermals with our regulator. We are interested in a well-regulated and low noise DC power from the input power rail. When looking at the input side we can look for power quality, current harmonics, and inrush. We then look at the power going to the transistors of the regulator. This regulates the voltage with the duty cycle or the amount of times it is on versus off. Here we will be looking at power losses, modulation analysis, slew rate, and safe operating range. After the transistors the signals get filtered and rectified so you will get a stepped DC output which will then run power through the rest of the device. When focusing on the output side we will want to look at output ripple, turn on and turn off time, transient response, power supply rejection ratio, and efficiency. We had a test set up similar to the one in **Figure 47**.

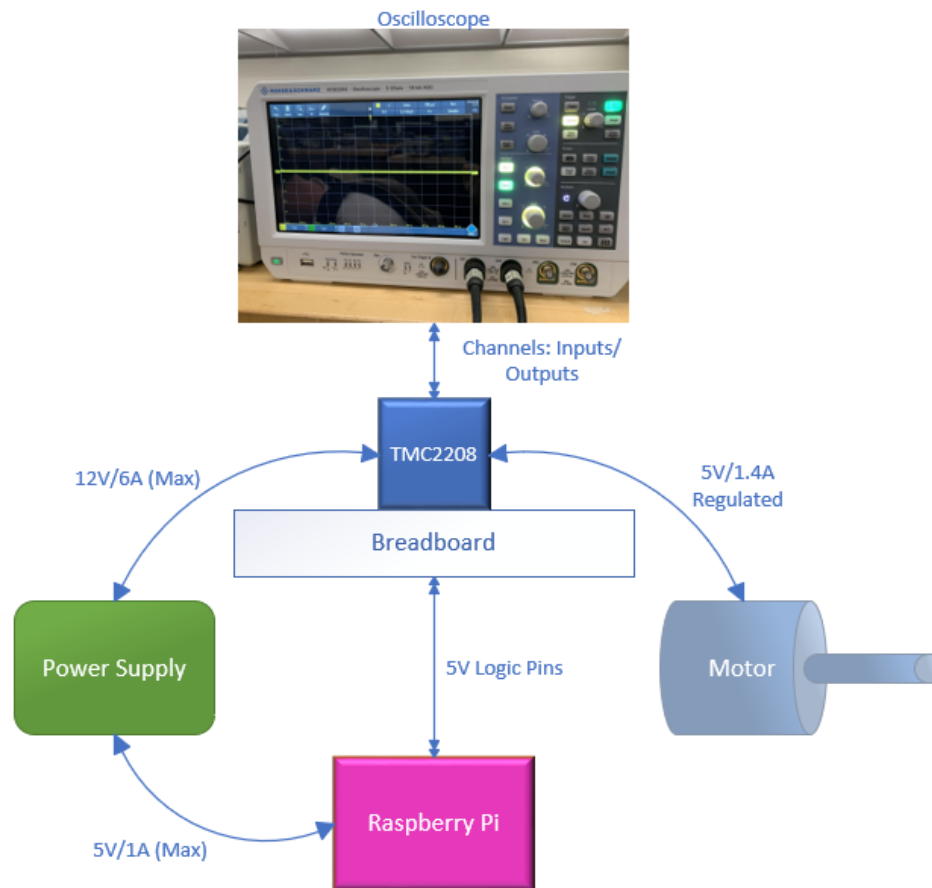


Figure 47: Test Set Up for Oscilloscope

6.2.2.1 - TMC2208 (Motor Driver)

The first thing we will look at will be power quality. We had previously discussed that we can use the multimeter to look at these values, but to get a more detailed explanation we will be using the oscilloscope for the regulator. Make sure to clear any existing settings on the oscilloscope by selecting default set up. Then select power analysis measurement. A differential probe will be connected to the power supply on channel 1 and a current probe will be connected on the input power supply on channel 2. We can then choose the amount of cycles as well. We can then power our circuit and press the auto set. We expect to see about 12V and 6A since this is what the power supply is providing. If not we will want to further analyze why we are not getting these values.

When looking at the output of our regulator we are interested in looking at the output ripple and the transient response. The output ripple causes noise on the power rail which could cause jitter and timing uncertainty which affects the validity of our measurement. We can measure this with the oscilloscope with the output connected to channel one of our oscilloscope and then set the auto set.

We will then use the output ripple measurement. This essentially measures the peak to peak voltage of the signal. We want to minimize the output ripple as much as possible. We can see the difference between when the regulator is placed on versus when it is off to see how it changes. We can also further look at ways to improve the ripple to reduce the amount of noise in the PCB.

6.2.2.2 - Microcontroller

When testing the microcontroller we first had a breadboard design as pictured in **Figure 48**. In order to program the ATMEGA328P we used the pins PB3 (MOSI), PB4 (MISO), PB5 (SCK), and PC6 (RESET). The pins were connected to the same pins on the Arduino Uno, except RESET is placed on digital pin 10 of the Arduino Uno which is the SS pin. Along with these pin connections, pins PD1 (TX) and PD0 (RX) were also connected to TX and RX on the Arduino Uno for serial communication. A 16MHz external crystal was connected to pins PB6 and PB7 with 22pF ceramic capacitors connected to ground. A 10K ohm pull up resistor was placed on the RESET pin. The VCC pins were connected to 3.3V and the GND pins were connected to ground. Below shows the connections between the ATMEGA328P and the Arduino Uno.

After these connections are made, the bootloader is burned onto the microcontroller. The ArduinoISP sketch is uploaded onto the Arduino board, then the Arduino Duemilanove is selected as the board, the Arduino as ISP is selected as the programmer, and then you burn the bootloader. After this is performed the connections for MOSI, MISO, and SCK can be removed. RESET on the ATMEGA328P is now going to be connected to RESET on the Arduino Uno. The microcontroller on the Arduino Uno will now be removed since we are looking to program the ATMEGA328P on the printed circuit board for our project.

The Arduino IDE is used to program the ATMEGA328P. Some things that the ATMEGA328P needs to do include having a serial connection between it and the Raspberry Pi 3B+ so they can communicate as needed. It does this through the TX and RX pins. The other signals that the Raspberry Pi will be sending to the microcontroller include EN, STEP, DIR, and SWITCH which are needed to control the motor. The program on the microcontroller will only output the STEP, DIR, and SWITCH signals if the EN pin is pulled low. Otherwise, it will pull the EN pin high and communicate that the “motor driver is not enabled” on the ATMEGA. DIR controls which direction the motor moves in and STEP controls the amount of steps to turn the motor. SWITCH is connected to an external switch that is used to reset the position of the camera. The ATMEGA328P will be able to communicate effectively between the Raspberry Pi 3B+ and the TMC2208 to

control the motor to make appropriate measurements of the laser with the camera.

