

# P.I.D.I.

## Particle Inspector via Darkfield Illumination

### Authors:

Christian Carroz	Kirolos Kellini	Julian Lopez	Ryan Palumbo
Computer Engineering	Photonics Sciences And Engineering	Photonics Sciences And Engineering	Mechanical Engineering

### Reviewer Committee

Dr. Kyu Young Han	CREOL	Associate Professor
Dr. Azadeh Vosoughi	ECE	Professor
Dr. Hadi Mardani Kamali	ECE	Assistant Professor
Dr. Sagnik Mazumdar	MAE	Lecturer
Dr. Navneet Dhindsa	ASML	Sponsor
Dr. Lirong Wang	ASML	Sponsor
Dr. Akram Raub	ASML	Sponsor
Dr. Kimi Huang	ASML	Sponsor
Dr. Andres Guevara	ASML	Sponsor

### Advisors:

Dr. Chung Yong Chan	ECE	Senior Lecturer
Dr. Aravinda Kar	CREOL	Professor



# Table Of Contents

<b>Chapter 1 – Executive Summary .....</b>	<b>1</b>
<b>Chapter 2 - Project Description .....</b>	<b>2</b>
2.1 Project Background and Motivation .....	2
2.2 Existing Products .....	2
2.3 Project Objectives and Goals .....	3
2.4 Project Features and Functionalities .....	5
2.5 Project Specification .....	6
2.6 Block Diagrams .....	9
2.7 House of Quality .....	13
<b>Chapter 3 – Research and Investigation .....</b>	<b>14</b>
3.1 Hardware Selection.....	14
3.1.1 Camera Technology .....	14
3.1.1.1 Camera Technology Selection.....	15
3.1.1.2 CMOS Camera Product Comparison .....	16
3.1.1.3 IDS U3-3880CP-C-HQ .....	17
3.1.1.4 CS165CU .....	18
3.1.1.5 Teledyne DR-U3-50Y2C-C1-CS .....	19
3.1.1.6 CMOS Camera Product Selection.....	19
3.1.2 Lens Technology.....	21
3.1.2.1 Singlet Lenses .....	21
3.1.2.2 Achromatic Doublet .....	21
3.1.2.3 Apochromatic Lenses .....	21
3.1.2.4 Lens Technology Selection.....	21
3.1.3 Illumination Light Source Technology.....	22
3.1.3.1 Halogen Lamps .....	22
3.1.3.2 Arc Lamps .....	23
3.1.3.3 Light-Emitting Diodes (LEDs).....	23
3.1.3.4 Lasers .....	23
3.1.3.5 Illumination Light Source Technology Selection.....	24
3.1.3.6 LED Product Comparison .....	24
3.1.3.7 Green 10W High Power LED by Switch Electronics (311448).....	24

3.1.3.8	10 Watt Green LED by Vetco Electronics (VUPN1856)	25
3.1.3.9	High Power LED Chip 10W Green by CHANZON	25
3.1.3.10	LED Product Selection	25
3.1.4	Collimating Lens Technology	26
3.1.4.1	Aspheric Lenses	26
3.1.4.2	Best Form Spherical Lenses	26
3.1.4.3	Plano-Convex Lenses	27
3.1.4.4	Collimating Lens Technology Selection	27
3.1.4.5	LBF254-040	27
3.1.4.6	LBF254-040-A	27
3.1.4.7	LBF254-040-B	28
3.1.4.8	Best Form Spherical Lens Product Selection	28
3.1.5	Optical Mirror Technology	28
3.1.5.1	Front-Surface Plano Metallic Mirrors	28
3.1.5.2	Front-Surface Plano Dielectric Mirrors	29
3.1.5.3	Back-Surface Plano Mirrors	29
3.1.5.4	Right-Angle Prism Mirrors	29
3.1.5.5	Optical Mirror Technology Selection	29
3.1.5.6	Plano Broadband Dielectric Mirrors Product Comparison	30
3.1.5.7	Fused Silica Broadband Dielectric Mirrors (BB1-E02)	30
3.1.5.8	Zerodur® Broadband Dielectric Mirrors (BB111-E02)	30
3.1.5.9	Plano Broadband Dielectric Mirrors Product Selection	30
3.1.6	Optical Beamsplitter Technology	31
3.1.6.1	Plate Beamsplitters	31
3.1.6.2	Cube Beamsplitters	31
3.1.6.3	Pellicle Beamsplitters	32
3.1.6.4	Optical Beamsplitters Technology Selection	32
3.1.6.5	Plate Beamsplitter Product Comparison	33
3.1.6.6	Plate Beamsplitter, 25.4mm 50R/50T (BSW10)	33
3.1.6.7	Plate Beamsplitter, 45:55 R:T (BSW41-532)	33
3.1.6.8	Beamsplitter Product Selection	33

3.1.7	Opaque Surface.....	33
3.1.7.1	Stainless Steel.....	34
3.1.7.2	Aluminum.....	34
3.1.7.3	Silicon Wafer.....	35
3.1.7.4	Copper.....	35
3.1.8	Particle Selection.....	37
3.1.8.1	Cotton Fiber Size.....	37
3.1.8.2	Human Hair.....	37
3.1.8.3	Cellulose Fiber.....	37
3.1.9	Camera Adjuster Technology.....	38
3.1.9.1	Manual Micrometer Screw.....	39
3.1.9.2	Grashof Crank Rocker or Regular Crank Rocker.....	39
3.1.9.3	Series of Gears and a Rack.....	41
3.1.9.4	Rail and slider.....	42
3.1.9.5	Camera Adjuster Technology Comparison.....	43
3.1.10	Optical Mounting Methods.....	44
3.1.10.1	Integrated Mounting Design.....	44
3.1.10.2	Commercial Optical Mounts.....	45
3.1.11	Filament Types.....	45
3.1.11.1	PLA CF (Carbon Fiber Reinforced PLA).....	45
3.1.11.2	PLA (Polylactic Acid).....	45
3.1.11.3	ABS (Acrylonitrile Butadiene Styrene).....	45
3.1.11.4	TPU (Thermoplastic Polyurethane).....	46
3.1.11.5	Filament Type Selection.....	46
3.1.12	Processor Technology.....	46
3.1.12.1	Microcontroller Unit (MCU).....	47
3.1.12.2	Single Board Computer (SBC).....	47
3.1.12.3	System-on-Module (SoM) / Compute Module.....	48
3.1.12.4	Processor Technology Selection.....	48
3.1.12.5	SoM Product Comparison.....	49
3.1.12.6	ArmSoM CM5.....	49

3.1.12.7 Raspberry Pi Compute Module 5 (CM5) .....	49
3.1.12.8 LattePanda Mu Compute Module.....	50
3.1.12.9 SoM Product Selection.....	50
3.1.13 I/O Carrier Board .....	51
3.1.13.1 (DFR1142) Lite Carrier Board .....	51
3.1.13.2 (DFR1141) Full-Function Evaluation Carrier Board .....	51
3.1.13.3 Carrier Board Product Selection .....	52
3.1.14 Touch Screen Display Selection .....	52
3.1.14.1 Panel Size Selections.....	52
3.1.14.2 Display Technology Selection.....	53
3.1.14.3 LED Display.....	53
3.1.14.4 LCD Display .....	53
3.1.14.5 OLED Display.....	53
3.1.14.6 Touch Input Technologies .....	54
3.1.14.7 7” IPS Touch Screen Panel 1024x600 HDMI.....	55
3.1.14.8 Waveshare 7” HDMI LCD IPS 1024x600 Touch Screen.....	55
3.1.14.9 7” HDMI/USB Capacitive Touch Screen Monitor for Raspberry Pi / PC.....	55
3.1.14.10 Touch Screen Display Part Selection .....	55
3.1.15 PSU .....	56
3.1.15.1 Battery Technology .....	56
3.1.15.2 Lithium-ion.....	57
3.1.15.3 Nickel-Metal Hydride (NiMH) .....	57
3.1.15.4 Sealed Lead Acid (SLA) .....	57
3.1.15.5 Battery Technology Selection.....	58
3.1.15.6 Battery Product Comparison .....	58
3.1.15.7 Panasonic Eneloop Pro (BK-3HCCA) .....	58
3.1.15.8 Amazon Basics High-Capacity NiMH AA.....	58
3.1.15.9 Energizer Recharge Power Plus AA.....	59
3.1.15.10 Battery Product Selection.....	59
3.1.15.11 Constant Current Driver Technologies .....	59