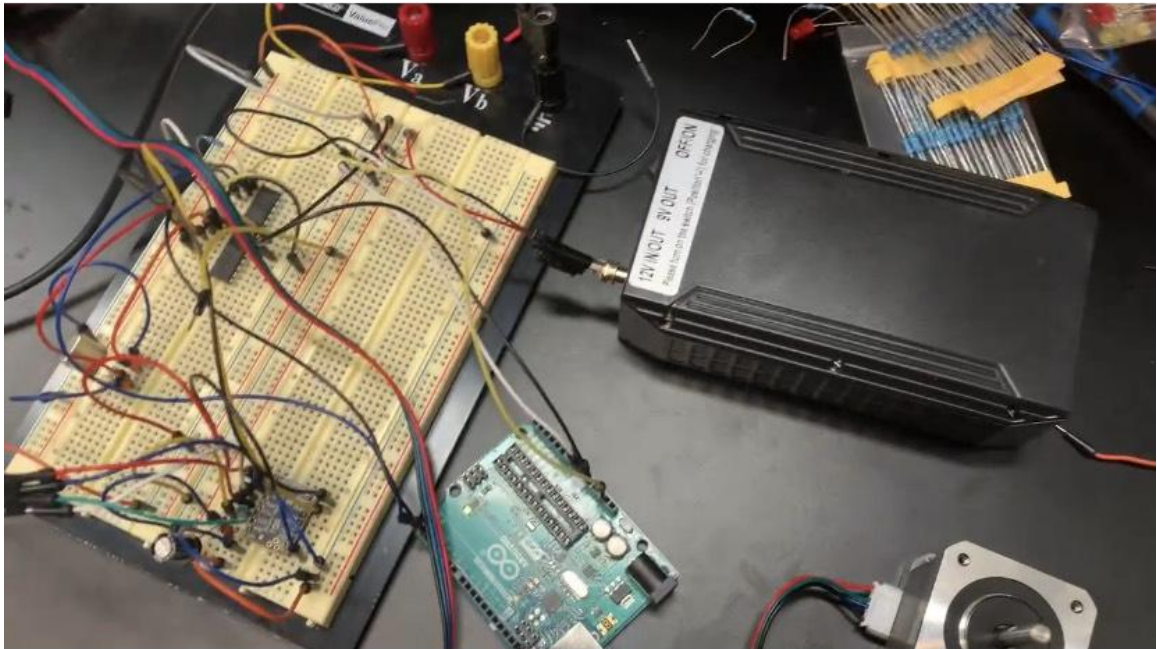


Figure 48. Microcontroller Breadboard

6.2.3 - Stepper Motor Testing

We will be testing our stepper motor using the oscilloscope in the senior design lab. As discussed previously we will have a setup similar to what the design section discusses along with the oscilloscope and probes.

6.2.3.1 - Coils

First we will be testing the motor itself to see if there are any troubleshooting issues with the coils. We will use two channels and connect the probes of the oscilloscope to the motor driver output coil pins. We are using two channels since we have two poles. Once we run our program, we expect to see the output sine waves to be similar in size. We are testing the strength of the magnetic field. It is important to change the scale of the oscilloscope to have the waves at the same scale. We want to make sure that they are similar in size because if not then there could be a problem with one of the coils which we can check by opening up the motor. We also performed a continuity test to check which wires were connected to which coils since the motor did not come with any manual.

6.2.3.2 - Microsteps

Now we can change the probes to be on the switching power supply to look at slew rate to determine how fast our device turns on and off. For our breadboard testing we first set up the TMC2208 with a frequency generator and oscilloscope. The frequency generator simulated the STEP and DIR pins of the we were able

to look at the oscilloscope to see the output of the motor. We simulated these pins by sending a square wave with a 50% duty cycle at and adjusted the frequencies to see the difference with how the stepper motor moved.

Next, we plan to see the differences of the microsteps using an oscilloscope. We will use a potentiometer to adjust the time delay between pulses which will let us control the motor speed. The driver partially energizes the coils to position the rotating positions between the two poles and it does so in a way that resembles a sine wave. The more positions you get between each pole then the better your sine wave will appear. So the better the stepper driver represents a sine wave then the less noise it will produce. The noise is also caused by magnetic fields, current ripple, and sharper frequency. We can use a breadboard with our stepper driver and a potentiometer along with a power supply (most likely will place a bypass capacitor here as well), the raspberry pi, and the motor. To measure the waveform, we will connect the oscilloscope in series with one of the coils to measure current flowing through the coil. We should be seeing a square wave produced. We can change the sample time to make sure we are getting the same square wave. We can increase and decrease the speed and we should still see a similar square wave. We can check each type of microstepping from 2 to 256 microsteps to see the changes and which one is going to be best to use. We can also listen to the stepper motor while we are doing these measurements to see how loud or quiet it is with each change. We will have to adjust the time scale as we increase the steps as well to see the full waveform as the motor is moving slower to achieve these steps. The motor's speed will decrease about half each time we increase the number of steps. Another thing to keep in mind is the torque produced. The more microsteps we have the less torque produced. One way to help with this is to increase the supply voltage to the motor to increase the torque.

6.2.3.3 - Translation Stage

Using the pins on the stepper motor driver IC, the stepper motor can be stepped manually by pulsing the STEP pin low and high. To test the operation of the stepper motor and the attached translation stage, the stepper motor should be stepped manually, and the travel distance of the stage recorded. Any deviations from the calculated theoretical travel distance may signal a problem in the design. While the stepper motor is being driven by software at full load, the current going into the stepper motor should be measured to ensure that it is not exceeding the maximum current rating of 1.5A listed on the datasheet. During a full load test, the temperature of the stepper motor driver should also be measured, as it must supply the load current and will likely heat up doing so. The prototype PCBs will feature breakouts to access the pins of the stepper motor driver and stepper motor, as well as a jumper to be able to measure current going into the motor.

The process of measuring the distance that the translation stage moves when the stepper motor is stepped can be done easily. To perform this test we will first

mount the Raspberry Pi camera module on the translation stage. Then by shining a laser beam onto the camera sensor, an image will be displayed. The starting image will be the reference image, and every step of the motor a new image will be taken. After at least 10 points are taken, the number of pixels the beam has moved can be averaged. This value will tell us how far the stage moves each step of the motor. To measure the accuracy of returning to the same position, we will need to save the position of the starting image. We will then translate the stage to the other end and back, returning to the same position. An image at this position will be saved, and the two images will be compared for any horizontal displacement of the beam. This procedure will be tested at least 5 times to ensure that the accuracy of the device is reliable.