3.1.15.12	LD24AJTA Constant Current LED Driver.....	60
3.1.15.13	Chanzon LED Driver 900mA (Constant Current Output) .....	60
3.1.15.14	B0F1K97V61 Constant Current Led Driver .....	60
3.1.15.15	Constant Current Driver Product Selection.....	60
3.1.15.16	LED Microcontroller.....	60
3.1.15.17	Arduino Nano ESP32 [ABX00083] .....	61
3.1.15.18	MCP2221A USB to GPIO.....	61
3.1.15.19	Arduino Uno REV3 [A000066] .....	61
3.1.15.20	LED Microcontroller Product Selection .....	61
3.2	Communication Protocol Selection .....	62
3.2.1	UART.....	62
3.2.2	SPI.....	62
3.2.3	Communication Protocol Selection .....	63
3.3	Software Selection .....	63
3.3.1	Data Analysis Software.....	63
3.3.1.1	Python.....	63
3.3.1.2	MATLAB .....	63
3.3.1.3	Microsoft Excel .....	64
3.3.1.4	Data Analysis Software Selection .....	64
3.3.2	Optical Design and Simulation Software.....	65
3.3.2.1	Ansys Zemax OpticStudio.....	65
3.3.2.2	Code V.....	65
3.3.2.3	3DOptix.....	66
3.3.2.4	Optical Design and Simulation Software Selection .....	66
3.3.3	Optical Schematic Software.....	66
3.3.3.1	Microsoft PowerPoint .....	67
3.3.3.2	Inkscape.....	67
3.3.3.3	Optical Schematic Software Selection .....	67
3.3.4	CAD/3D Modeling Design Software.....	68
3.3.4.1	Fusion360.....	68
3.3.4.2	SolidWorks .....	68
3.3.4.3	AutoCAD.....	69

3.3.4.4 CAD Software Selection .....	69
3.3.5 3D-Printing Software .....	69
3.3.5.1 Ultimaker Cura .....	69
3.3.5.2 Slic3r .....	70
3.3.5.3 PrusaSlicer.....	70
3.3.5.4 3D-Printing Software Selection .....	70
3.3.6 Embedded Programming Language.....	71
3.3.6.1 C .....	71
3.3.6.2 C++.....	72
3.3.6.3 Python.....	72
3.3.7 Development Environment .....	73
3.3.7.1 Eclipse .....	73
3.3.7.2 Visual Studio Code (VS Code).....	74
<b>Chapter 4 – Standards and Design Constraints .....</b>	<b>75</b>
4.1 Industry Standards .....	75
4.1.1 I2C Standards.....	75
4.1.2 Battery and Power Standards .....	75
4.2 PCB Design Standards.....	76
4.2.1.1 Manufacturer’s Design Standards .....	76
4.2.1.2 Industry’s Design Standards.....	76
4.3 Mechanical Hardware standards .....	77
4.4 Design Constraints.....	78
4.4.1 Initial System Design Constraints.....	78
4.4.2 Current Iteration Design Constraints .....	79
4.5 Size Constraints .....	81
4.5.1.1 Lens Diameter .....	81
4.5.1.2 Focal Length Limitations .....	81
4.5.2 Two 4f System Design Constraints.....	82
4.6 Time Constraints.....	83
4.7 Economic Constraints .....	84
<b>Chapter 5 – Application of ChatGPT.....</b>	<b>84</b>
5.1 Case Study 1 .....	85
5.2 Case Study 2 .....	86

5.3	Case Study 3 .....	87
5.4	Case Study 4 .....	87
<b>Chapter 6 – Hardware Design.....</b>		<b>88</b>
6.1	CAD Model.....	88
6.1.1	Case – Current Design .....	88
6.1.2	Periscope Unit.....	93
6.1.3	Camera Unit.....	95
6.1.4	Annular Stop .....	100
6.2	Optical Illumination Sub-System.....	100
6.2.1	LED Optical Power.....	101
6.2.2	Annular Stop Size .....	101
6.2.3	Illumination Lenses Optical Path and Magnification .....	102
6.3	Optical Detection Sub-System.....	104
6.3.1	Detection Target Magnification.....	104
6.3.2	Detection Lenses Focal lengths .....	104
6.3.3	Numerical Aperture .....	105
6.3.4	Rayleigh Criterion.....	106
6.4	Board to Board Electrical Diagram.....	107
<b>Chapter 7 – Software Design.....</b>		<b>108</b>
7.1	Software Overview .....	108
7.2	User Interface.....	108
7.3	Image Processing Algorithm.....	110
7.4	Particle Inspector User Interface.....	111
<b>Chapter 8 – System Fabrication .....</b>		<b>114</b>
8.1	Mechanical Fabrication.....	114
8.2	3D Printer.....	114
8.3	3D Printing Setting and Parameters .....	115
8.4	Optical 3D Printed Optical Mounts .....	116
<b>Chapter 9 – System Testing and Evaluation.....</b>		<b>117</b>
9.1	Zemax Simulations .....	117
9.2	Optoelectronics Feasibility .....	119
9.2.1	Annular Stop Testing .....	119
9.2.2	LED Power Efficiency Testing.....	121
9.2.3	Detection Subsystem Testing.....	122

<b>Chapter 10 Administrative Content .....</b>	<b>122</b>
10.1 Budget and Finances .....	122
10.2 Project Milestones.....	124
10.3 Table of Work Distribution.....	126
<b>Chapter 11 – Conclusion.....</b>	<b>127</b>
11.1 Optical Challenges .....	127
11.2 Mechanical Challenges .....	127
11.3 Future Improvements .....	128
<b>Chapter 12 Appendices .....</b>	<b>I</b>
12.1 Appendix A – References .....	I
12.2 Appendix B LLMs .....	IV

# LIST OF TABLES

TABLE 2.5.1 OVERALL SYSTEM SPECIFICATIONS .....	6
TABLE 2.5.2 COMPONENT SPECIFICATIONS.....	7
TABLE 3.1.1 CAMERA SENSOR TYPE SELECTION .....	15
TABLE 3.1.2 CMOS CAMERA PART SELECTION .....	19
TABLE 3.1.3 LENS TYPE COMPARISONS .....	21
TABLE 3.1.4 ILLUMINATION LIGHT SOURCE TECHNOLOGY COMPARISON .....	24
TABLE 3.1.5 LED PRODUCT SELECTION .....	25
TABLE 3.1.6 COLLIMATING LENS TECHNOLOGY SELECTION .....	27
TABLE 3.1.7 BEST FORM SPHERICAL LENS PART SELECTION .....	28
TABLE 3.1.8 MIRRORS TECHNOLOGY SELECTION .....	29
TABLE 3.1.9 MIRRORS PRODUCT SELECTION.....	30
TABLE 3.1.10 BEAMSPLITTER TECHNOLOGY SELECTION.....	32
TABLE 3.1.11 BEAMSPLITTER PRODUCT SELECTION.....	33
TABLE 3.1.12 OPAQUE SURFACE SELECTION .....	36
TABLE 3.1.13 PARTICLE SELECTION .....	38
TABLE 3.1.14 CAMERA ADJUSTER TECHNOLOGY COMPARISON .....	43
TABLE 3.1.15 FILAMENT PROPERTIES .....	46
TABLE 3.1.16 PROCESSOR TECHNOLOGY COMPARISONS .....	48
TABLE 3.1.17 SoM PRODUCTS COMPARISONS.....	50
TABLE 3.1.18 CARRIER BOARD PRODUCTS COMPARISON.....	52
TABLE 3.1.19 SCREEN SIZE SELECTION .....	52
TABLE 3.1.20 DISPLAY TECHNOLOGY SELECTION .....	53
TABLE 3.1.21 TOUCH INPUT TECHNOLOGIES COMPARISON.....	54
TABLE 3.1.22 TOUCH SCREEN PRODUCT COMPARISONS.....	55
TABLE 3.1.23 POWER SUPPLY REQUIREMENTS.....	56
TABLE 3.1.24 BATTERY TECHNOLOGY SELECTION.....	58
TABLE 3.1.25 BATTERY PRODUCT COMPARISONS .....	59
TABLE 3.1.26 CONSTANT CURRENT DRIVER PRODUCT COMPARISONS.....	60
TABLE 3.1.27 MICROCONTROLLER PRODUCT COMPARISON .....	61
TABLE 3.2.1 COMMUNICATION PROTOCOL SELECTION.....	63
TABLE 3.3.1 DATA ANALYSIS SOFTWARE COMPARISONS.....	64
TABLE 3.3.2 OPTICAL DESIGN AND SIMULATION SOFTWARE COMPARISONS.....	66
TABLE 3.3.3 OPTICAL SCHEMATIC SOFTWARE SELECTION .....	67
TABLE 3.3.4 CAD SOFTWARE COMPARISON.....	69
TABLE 3.3.5 3D PRINTING SOFTWARE COMPARISON .....	70
TABLE 3.3.6 EMBEDDED LANGUAGE SELECTION.....	72
TABLE 3.3.7 DEVELOPMENT ENVIRONMENT SELECTION .....	74
TABLE 10.1.1 BILL OF MATERIALS .....	122
TABLE 10.2.1 SENIOR DESIGN I PROJECT MILESTONES .....	124
TABLE 10.2.2 SENIOR DESIGN II PROJECT MILESTONES .....	126
TABLE 10.3.1 WORK DISTRIBUTION .....	126

# List of Figures

FIGURE 2.4.1 OPTICAL DESIGN SCHEMATIC.....	5
FIGURE 2.6.1 HARDWARE BLOCK DIAGRAM.....	9
FIGURE 2.6.2 IMAGE PROCESSING SOFTWARE BLOCK DIAGRAM.....	9
FIGURE 2.6.3 USER INTERFACE SOFTWARE BLOCK DIAGRAM .....	10
FIGURE 2.6.4 CAMERA UNIT TOP (MM).....	10
FIGURE 2.6.5 CAMERA UNIT BOTTOM (MM).....	11
FIGURE 2.6.6 PERISCOPE UNIT ISOMETRIC VIEW (MM) .....	12
FIGURE 2.6.1 HOUSE OF QUALITY DIAGRAM .....	13
FIGURE 3.1.1 CRANK ROCKER COMPRESSED .....	40
FIGURE 3.1.2 CRANK ROCKER EXPANDED .....	41
FIGURE 3.1.3 GEAR AND RACK .....	42
FIGURE 3.1.4 RAIL AND SLIDER .....	43
FIGURE 4.4.1 INPUT RAYS PARALLEL .....	80
FIGURE 4.4.2 INPUT RAYS COME FROM FOCUS.....	80
FIGURE 6.1.1 CASE CLOSED .....	89
FIGURE 6.1.2 CASE EXPANDED .....	90
FIGURE 6.1.3 CASE LID DRAWING (MM) .....	91
FIGURE 6.1.4 MAIN CASE DRAWING (MM).....	92
FIGURE 6.1.5 PERISCOPE DRAWING (MM) .....	93
FIGURE 6.1.6 PERISCOPE UNIT LID DRAWING (MM).....	94
FIGURE 6.1.7 CAMERA UNIT TOP AND BOTTOM VIEW DRAWLING (MM).....	95
FIGURE 6.1.8 CAMERA UNIT ISOMETRIC VIEW (MM) .....	96
FIGURE 6.1.9 CAMERA ILLUMINATION SECTION OPTICAL MEASUREMENTS (MM).....	97
FIGURE 6.1.10 CAMERA UNIT DETECTION SECTION OPTICAL MEASUREMENTS (MM) .....	97
FIGURE 6.1.11 CAMERA UNIT BOTTOM LID (MM).....	98
FIGURE 6.1.12 CAMERA UNIT TOP LID (MM) .....	99
FIGURE 6.1.13 ANNULAR STOP (MM) .....	100
FIGURE 6.2.1 ILLUMINATION SYSTEM RAY TRACE .....	101
FIGURE 6.3.1 RAY TRACING OF THE DETECTION PATH .....	106
FIGURE 6.4.1 BOARD TO BOARD ELECTRICAL DIAGRAM.....	107
FIGURE 7.2.1 USE CASE DIAGRAM.....	108
FIGURE 7.2.2 ACTIVITY DIAGRAM.....	109
FIGURE 7.3.1 IMAGE PROCESSING ALGORITHM BLOCK DIAGRAM.....	110
FIGURE 7.4.1 TOUCH SCREEN HUB HOME INTERFACE .....	111
FIGURE 7.4.2 REFERENCE SELECTION INTERFACE.....	112
FIGURE 7.4.3 CAPTURE SCREEN INTERFACE .....	113
FIGURE 7.4.4 TEST HISTORY INTERFACE.....	114
FIGURE 8.4.1 OPTICAL MOUNT FRONT (MM) .....	116
FIGURE 8.4.2 OPTICAL MOUNT BACK (MM).....	117
FIGURE 9.1.1 ILLUMINATION SEQUENTIAL ZEMAX SIMULATION .....	118
FIGURE 9.1.2 DETECTION SEQUENTIAL ZEMAX SIMULATION .....	118
FIGURE 9.1.3 NON-SEQUENTIAL ZEMAX SIMULATION.....	119
FIGURE 9.2.1 ANNULAR STOP TEST SETUP .....	119
FIGURE 9.2.2 ANNULAR STOP TEST SETUP .....	120
FIGURE 9.2.3 ANNULAR MASK BEFORE 1ST LENS OF 2ND 4F SYSTEM OF THE ILLUMINATION SUB-SYSTEM ..	120
FIGURE 9.2.4 ANNULAR MASK AFTER 1ST LENS OF 2ND 4F SYSTEM OF THE ILLUMINATION SUB-SYSTEM ...	120
FIGURE 9.2.5 ANNULAR STOP INNER-DIAMETER VS ILLUMINATED SPOT SIZE .....	121
FIGURE 9.2.6 DETECTION HALF - ANGLE VS ANNULAR STOP - INNER DIAMETER.....	121
FIGURE 9.2.7 POWER EFFICIENCY TEST SETUP .....	121
FIGURE 9.2.8 OPTICAL POWER THROUGH ILLUMINATION SUB-SYSTEM .....	121
FIGURE 9.2.9 DETECTION SUBSYSTEM ON AN OPTICAL RAIL.....	122

# Chapter 1 – Executive Summary

Modern semiconductor manufacturing requires very precise fabrication with machines that ensure high product yield in very controlled environments. Modern semiconductor chips have electrical components way smaller than the wavelengths of light used in their production. This requires extreme caution of contamination in the fabrication machines due to the precision needed. ASML's EUV (extreme ultraviolet) and DUV (deep ultraviolet) lithography machines can produce 2 nm logic nodes.