6.2.3.4 - Evaluation Kit

Prior to testing our stepper motor design one option we are considering is purchasing the stepper driver evaluation kit to get a better idea of how the driver works, how to power it properly, and how to program it to control the motor. It will allow us to explore all functions of the TMC2208 including full access to all registers, full functionality, and diagnostics. This way we can test all the functions to find what is best for our design and utilize it. All we need is a power supply and access to Trinamic's TMCL-IDE which is free. The GUI provides easy setup, monitors real time behavior with graphs, and logs and stores data for us. We can simulate the other drivers included with the TMC2208 which are MicroPlyer, SpreadCycle, and StealthChop2. The eval board can be interfaced through USB to our PC to use.[FJ1] When looking for this we found it is in stock but is around \$125 which was a bit pricey. We found that since we are getting many TMC2208 modules that we can use these to test with our Raspberry Pi which will further be discussed in the software testing section.

6.2.4 - Continuity Testing

We will need to place the IC onto the PCB since JLCPCB does not have the part. There are many ways to do this, however, we are interested in soldering the IC directly to the board. The check that it has been properly soldered we can perform a continuity test. A continuity test is used to determine the electrical path between two points. The equipment includes an indicator in series with a power supply. To verify that the connections have been properly made, the indicator will be activated. We can use the multimeter in the lab to achieve this test since it has a continuity setting. To use it we will use the black test lead that will be plugged into the COM connector and the red test lead that will be plugged into the V connector. We can then connect the test leads across the pins that we need to test. The multimeter should beep if the complete circuit is detected and will not beep if the circuit is open.

6.3 - Software Testing

Software testing is necessary in order to ensure that the software properly interfaces with the stepper motor and camera hardware, as well as performs the image-processing computations correctly. The software testing is divided into four sections: physical, network, image processing, and user interface.

6.3.1 - Software Testing Environment

Before the software can be tested, a testing environment must be established. The test environment will consist of a Windows 10 PC with Python 3.10.4 installed, a Raspberry Pi 3 B+ with Python 3.10.4 installed, the TMC2208 Stepper Motor Driver, the NEMA 17 Stepper Motor attached to the linear rail, and the Raspberry Pi Camera V2 NOIR. The Windows 10 PC will be responsible for hosting the test client program, and will be connected to the RPi over an Ethernet connection. The RPi will host the test server program, and along with the Ethernet connection will have the Pi Camera V2 NOIR connected to the CSI connector on the top of the board, and the TMC2208 to its GPIO pins. This setup can be configured outside of the device housing on a desk or bench surface for easy access to each component of the testing environment. After the tests over Ethernet are conducted, the RPi will then be configured to communicate over a wireless LAN connection to the Windows 10 PC.

6.3.2 - Hardware-Software Interface Testing

Testing of the Hardware-Software interface focuses on driving the stepper motor properly, and retrieving image data from the sensor. The functionality of the stepper motor is wrapped in a Python class, for ease of access for testing and programming. In order to verify that the Raspberry Pi is moving the stepper motor to the correct location, a simple program will be used that will drive the stepper motor, moving the translation stage to a fixed point of 10mm, and the translation can be measured physically by capturing an image from a fixed location and counting the pixels to verify that the drive is fault-free. This simple program will also include functionality to zero and conduct the 10mm translation repeatedly to ensure that the translation distance does not deviate from this 10mm value over repeated translations beyond the specified 0.03mm in the linear rail datasheet. The software should also ensure that once the zeroing micro-switch is activated, that the stepper motor immediately halt stepping to avoid potential damage to the switch or motor.

The camera is also handled by a Python class provided by the library PiCamera, which provides class functions for interfacing with the camera. Another simple program will be created to ensure that the camera is able to capture and save image data while also minimizing background image noise through background subtraction and visualizing any obstructions in the beam image path. The camera will capture images of a known reference beam with a set of reference camera parameters to compare for deviations in the capture quality and data acquisition.

Figure 49 showcases an example of a “bad image” from the sensor, where high levels of background noise are present, and a dust particle obstructing a small area of the beam-cross section.

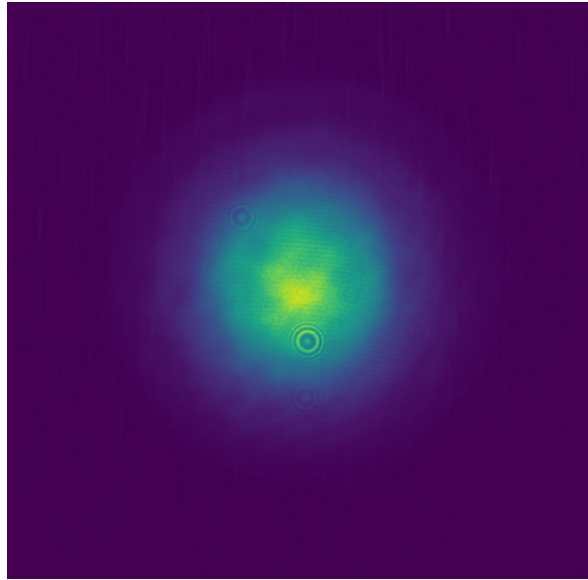


Figure 49: Example Beam-Cross section with background streaking and dust

As seen in the image, the green laser is focused on the center of the sensor, but there is a lot of background noise being captured by the sensor. The light streaking is from stray reflections coming from the beam, and the circles are caused from reflections off dust particles. In order to improve the image quality, a tube or other light blocking device must be placed in front of the image sensor to block a significant portion of the ambient light and stray reflections from the beam. The final design incorporates a tube after the focal lens to improve the fidelity of the beam image.

Dust is shown by the small dark circles visible on the bottom side of the beam. The aforementioned tube alleviates this issue somewhat, but proper care must be taken such that the image sensor does not have any obstructions. This heavily influences the values calculated by the image processing algorithms. Another problem that must be addressed is camera sensor saturation. The optical design attenuates the incoming signal by a large percentage, but the camera has a variable exposure time, which needs to be found every time a beam is put under test. Connecting the Raspberry Pi to an external monitor through the HDMI port allows a live-view of the camera feed, which can help identify saturation spots or any sensor obstructions.

The exposure time can be adjusted manually using hard-programmed values in the live-view program, or be adjusted automatically using the auto exposure subroutine. The auto exposure subroutine should be tested and tweaked to ensure that a suitable exposure time is found for all lasers under test, as well as tested for convergence. An example where the auto-exposure subroutine may

not converge is if the intervals of increment/decrement are too large, the exposure time will increase to a point where there is saturation, then decrease to a point where all values in the image are beneath the desired threshold, and repeat.