The Particle Inspector via Darkfield Illumination (P.I.D.I) project main goal is to ensure cleanliness of components that are used in ASML's lithography machines. It is designed to detect contamination in less than 2 seconds per spot size needed for component. The darkfield illumination is essential to increasing the contrast of clear or opaque contaminates, such as dust or fibers left from cleaning wipes. The P.I.D.I is designed to detect particles that are 50  $\mu\text{m}$  – 300  $\mu\text{m}$  in height and 50  $\mu\text{m}$  – 300  $\mu\text{m}$  in width. The P.I.D.I is composed of three elements, the camera module, the periscope unit and the electrical unit.

*Camera Module:* The camera module incorporates the illumination source and the camera detection optical paths that are split via a beamsplitter. Both optical paths are extended with dielectric mirrors to ensure proper demagnification and light rays not miss the working diameters of other lenses in the system. The illumination source and the camera are connected to the electrical unit with IO interfaces. The camera data will be used in the contamination detection algorithm, and the illumination source connection will provide electrical power. The camera module has dimensional constraints that are provided by ASML.

*Periscope Unit:* The periscope unit is designed to be a modular piece of optics that bends the optical paths of the detection and illumination twice to connect the camera module to the opaque surface. It is designed to be a sponsor specified distance away from the camera module and the opaque surface. This unit is modular to allow for future integration of different magnification levels that could be needed for other particle sizes. The periscope unit has dimensional constraints that are provided by ASML.

*Electrical Unit:* The electrical unit holds all the electrical components of the project that perform the contamination inspection algorithm and power the optoelectronics. The electrical unit does not have dimensional constraints since a small PCB would require small electronic components which is not ideal for prototype development.

The system will utilize the three different elements to ensure cleanliness of components via an automated algorithm that identifies the contaminants on surfaces from comparing the taken image with a reference image.

This document is a record of P.I.D.I's design process. This includes the theories and technologies used for the design, part selection research, software selection research, project constraints, engineering standards, system testing and planning to achieve a successful prototype. It also contains all of the systems electrical, mechanical, optical and software schematics with detailed explanation of the components and reasoning behind the design. The final section concludes with potential future improvements and

recommendations. The appendix's that contain references and copyright permissions can be found at the end of the document.

## **Chapter 2 - Project Description**

### **2.1 Project Background and Motivation**

The ability to quickly inspect small particles plays an important role in many manufacturing disciplines, including semiconductor fabrication, automotive production, and quality control for 3D-printed components. This technology is also invaluable in medical research, where it can enable faster surface cleanliness checks and a cleaner view of low contrast samples.

Brightfield microscopy, while widely used and simpler, has difficulties with resolving very small particles and low contrast. The solution to this is darkfield microscopy. In a brightfield microscope, the sample is hit with direct light illuminating the whole sample and the background resulting in a bright image that can hide finer details. A darkfield microscopy is similar but setup in such a way that the light that goes through the sample does not enter the eyepiece and is not collected. The only light a dark-field microscope collects is the small amount of light scattered off the particles on your sample. This results in a dark background with lights coming off the edges of features in your sample. This increases contrast and allows you to view samples that were previously very difficult to see in brightfield.

Darkfield microscopy is an effective tool for imaging, specifically live imaging, as well as discovering contaminants. However, these tools are bulky and expensive, which limits their usability. Our project aims to create a more space efficient, less expensive, and modular darkfield imaging system capable of quickly processing the presence of contaminants on a surface.

### **2.2 Existing Products**

Darkfield illumination imaging systems are mainly used in two industries, biomedical and component manufacturing. In the biomedical industry, scientists use darkfield illumination microscopy for detecting and observing specimens that are transparent or have low absorption properties, which would cause them to be invisible to detect using a brightfield microscope. The imaging technique also allows scientists to observe live specimens without staining or using dyes, which preserves the natural physiology of the specimen. This is due to the high contrast images that are formed using darkfield microscopy. In the component manufacturing industry, companies use darkfield microscopy as part of their quality assurance process to perform visual inspection or to measure small components. This is a very similar application of the imaging technique used in our project.

The MC-70 Trinocular Reflected Light Brightfield/Darkfield Tool Makers/Measuring Microscope made by Meiji Techno is an example product that is designed to use darkfield microscopy for component or product quality assurance. This microscope has other features such as brightfield microscopy and polarization switching modes which increase the price of the product substantially. Since these systems are mainly

sold to companies, the official price is not listed on Meiji Techno's website, rather a "get a quote" button to allow companies to customize the system to best fit their application. A used system costs \$13,000. This is the biggest negative of buying this system, it costs more than double of our budget while customers may buy it only for the darkfield microscopy. Of course it has extra features, but if a customer would just want darkfield illumination to scan for contaminants, our project design is superior due to the price difference. This product inspired our camera capturing system, with a screen to show the contaminants.

Our system layout is closest to Olympus BX60M Reflected Light Brightfield & Darkfield Microscope, which can be used to inspect surface features in manufacturing applications or cellular structures in the biomedical application. When this microscope is in darkfield mode, the light rays from the source get collimated and go through a condenser aperture, field diaphragm and an opaque stop to create the hollow ring of light that is necessary for darkfield illumination. This is where the systems begin to differ in design yet carry the same principles. In this Olympus microscope, this ring then bounces off a 45-degree ring mirror into the reflective objective lens down to the objective surface. The scattered light then goes into the objective lens and through the hole of the ring mirror into a prism then into the eyepiece lens or a tube lens if a camera is attached. Since our system is built with off-the-shelf components, we use disc mirrors in a periscope to design accurate optical paths for our illuminated light and reflected image. This design inspired our stretch goal of being able to switch between darkfield and brightfield illumination depending on the particles the inspector is supposed to detect. They use a slider that lets the user switch between illumination modes by adding or removing the annular stop from the optical path. We will try to implement this feature in our system since our design is very similar to this microscope.

## 2.3 Project Objectives and Goals

### Basic Goals:

- Build and design an optical module to inspect particles of  $50\ \mu\text{m}$  –  $300\ \mu\text{m}$  in height and  $50\ \mu\text{m}$  –  $300\ \mu\text{m}$  in width on an opaque surface using darkfield illumination.
- Image the opaque surface and determine if the opaque surface contains contaminants.
- Ensure the optical module is simple to use and can be operated by anyone.
- Design a durable housing that self-aligns during assembly and maintains optical alignment during handling.
- Ensure the device is easy to hand-carry.

### Advanced Goals:

- Control the system using a small touch screen and display the image and results with an LCD screen.
- Design the device to be battery operated.
- Stabilize LED and electronics temperatures during extended operation.

### Stretch Goals:

- Accurately record the frequency and types of contaminants found on surface.

- Add wireless data transfer (Bluetooth or Wi-Fi) to easily send the results to other devices
- Add internal data storage to select from a variety of reference images
- Enable precise, repeatable fine-focus adjustments.
- Switch between brightfield and darkfield illumination
- Enable a particle identification system, to pinpoint the location of discovered particles.

Basic Objectives:

- Design a magnification system capable of viewing particles of 50  $\mu\text{m}$  – 300  $\mu\text{m}$  in height and 50  $\mu\text{m}$  – 300  $\mu\text{m}$  in width.
- Design darkfield illumination system.
- Determine the appropriate LEDs and camera.
- Develop an algorithm to identify the presence of contaminants in less than a second.
- Design a case to house and align all components ensuring that minimal optical alignment is required for the user.
- Designed using mostly off-the-shelf components.
- Complete module assembly or disassembly in under 5 minutes, while preserving optical alignment.
- Keep total device mass  $\leq 10$  kg.

Advanced Objectives:

- Implement an on-device UI that fits the size constraints.
- Utilize a common illumination and detection path to make the module spatially efficient and simpler to use.
- Implement a system that allows us to quickly log whether our algorithm reported a false positive.
- Integrate passive thermal management to limit internal temperature rise to  $\leq 5$   $^{\circ}\text{C}$  above ambient over 30 minutes of continuous LED & camera use.

Stretch Objectives:

- Design an algorithm to automatically adjust the brightness of LED and use image processing techniques to detect hard to observe contaminants.
- Design an algorithm that will automatically store the results and log the types and frequency of contaminants found on objective.
- Determine the device and storage space needed to store the reference images efficiently.
- Add a manual focus knob that shifts the sample stage in 1 mm increments.
- Add a focus knob that adjusts the camera within a 8mm range.
- Design modular system to switch between brightfield and darkfield illumination.

## 2.4 Project Features and Functionalities

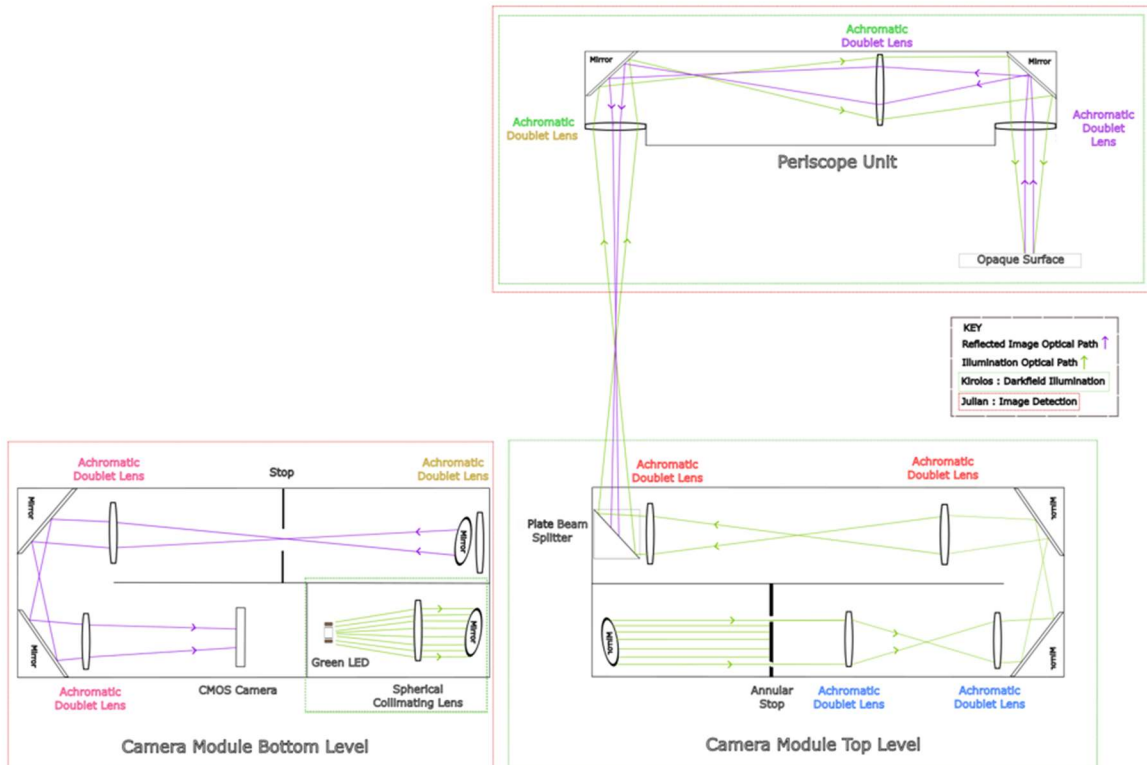


Figure 2.4.1 Optical Design Schematic

Through the guidance of our sponsor, our particle inspector will possess several features and functionalities. Additionally, through our own ambitions there will be extra features and functionalities that will be implemented. In essence, our particle inspection system is designed to use darkfield microscopy to image small particles on an opaque surface while remaining small, portable, easy to use, and more cost effective than standard darkfield microscopy setups

A darkfield illumination and detection system will be used to create a hollow cone of light and illuminate the target, collecting only the light scattered off the target. We will use an LED (to provide light), an annulus ring (to create a hollow cone), and a reflective objective lens to capture that scattered light. The resultant image captured by the camera will be processed by an algorithm to improve contrast and clarity.

A two-lens magnification system will be used to magnify the sample and image particles  $50\ \mu\text{m} - 300\ \mu\text{m}$  in size. In addition to magnification, either small motors or image processing techniques will be used to automatically find the best focus and brightness to produce the clearest image.

The module is designed to be portable, with the camera and illumination unit fitting within our sponsor's dimensional specifications.

The mechanical framework will be engineered to maintain precise optical alignment throughout operation, with an additively manufactured chassis incorporating vibration

damping mounts to isolate the periscope and camera modules from environmental disturbances. The object stand will incorporate an adjustable kinematic mount to enable tuning of the sample position within the optical path, ensuring optimal focus and illumination. Thermal effects will be minimized through material selection, ventilation, and location of heat generating components. This mechanically stable platform is essential to preserving the darkfield illumination geometry and ensuring consistent particle detection accuracy.

A fast image processing software will be integrated into the device. This software will analyze and compare the image captured by the camera to a reference image. This algorithm is expected to run in under a second, thus providing fast and effective results. In order to visualize the results as well as establishing concise communication of the results, there will be a green LED and a red LED. These will be used to indicate clearly whether or not contamination was found to be present. If the image processing algorithm finds there to be foreign particles in the object, the red LED will light up. Otherwise, if everything is determined to be in order then the green LED will light up.

Included in the particle inspector there will be an LCD display. This display will show the images captured by the camera. Given that the object will not be visible, the screen will offer a way to visualize the particles, as well as checking that the results given were not false positives.

Another feature included in the particle inspector will be a simple control panel. This panel will consist of a few buttons and switches that will allow for control of the device. The idea is for the user to be able to control the LED as well as being able to capture either a reference image or an image to be analyzed. Furthermore, by being able to turn the LED off despite the device itself being on we are able to reduce power consumption and increase battery life.

## 2.5 Project Specification

In consideration of the required objectives, the following specifications are denoted:

**Table 2.5.1 Overall System Specifications**

Parameter	Description	Target Value and Unit(s)
Image Processing Time	Amount of time image processing takes to compute difference between tests	< 2 seconds
Magnification	Required magnification to detect for a size between 50 $\mu\text{m}$ – 300 $\mu\text{m}$	0.20x – 0.5x magnification
False Positive Rate	Image processing accuracy	< 0.1%
Image Capture Response Time	Time it takes to capture image	<1 second

Dimensions	Meet dimensions and spacing requirements for camera module, periscope unit and opaque surface	<ul style="list-style-type: none"> <li>- Camera module: Sponsor provided dimensions</li> <li>- Periscope unit: Sponsor provided dimensions</li> <li>- Distance between camera module and periscope unit: Sponsor provided dimensions</li> <li>- Distance between periscope unit and opaque surface: Sponsor provided dimensions</li> </ul> <p>(We can not share these dimensions due to an NDA)</p>
Weight	It should be easy to move around and transport.	$\leq 2.5$ kg
Battery life	Should last long enough to inspect at least one component	$> 30$ minutes