6.3.3 - Network Testing

Since the design uses python sockets, testing of the network can be conducted easily using a test client program configured on the same LAN as the server. This test client program was used to simulate commands coming from the socket thread, and verifying the server's responses. The test client program will send a series of commands asking the server to run a scan, halt the scan, re-run the scan, then retrieve the image data from the server, and the server is expected to respond accordingly. The messages sent and received by both the server and the client will be parsed into their arguments, and logged for evaluation to ensure that messages are being transferred without issues such as data-loss or retransmission errors

6.3.4 - Measurement Testing

To verify the measurement capabilities of our device, the following test setup was used. A ThorLabs HNL020LB HeNe laser was used as the beam-under test. The beam first passes through a 5x beam expander constructed out of 38mm and 200mm focal length lenses. After, the laser was aligned into the system using two external uncoated mirrors through the two irises mounted on the front of the device. The internal ND 3.0 filter was placed in the beam path to limit the power of the beam hitting the sensor.

The device moved the camera sensor to 28 points along the beam propagation path, capturing 10 points 1 mm apart near the focal point to provide the curve-fitting algorithm with an accurate description of how the beam focuses. **Figure 50** shows the measured waists in the x-axis with respect to their position of measurement. The curve generated from the Seigman curve fit results can be seen in orange, and the measured waists are represented in blue. Note that all the position values are relative to the ball screw, and to convert to back-focal length, you must add 201mm.

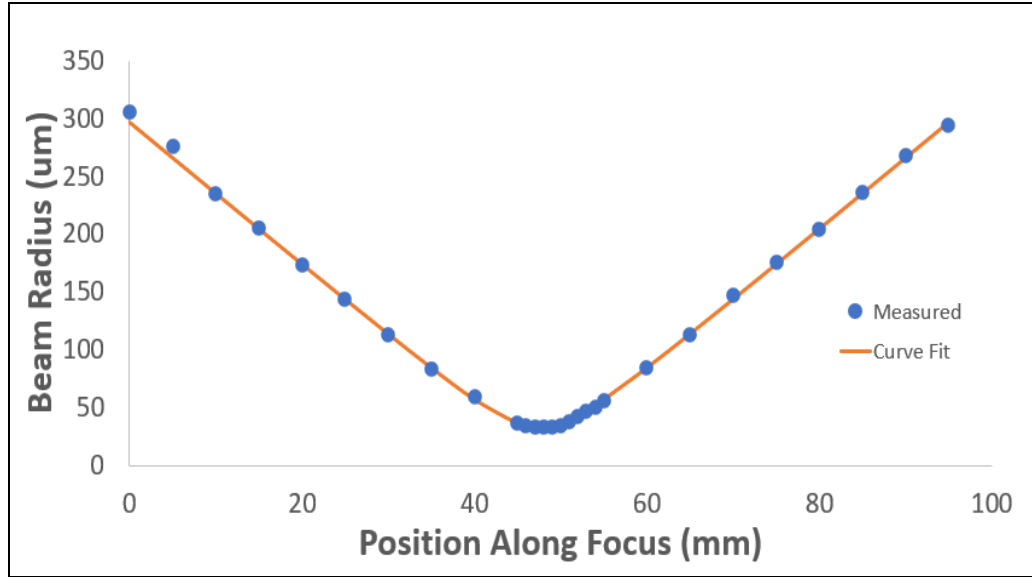


Figure 50: Beam Waist measurements plotted against the position along the ball screw travel length.

Fitting all the waist measurements and measurement positions to equation (9) yielded $W_0 = 29.5989 \pm 1.10 \text{ um}$, $M^2 = 0.96767 \pm 0.0695$, and $Z_0 = 47.6378 \pm 0.115 \text{ mm}$. The M^2 value calculated by the device is less than one, but the uncertainty puts the real M^2 between 1 and 1.03, which is within the specification of the HL020LB HeNe laser used (between 1 and 1.1).

Error analysis was done and it was found that although our camera has a 1.12 pixel size, we are limited by this size when measuring the waist specifically near the focus. This error will cause the M^2 to fall below one. Verification was done by comparing the far-field divergence angle of our measurements and a knife edge along the focus. The divergence angle from the curve fit was 6.5 mrad. Points along the focus were measured using a knife edge and can be seen in **Figure 51**. Diameters for each position are: $z = 0 \text{ mm}$ (blue) is 3.30 mm, $z = 100 \text{ mm}$ (orange) is 1.83 mm, $z = 160 \text{ mm}$ (gray) is 1.09 mm, and $z = 230 \text{ mm}$ (yellow) is 0.188 mm.

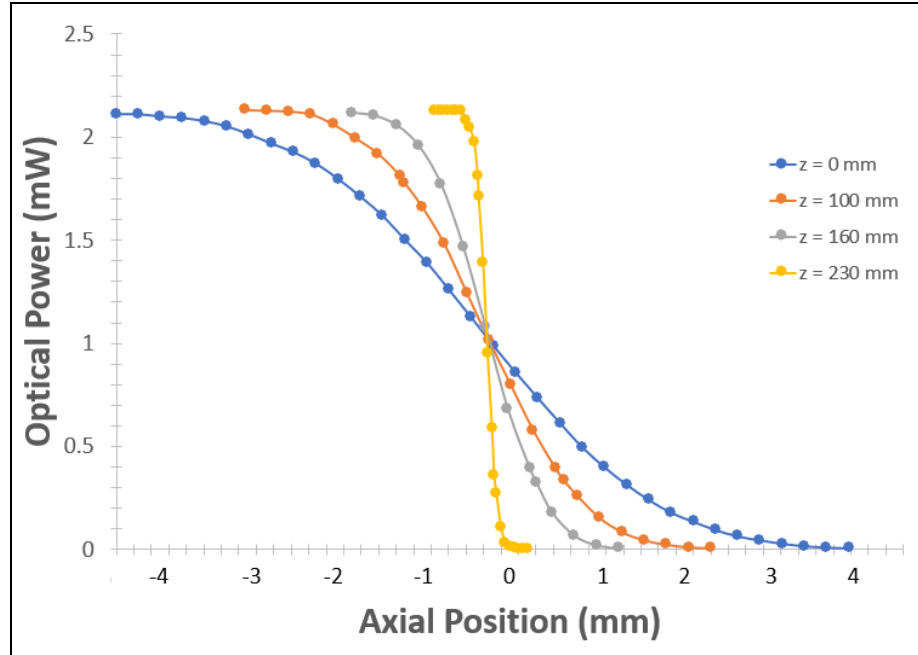


Figure 51: Knife-edge measurements in the x-axis along the focus of the lens.

The far-field divergence angle calculated by the knife-edge was 6.15 mrad. This gives our device around 6% accuracy. Points near the minimum waist were not able to accurately be measured due to the manual measurements of the translation stage, and power meter, as well as laser power fluctuations making it hard to read an exact value. With this verification, our device has an accuracy greater than 90%.

6.3.5 - User Interface Testing

In order to test the graphical user interface for all of its functionality, the buttons were configured to print to console when they were pressed, as well as the message being sent to the socket thread over the queue. The entry fields were verified to be vetted properly by filling them with non-valid characters and attempting to submit them. To test the plots, pre-captured image data was loaded into the socket thread to simulate a series of captures coming from the server, and watching to see if the appropriate interface elements were being updated as the images were being measured.

6.3.6 - Measurement Time Testing

In order for the device to be usable, the device must be able to capture, transmit, and measure a beam waist in a reasonable amount of time. The specification we set was 20 minutes per M^2 measurement. A minimum of 11 images need to be captured and transmitted for a valid M^2 measurement, so a goal of at least 1 minute and 49 seconds per image was set. This time includes the time the server needs to determine the appropriate exposure time for the capture, the capture and average itself, the transmission, and the measurement by the socket thread.

For this test, the independent variables changed were the number of images per capture and doing the Bayer demosaic on the client or server. Each test was conducted over 100 images over differing locations on the beam propagation axis to verify accuracy. **Table 28** outlines the results of this test

Test	Average (s)	Min (s)	Max (s)
1 image per capture, server bayer	24.25	22.75	39.32
1 image per capture, client bayer	7.8	7.1	15.82
5 images per capture, server bayer	43.52	41.46	58.42
5 images per capture, client bayer	23.74	22.41	38.54

Table 28: Measurement Time Test Results

Doing the Bayer demosaicing on the client significantly improves the time per image, and was the option selected for the final design. Although the 43.52 seconds per image of the ‘5 images per capture, server bayer’ was within the set goal, the improvement of measurement time of the client bayer option significantly improved device usage. The maximum values of the chart are caused by the auto exposure failing to find a suitable exposure time for the camera, resulting in several seconds added to the measurement. Although the 1 image per capture produces the lowest average time, the accuracy of the measurement suffers without a multiple image average, and caution should be used when recording measurements generated this way.

7.0 - Project Operation

The objective of this device is to measure the beam quality factor (M^2 value) of a laser beam and have the results be available on a computer running Windows. This section outlines the typical user experience when using the device for the first time. The server software will be pre-loaded onto the internal Raspberry Pi, and the client program will be packaged as two executable program files for Windows and two object files for Linux. The two program files for the client are the main, scanner interface, and the initial configuration program. If the device is directly connected to the network through the CAT5e port, then the configuration program can be bypassed, as the network configuration will be handled by the network host.

If the user wishes to connect the device over 802.11 a/c/n wireless, then the configuration program must be run to inform the Raspberry Pi of the wireless access point SSID and password. Before any software is run, ensure that all ND filters are active and in the beam path. This ensures that the camera is not damaged. Then the device must be placed in the beam propagation path of the beam under test. The device hosts two apertures to align the incoming beam

along with the outgoing beam to ensure squareness. Internal apertures will help assist beam alignment. Note that external optics may be required to properly align the beam into the device. Once the device is properly aligned, the device may be powered on. When the server powers on, it will begin by attempting to zero the stepper motor, and it will check if there is an existing network connection. If this check returns successful, it will begin opening the sockets and entering the main program loop, waiting for commands from the client. If not, the program will open a bluetooth server, which is utilized by the client to configure the network settings.

If no prior network connection is established, the user must connect their computer to the Raspberry Pi over bluetooth, and run the configuration program to pass on the access point settings to the server. After the server receives these parameters, it will enter the main program loop. Once the server is properly configured, the user may then open the other program to interface with the server over LAN. Once the scanner interface program is opened, the user will be presented a graphical user interface to control the device. Before a scan is run, the beam-under-test parameters must be inputted into the interface under the button labeled “Scan Parameters”.

Once these have been inputted, the “Scan” button will become enabled, allowing the user to begin a scan. After the “Scan” button is selected, the client will send the server the parameters for the scan, and the device will begin capturing images of the beam along the beam propagation path. If the user would like to halt the scan at any point, the “Halt” button can be used to signal the server that the scan has been canceled. As the device is capturing the images of the beam, the text log in the graphical user interface will update with the current information, to allow the user to know that the scan is making progress. The same is true when the device has finished taking the images, and is sending them back to the client. Once all the images have been received by the client, the server will enter an idle state, and the “Halt” button will change functionality to signal the client to halt any image processing in progress.

The program will then begin the image processing subroutine to calculate the second-moment beam width for each of the scan images, followed by curve fitting to find the M^2 value. As each of the second-moment beam widths are calculated, the results will be plotted on the GUI on the corresponding plot. To visualize the width of the beam at each point. Finally, once the M^2 value is computed, it will be displayed in its corresponding text box on the user interface, as well as logged in a file. The client will then enter an idle state, waiting for the next scan to be signaled. The user can close the client program by clicking the “Quit” button, and the server can be shut down by simply powering off the device. If the device is powered off, but the client program is still running, the client program will detect the drop in connection to the server, and will display that to the user in the text log, along with disabling the “Scan” button. A similar thing will

happen if the client closes while the server is still on, but instead the server will enter an idle state waiting for a new connection from the client.

8.0 - Administrative Content

This section will further discuss the budget given to us by CMS Laser along with our bill of materials (BOM) for both the optical and electrical components. We also will discuss the milestones we have set to achieve associated with the design process.

8.1 - Budget

Our sponsor CMS Laser will be financing this project. They have given us a budget of \$1000. We will find the most cost effective materials for this project to stay within the budget. The list of materials we need are listed in **Tables 29, 30** and **31**.

8.2 - BOM (Bill of Materials)

Included in this section is the bill of materials for the optical and hardware components that we plan on purchasing after the designs have been finalized. The BOM is divided into three parts: one optical, one hardware, and one printed circuit board. The optical BOM is shown in **Table 29**, and holds all the parts for the optical design, including parts like the wedges, lenses, apertures, and the corresponding mounts for all. **Table 30** holds the BOM for the hardware design, and covers the stepper motor and related parts, and power supply respectively. **Table 31** shows the parts needed for the printed circuit board. At the end of Senior Design I, both of these BOMs were sent to our project sponsor for ordering. We planned a design such that the BOM does not exceed the amount that the sponsor is providing us, however, due to needing additional parts for testing and ordering two printed circuit boards we paid for more out of our pockets.