**Table 2.5.2 Component Specifications**

Components	Parameter	Subsystem and Description	Specification
Camera	Resolution	Optical Detection - The camera sensor must have enough resolution to provide a high-quality image of the opaque surface.	1 MP – 5 MP CMOS Sensor
LED	Power and Wavelength	Optical Illumination - The LED must be bright enough to perform darkfield illumination on	Power : 5-15 Watts Input Current : 900 mA

		the opaque surface. Green wavelength is ideal to limit chromatic aberrations in our image.	Input Voltage : 9V – 11V Center Wavelength : 500 nm -560 nm range
Annular stop	Inner / Outer Radius	Optical Illumination - The annular stop must create a ring of light that is necessary to perform darkfield illumination on the opaque surface.	Inner radius: 15 mm – 25 mm Outer radius: 20 mm – 30 mm
Achromatic Doublets	Focal Length and Diameter	Optical – The achromatic doublets mask the annular stop through the illumination system to the condenser lens via three 4f systems. The image is reflected through three different 4f systems to the camera. The spacing and focal lengths of the lenses enable magnification and demagnification.	Achromatic Doublet Lenses:  Lens 1 – 10 Focal Length must be a standard size:  10 mm – 90 mm  Diameter : 10 mm – 50 mm
Pellicle Beam splitter	Reflection : Transmission Ratio and Wavelength Performance	Optical - The beam splitter must reflect illuminated beam into periscope unit and transmit the reflected image to the set of achromatic doublets in the detection optical path.	45:55 (R:T)  Center Wavelength : 500 nm – 560 nm
Flat Mirrors	Diameter	Optical - The flat mirrors must reflect the illuminated ring from camera module to the reflective objective lens and the reflected image back to the camera module.	10 mm – 50 mm
Camera Module Mechanical Housing	Dimensions	Mechanical - The camera module must fit all optical and electronic components in the dimensions provided by ASML.	Sponsor provided

Periscope Mechanical Housing	Dimensions	Mechanical - The periscope unit must fit all optical components in the dimensions provided by ASML.	Sponsor provided
------------------------------	------------	---	------------------