8.2.1 - Optical BOM

	Quantity	Vendor	Item Stock #	Description	Price Each	Price All
1	2	Edmund Optics	#45-555	1° Nom. Uncoated, N-BK7 Wedged Prism	\$44.00	\$88.00
2	1	Edmund Optics	#43-651	2° Nom. Uncoated, N-BK7 Wedged Prism	\$44.00	\$44.00
3	1	Edmund Optics	#13-787	25.0/25.4mm Optic Dia., SM1 Thin Mount, M4	\$17.00	\$17.00
4	3	Edmund Optics	#15-866	E-Series Kinematic Mount, 25.0 / 25.4 mm Optic Diameter	\$39.00	\$117.00
5	2	Edmund Optics	#32-622	24mm Outer Diameter, Iris Diaphragm	\$48.00	\$96.00
6	1	Edmund Optics	#67-150	25.0mm Dia. x 250.0mm FL, Uncoated, Plano-Convex Lens	\$31.00	\$31.00
7	1	Thorlabs	ND15A	ND15A - Reflective Ø25mm ND Filter, SM1-Threaded Mount, Optical Density: 1.5	\$68.92	\$68.92
8	1	Thorlabs	ND30A	ND30A - Reflective Ø25mm ND Filter, SM1-Threaded Mount, Optical Density: 3.0	\$77.37	\$77.37
9	1	Thorlabs	ND40A	ND40A - Reflective Ø25mm ND Filter, SM1-Threaded Mount, Optical Density: 4.0	\$77.37	\$77.37
10	4	Edmund Optics	#58-952	50.8mm Length, M4 Stud, Steel Post	\$9.75	\$39.00
11	4	Edmund Optics	#58-972	50.8mm Length, M6 Thread, Post Holder	\$13.50	\$54.00
12	4	Edmund Optics	#15-859	Clamping Fork	\$8.90	\$35.60
13	4	Thorlabs	BE1/M	BE1/M - Ø31.8 mm Studded Pedestal Base Adapter, M6 Thread	\$10.31	\$41.24
Total Price						\$786.50

Table 29: Optical BOM

8.2.2 - Hardware BOM

	Quantity	Vendor	Item Stock #	Description	Price Each	Price All
1	1	Amazon	701715439610	2PCS TMC2208 V1.2 3D Printer Stepper Motor Driver, Upgrade TMC2208 Stepper Driver Module with Heat Sink Screwdriver for 3D Printer Controller Mother Boards Reprap MKS Prusa i3 Ender-3 Pro	\$8.99	\$8.99
2	1	Amazon	B06XNN9F67	HTTX 10-Pack 5.5mm x 2.5mm DC Socket Male Barrel Jack, Solder On DIY Custom Power Plug Adapter	\$7.99	\$7.99
3	1	Amazon	303277375131	Befenybay 100mm Length Travel Linear Stage Actuator with Square Linear Rails Ballscrew SFU1605 with NEMA17 Stepper Motor for DIY CNC Router Parts X Y Z Axis	\$73.80	\$73.80
4	1	Amazon	B01337QXMA	TalentCell Rechargeable 72W 100Wh 12V/8300mAh 9V/11000mAh 5V/20000mAh DC Output Lithium ion Battery Pack for LED Strip, CCTV Camera and More, Portable Li-ion Batteries with AC/DC Charger	\$53.79	\$53.79
5	1	JLPCB	-	Custom PCB Board, 5 Pieces	\$13.24	\$13.24
6	1	Own	-	Raspberry Pi B+ 3	\$0.00	\$0.00
7	1	Own	-	Raspberry Pi Camera V2.1	\$0.00	\$0.00
Total Price						\$157.81

Table 30: Electrical BOM for Power Supply, Motor and Related Parts

8.2.3 - PCB BOM

ID	Name	Designator	Footprint	Quantity	Manufacturer Part	Manufacturer	Supplier
1	22nF	C1	C0603	1	CL10B223 KB8NNNC	SAMSUNG	LCSC
2	100nF	C2,C4,C5, C7,C8,C9, C10,CBST,	C0603	9	CC0603K RX7R9BB 104	YAGEO(国巨)	LCSC

		CINX					
3	2.2uF	C3	C0603	1	CL10A225 KO8NNNC	SAMSUN G(三星)	LCSC
4	100uF	C6	CAP-SMD _BD6.3-L6 .6-W6.6-L S7.2-FD	1	RVE100U F16V67RV 0046	KNSCHA(科尼盛)	LCSC
5	10nF	C11	C0603	1	0603B103 K500NT	FH(风华)	LCSC
6	22pF	C12,C13	C0603	2	CL10C220 JB8NNNC	SAMSUN G(三星)	LCSC
7	15pF	CFF	C0603	1	CL10C150 JB8NNNC	SAMSUN G	LCSC
8	10uF	CIN1,CIN2	C0805	2	GRM21BR 61E106MA 73L	Murata Electronics	LCSC
9	47uF	COUT	C1206	1	GRM31CR 60J476KE 19L	MuRata	LCSC
10	1uF	CREG	C1206	1	CL31B105 KBHNNNE	SAMSUN G	LCSC
11	DC-005-5 A-2.5	DC2	DC-IN-TH _DC-005-5 A-2.5	1	DC-005-5 A-2.5	XKB Enterprise	LCSC
12	Header-M ale-2.54_1 x4	H1,H3	HDR-TH_4 P-P2.54-V- M	2	B-2100S0 4P-A110	Ckmtw(灿 科盟)	LCSC
13	Header-M ale-2.54_1 x3	H2	HDR-TH_3 P-P2.54-V- M-1	1	Header2.5 4mm 1*3P	BOOMELE	LCSC
14	2.2uH	L1	L0805	1	CMI20120 9U2R2KT	FH(风华)	LCSC
15	180mΩ	R1,R2	R0603	2	0603WAF 180LT5E	Uniroyal Elec	LCSC
16	10kΩ	R3,R4,RE N	R0603	3	0603WAF 1002T5E	UniOhm	LCSC
17	22.1kΩ	RFBB	R0603	1	RMC0603 22.1K1%N	Tyohm	LCSC
18	73.2kΩ	RFBT	R0603	1	RMC0603 73.2K1%N	Tyohm(幸 亚电阻)	LCSC

19	100kΩ	RPGOOD	R0603	1	0603WAF 1003T5E	UNI-ROYA L(厚声)	LCSC
20	K2-1107S T-A4SW-0 6	SW1	KEY-SMD _L6.2-W3. 6-LS8.0	1	K2-1107S T-A4DW-0 6	韩国韩荣	LCSC
21	5002	TP1,TP2,T P3,TP4,TP 5,TP6,TP7 ,TP8,TP9, TP10,TP1	TEST-TH_ BD2.54-P1 .39	11	5002	Keystone	LCSC
22	TMC2208- LA-T	U1	QFN-28_L 5.0-W5.0- P0.50-BL- EP	1	TMC2208- LA-T	TRINAMIC	LCSC
23	TPS56628 DDAR	U2	SOIC-8_L 4.9-W3.9- P1.27-LS6 .0-BL-EP	1	TPS56628 DDAR	TI(德州仪 器)	LCSC
24	ATMEGA3 28P-AU	U3	TQFP-32_ L7.0-W7.0- P0.80-LS9 .0-BL	1	ATMEGA3 28P-AU	MICROCH IP	LCSC
25	16MHz	X1	CRYSTAL- SMD_4P-L 2.5-W2.0- BL	1	7E16000E 09UCG	SJK(晶科 鑫)	LCSC

Table 31: BOM for Printed Circuit Board

8.3 - Milestones

This section outlines the two terms that the project is split up into including their proposed dates to complete certain tasks to stay on target with our end date. Tables 27-28 show Senior Design Term 1 and 2 deadlines respectively for project completion goals.

8.3.1 - Term 1 (Senior Design I)

For Senior Design I, the main focus is forming a group, and realizing a plan through an interactive document that we will work on throughout the semester. Within the first week, our group will be formed, and ideas for the project will be considered and evaluated for realization. At this stage, rough ideas for the project's subsystems will be considered, and all but one will be considered for continuation. Within the next two weeks, the final project ideas will be selected,

and we will form the Divide and Conquer Version 1 document. The week after the feedback for Divide and Conquer Version 1 is received, a new version will be created solving the problems of the first and adding more details to the design. Within the next month, the Divide and Conquer Version 2 will be extended to 60 pages, then 100 pages. Finally, at the end of Senior Design 1, the final documentation will be completed and be ready for submission. **Table 32** outlines these milestones for Senior Design I.

Week	Description
1 (5/15/22-5/21/22)	Form project group and start brainstorming project ideas
2-3 (5/22/22-6/4/22)	Finalize project idea and complete Divide and Conquer Version 1
4-5 (6/5/22-6/18/22)	Attend boot camp and complete Divide and Conquer Version 2
6-8 (6/19/22-7/9/22)	Start and complete 60 page draft documentation
9-10 (7/10/22-7/23/22)	Complete 100 page draft documentation. Start ordering parts from BOM.
11-12 (7/24/22-8/6/22)	Complete final documentation.

Table 32: Senior Design I Goal Milestones with dates

8.3.1 - Term 2 (Senior Design II)

In Senior Design II, we began building prototypes based on the design document created in Senior Design I. During the weeks between Senior Design I and Senior Design II, we finalized the Bill of Materials, and began placing orders for parts. We hope that because we are doing this two weeks before the start of Senior Design II that we will receive our parts before or during the first week. After all of the parts were received, we built the first prototype between weeks 2-5, which includes but is not limited to mounting the optics on the baseplate, soldering our components to the PCB and performing the electrical tests outlined in the Hardware Testing section, and finalizing the first draft of the software and performing the validation tests on that as well. We then spent the following three weeks (weeks 6-9) testing and making adjustments to the design as needed. By the 10th week, the device was mostly functional with only minor tweaks required. After the final build was completed, we began working on the documentation and finally presenting our device to the university faculty. An outline of the milestones for Senior Design II is displayed below in **Table 33**.

Week	Description
1 (8/21/22-8/27/22)	Start to receive parts and start the prototype • Parts ordered (8/01/22 - 8/06/22) • PCB arrival (8/22/22)
2-5 (8/28/22-9/24/22)	Complete building prototype • Weeks 2-3 ◦ Build optical prototype ◦ Build electrical prototype ◦ Software code writing • Weeks 4-5 ◦ Combine electrical and software prototype
6-9 (9/25/22-10/22/22)	Testing and redesign • Weeks 6-7: Individual testing ◦ Stepper Motor ◦ Translation rail ◦ Camera sensor ◦ Software • Weeks 8-9: Combined testing ◦ All sections combined ◦ Testing and tweaking
10-13 (10/23/22-11/19/22)	Final build • Weeks 10-11 ◦ Build device structure ◦ Transport components • Weeks 12-13 ◦ Test and tweek final build
14-16 (11/20/22-12/10/22)	Final report and presentation • Weeks 14-16 ◦ Finalize for CMS Laser

Table 33: Senior Design II Goal Milestones with dates

9.0 - Conclusion

The design presented in this paper proposes a novel way to measure beam-cross section width and M^2 values of a laser beam, in a compact and portable form factor. Once aligned, the beam-under-test will enter the device, have its power reduced significantly through means of wedges, a beam dump,

and neutral density filters, then be captured on a CCD image sensor. A Raspberry Pi 3 B+ is used to host a server that will act as the gateway to the hardware for the client program. When requested by the client, the server program will drive the stepper motors, driving the translation stage down the beam-propagation axis to stop at at least 10 pre-calculated points. The server program will then capture images at each stop to be sent back to the client program for processing. The whole design will be powered by an external 12V source, with an internal switch-mode power supply, allowing for a diversity of powering options from traditional wall plugs to battery packs

The reason why our group chose to take on this project is to expose ourselves to the creation of test equipment for the optical industry. From what our research has exposed, there are not many commercial or hobby M^2 measuring devices on the market, and we believe that the creation of this device at this price point will be important in showing off the group's capabilities to bring such an idea into fruition. Our objective for this project is to challenge ourselves to design and construct a device that will automate the measuring processes of a beam's quality factor as well as beam diameter.

With the financial support of CMS laser, the group believes that the parts selected using the budget provided are adequate in ensuring the construction of a physical prototype within the first few weeks of Senior Design II. With the design as this paper is written, we are currently under the set budget of \$1000, so we have some room to make mistakes. If we do end up going over the set budget, the group can purchase parts out of pocket to ensure that the design is realized. There are many obstacles that have been faced, and many still yet to be encountered. As our group is composed of three students of the College of Engineering and Computer Science and one student of the College of Optics and Photonics, overcoming the knowledge barrier was and still proves to be a hurdle that we have yet to fully overcome. All of the optical calculations and supporting theory had to be learned in order to realize these concepts in software.