## 2.6 Block Diagrams

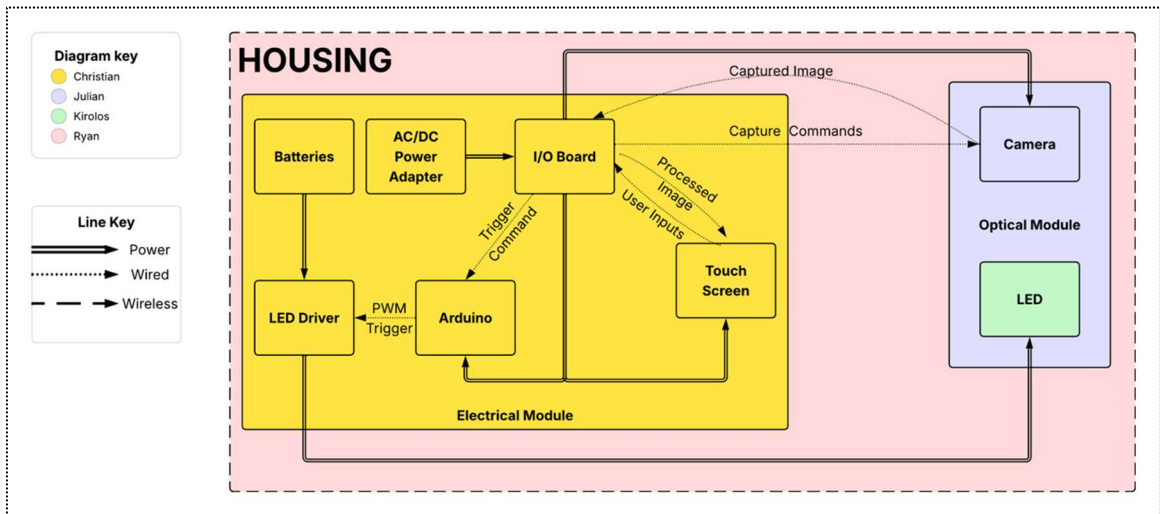


Figure 2.6.1 Hardware Block Diagram

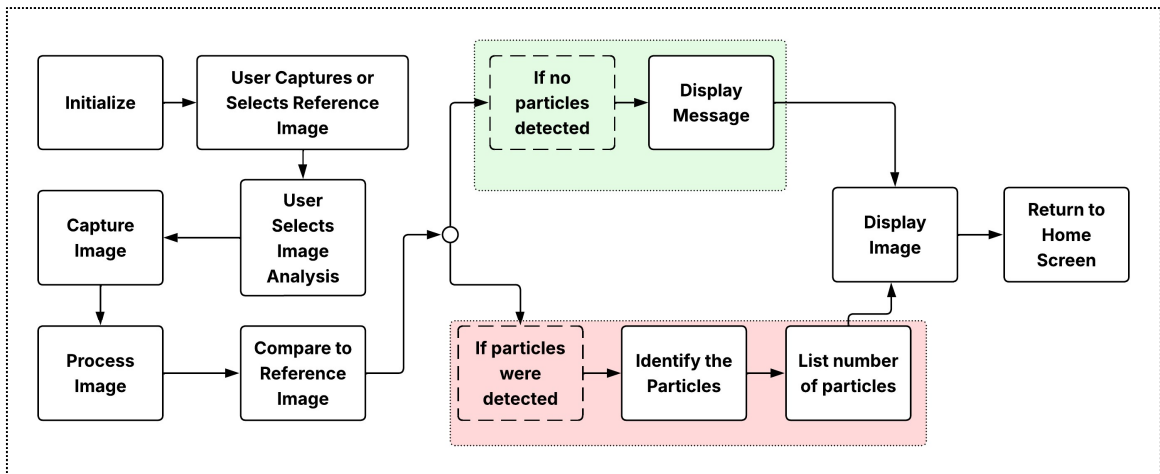


Figure 2.6.2 Image Processing Software Block Diagram

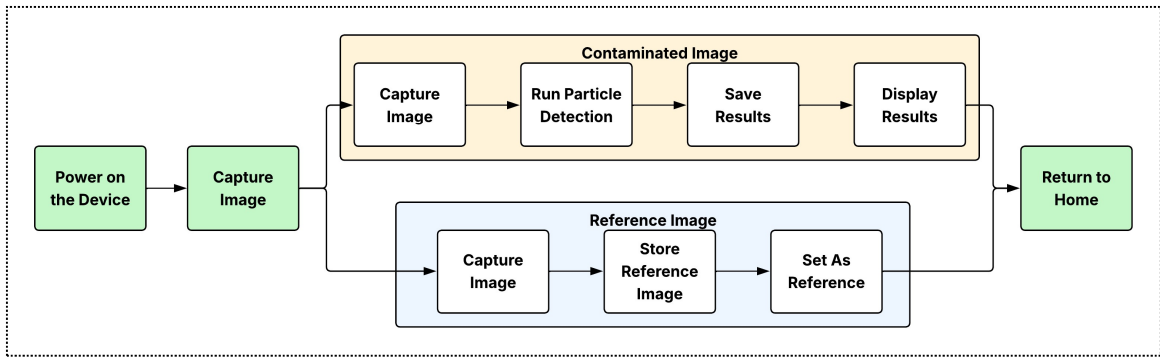


Figure 2.6.3 User Interface Software Block Diagram

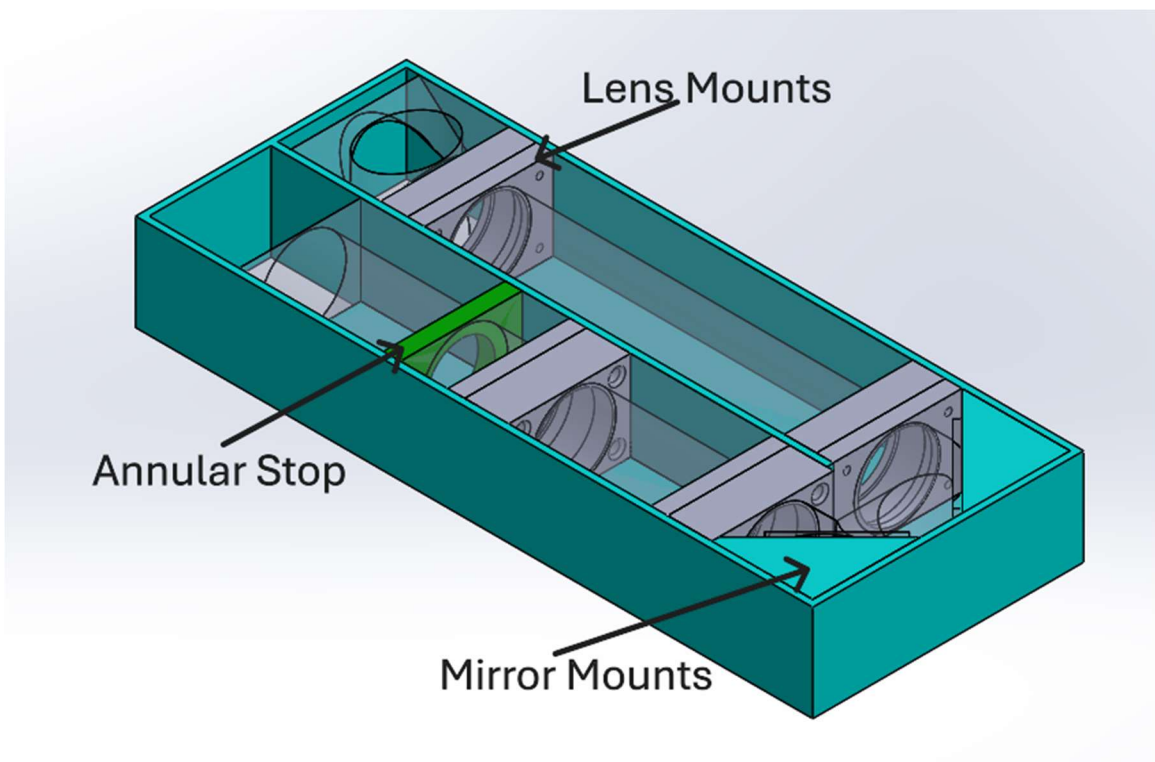
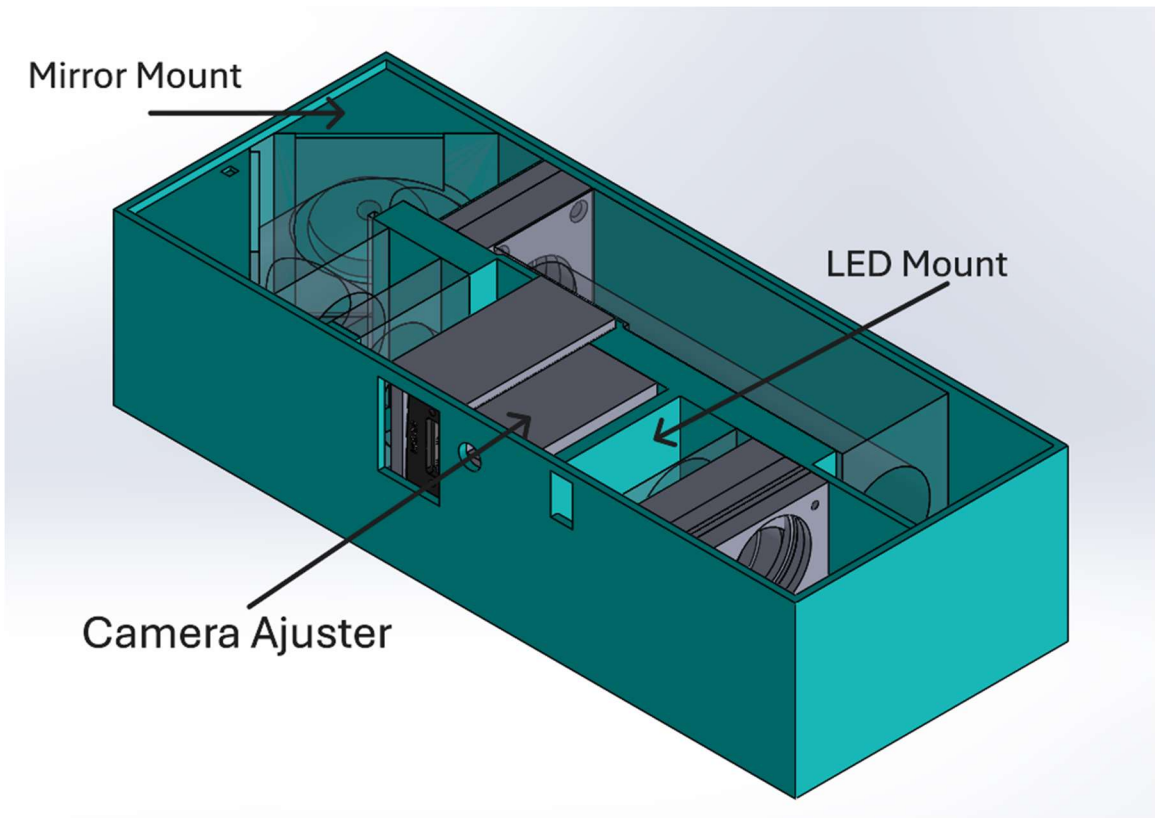


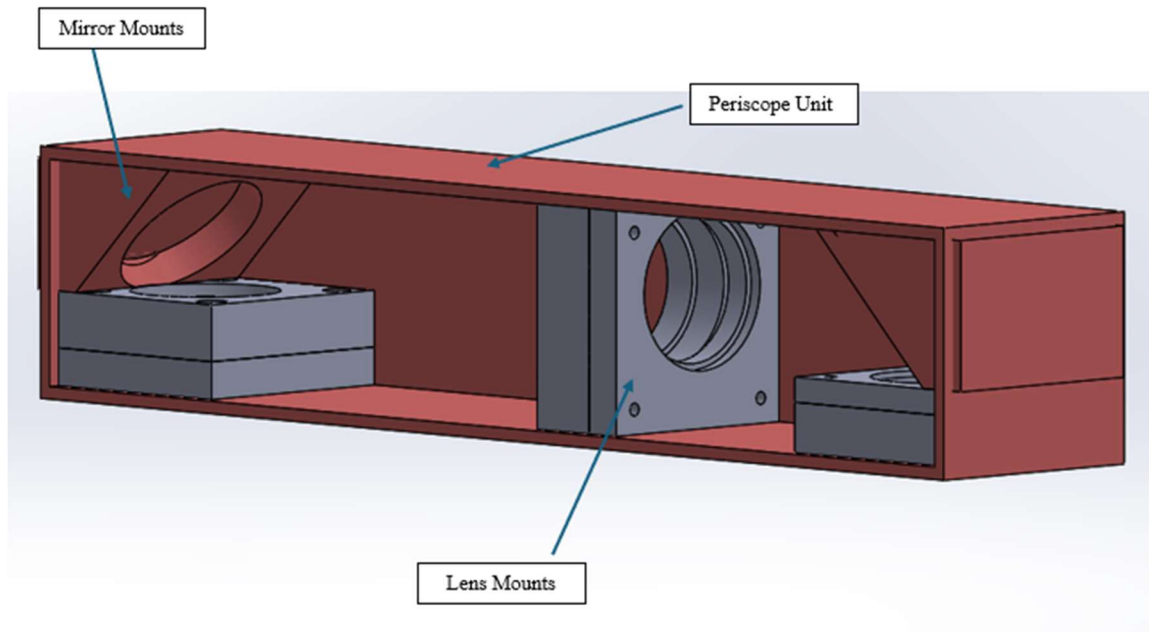
Figure 2.6.4 Camera Unit Top (mm)

This figure shows the assembled upper section of the camera unit, with all dimensions given in millimeters. This section contains most of the illumination components used in the system. The green component represents the annular stop, which helps control the light path. The transparent gray components are the illumination tunnels, and the solid gray components are the lens mounts used to position and secure the optical elements.