One potential issue we faced is the mounting of the optical and electrical hardware onto the baseplate, as none of the members of the group have experience drilling and threading holes in aluminum. If a drill hole is done incorrectly, the base plate may have to be scrapped, as the hole locations are specific to the design. Another issue was threading the holes with a thread tap. If a tap breaks inside of the base plate, the plate would have to be scrapped. Along with the base plate, the cover may prove to be challenging to work with for the same reasons. There is also the hazard of metal shavings and sharp edges that we must take precautions for on both.

Another potentially hazardous issue that the group faced was working with high wattage lasers, as none of the students outside of the College of Optics and Photonics have experience working with such optical systems. Extra precaution must be taken to ensure that none of the members get hurt while doing validation

testing for their respective systems. Proper PPE must be worn at all times around such lasers and care must be taken such that no dangerous reflections will be pointed in the direction of the group members.

10.0 - Appendices

10.1 - References

- [1] A4988 Stepper Driver
<https://www.tme.eu/Document/25459777e672c305e474897eef284f74/POLOLU-2128.pdf>
- [2] Discharge curve 6000mAh TalentCell
<https://power.tenergy.com/productimages/li-ion/Data%20sheet%2018650%202200mAh.pdf>
- [3] 7805 Voltage Regulator
<https://datasheetgo.com/7805-voltage-regulator/>
- [4] Tanenbaum, Andrew S., and David J. Wetherall. *Computer Networks 5th By Andrew S. Tanenbaum (International Economy Edition)*. Prentice Hall, Indian International Ed.
- [5] NEMA 17 Stepper motor
https://www.amazon.com/STEPPERONLINE-Bipolar-Stepper-22-6oz-Extruder/dp/B00PNEQ79Q?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&psc=1&smid=A WQBCGWISS7BL
- [6] <https://x-engineer.org/dc-dc-converter/>
- [7] <https://www.digikey.com/en/maker/blogs/introduction-to-dc-dc-converters>
- [9] <https://www.electronicshub.org/types-of-batteries/>
- [10] <https://us.aspina-group.com/en/learning-zone/columns/what-is/005/>
- [11] <https://www.automate.org/blogs/understanding-stepper-motor-types-and-operating-modes-for-motion-control>
- [15] <https://solectroshop.com/en/blog/choosing-the-right-type-of-motor-driver-for-your-project-n14>
- [16] <https://gesrepair.com/motor-drive-types/>
- [17] <https://ieeexplore.ieee.org/document/1309986>

[18] USBXpress(™) Family CPM2102N Data Sheet,
https://www.mouser.com/datasheet/2/368/cp2102n_datasheet-1634912.pdf

[19] Future Devices International Ltd. FT230X (USB to BASIC UART IC)
https://www.mouser.com/datasheet/2/163/DS_FT230X-5395.pdf

[20] <https://www.gentec-eo.com/products/ba16-60s>

[21] <https://www.electricaltechnology.org/2021/01/types-of-electric-motors.html>

[22]
https://www.amazon.com/Torque-Stepper-Motor-425oz-Router/dp/B00PNEPW4C/ref=sr_1_1?crid=1WCTSPPEZAE0&keywords=nema+23+3Nm+4.2A&qid=1658267986&srefix=nema+23+3nm+4.2a%2Caps%2C128&sr=8-1

[23] <https://webstore.iec.ch/publication/27451>

[24] <https://webstore.iec.ch/publication/61994>

[25] <https://webstore.iec.ch/publication/66712>

[26] <https://webstore.iec.ch/publication/62426>

[27] <https://webstore.iec.ch/publication/6578>

[28]
https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC220x_TMC2224_datasheet_Rev1.09.pdf

[29] 1 Keln J. Kuhn, Laser Engineering, 1st ed. (Prentice Hall (Higher Education Division, Pearson Education), 1998), p.512.

[28]
<https://www.mouser.com/ProductDetail/ADDA/AD0812HS-D71?qs=MTB16yVWt2KzM%252B%252BN%2FY2zng%3D%3D>

[29] <https://store.cwc-group.com/n5010b1.html>

[30]
<https://www.3ccooler.com/cooler-master-12025-a12025-20rb-3bn-f1-df1202512rf-un-r4-l2r-20ac-gp-12v-0-37a-3-wires-led-cooler-fan.html#product-details-tab-specification>

[31]
http://www.autoflexible.com/file_upload/product/attach/NEMA%2017.pdf

- [32] <https://www.protoexpress.com/blog/pcb-power-supply-design-considerations/#:~:text=If%20you%20are%20considering%20a.provide%20exceptionally%20noiseless%20voltage%20outputs.>
- [33] <https://www.fluke.com/en-us/learn/blog/digital-multimeters/how-to-test-for-continuity>
- [34] <https://www.youtube.com/watch?v=Ut3Mj0AX5T4>
- [35] <https://www.iso.org/obp/ui/#iso:std:iso:11146:-1:ed-2:v1:en>
- [36] <https://www.thorlabs.com/thorproduct.cfm?partnumber=BP209-VIS>
- [37] <https://www.thorlabs.com/thorproduct.cfm?partnumber=M2MS>
- [38] <https://www.ophiropt.com/laser--measurement/beam-profilers/products/Beam-Profiling/Camera-Based-Beam-Profiling-with-BeamGage/SP932U-Beam-Profiling-Camera>
- [39] <https://www.raspberrypi.com/documentation/accessories/camera.html>
- [40] 1 James Keaveney, Review of Scientific Instruments 89 (2018).
- [41] https://www.rp-photonics.com/beam_quality.html
- [42] <https://www.youtube.com/watch?v=ah5FbLlOfUE>
- [43] https://www.rohde-schwarz.com/us/products/test-and-measurement/oscilloscopes/rs-rtm3000-oscilloscope_63493-427459.html
- [44] <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all>
- [45] Corporation, Atmel. *8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash Datasheet*, vol. 7810, no. D, Nov. 2009, pp. 1–293.
- [46] “Customize TPS56628DDAR - 8V-12V to 3.30V @ 6A.” *WEBENCH Power Designer*, Texas Instruments, <https://webench.ti.com/power-designer/switching-regulator/customize/13?noparams=0>.

- [47] “From Arduino to a Microcontroller on a Breadboard.” *Arduino*, 23 Nov. 2022,
<https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoToBreadboard>.
- [48] Instruments, Texas. *4.5V To 18V Input, 6-A Synchronous Step-Down Converter with Eco-Mode*, A, Oct. 2013, pp. 1–26.
- [49] Trinamic, ADI. *TMC2208/2 & TMC2224 Family Datasheet*, vol. 1, no. 13, 1 Nov. 2016, pp. 1–81.