**Figure 2.6.5 Camera Unit bottom (mm)**

This figure shows the assembled lower camera section of the camera unit, with all dimensions given in millimeters. This section mainly houses the detection components of the system, including the camera, camera adjustment mechanism, and LED. The transparent gray components represent the tunnels, while the solid gray components represent the lens mounts and the camera adjustment mechanism.



**Figure 2.6.6 Periscope Unit Isometric View (mm)**

This figure shows the assembled periscope unit; all units are millimeters. The transparent gray parts are the lenses and mirrors. It is unable to be seen in this view, but the periscope is also forty millimeters thick.

## 2.7 House of Quality

Correlations	
Positive	+
Negative	-
No Correlation	

Relationships	
Strongly Positive	↑↑
Positive	↑
No Relation	○
Negative	↓
Strongly Negative	↓↓

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

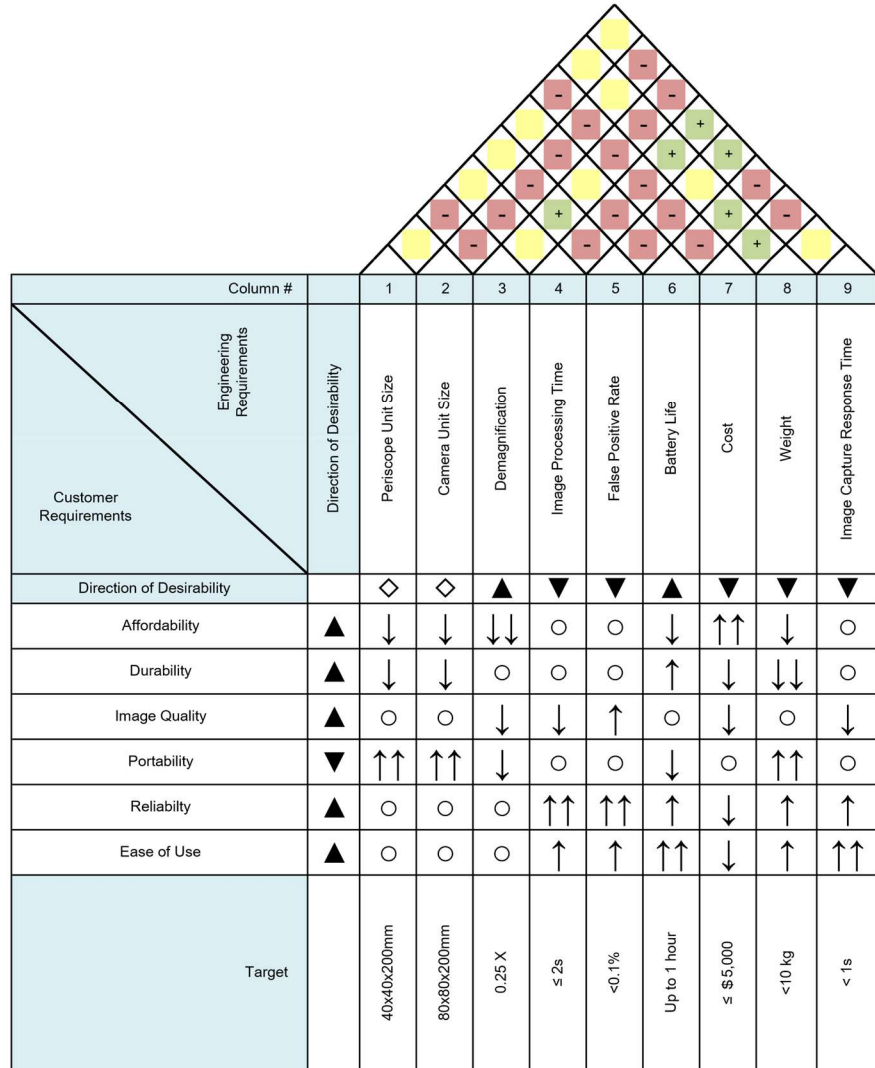


Figure 2.6.1 House of Quality Diagram

## Chapter 3 – Research and Investigation

### 3.1 Hardware Selection

We will discuss the project hardware components in the following section.

#### 3.1.1 Camera Technology

All visible light sensor technologies rely on the use of the photoelectric effect. Fundamentally, the photoelectric effect describes the idea that when particles of light, that exceed, a certain frequency strikes the surface of a metal, electrons are emitted. Photodiodes technology uses this property to convert incoming photons into a readable electrical signal that can be interpreted. The two primary technologies that achieve this are CCD (charge-coupled devices) and CMOS (complementary metal-oxide-semiconductors). While both of these technologies are similar in functionality, the methods in which they achieve it are slightly different.

A CCD sensor sends the charge absorbed per unit time by every pixel to (usually) a single output amplifier where that charge is converted into a voltage signal. Because every pixel is sent to the same amplifier, CCDs produce consistent images with low noise. The only major source of noise in this case is that single amplifier. CMOS sensors, on the other hand, utilize an amplifier for each individual pixel. This fact causes CMOS sensors to have more noise when compared to CCDs but in turn allow each pixel to be read independently, enabling faster processing speed. Another important fact about CMOS sensors is that they output a digital signal directly on the chip. CCDs, in comparison, read out an analog signal which needs additional circuits to digitize later.

Both CCDs and CMOS sensors were invented in the 1960s–1970s, but CMOS sensors were significantly less popular and for good reason. At the time, their efficiency in capturing light and noise was far behind the technological capabilities of a CCD sensor. Over time and development though, this gap started to close. Thanks to further development of CMOS technology, in recent years CMOS sensors have a comparable level of noise to current CCD sensors. In addition to this, CMOS sensors are cheaper to manufacture which has had a snowball effect on its popularity. As the technology has gotten cheaper and more developed, more people have started using CMOS sensors instead of CCDs; Causing a stronger demand for better CMOS sensors. In turn this research decreases the cost of production and increases the amount of research done, again increasing efficiency and then popularity. Nowadays CMOS sensors are a better choice for a large majority of applications.

One big reason for the initial development of CMOS sensors before this snowball effect was the promise of lower power consumption as opposed to CCDs. Due to this lower power consumption, CMOS sensors are preferred for battery powered devices. In terms of dynamic range, or the difference between the brightest and darkest objects your sensor can measure, modern CMOS sensors have an edge on CCD sensors, historically though this was not the case. The thermal performance, or how much heat a system generates, of a CCD sensor is better than that of a CMOS sensor. Despite that fact, CMOS sensors do not generate much heat in general and barred from applications that require low light levels, very long exposure time, or minimized thermal noise, cooling is not necessary.

For applications where cooling is necessary, there are specialized CMOS sensors called CMOS (scientific CMOS) that are designed with external cooling to reduce thermal noise as well as a low dark current. This specialty variety of CMOS sensors is used primarily for scientific or medical imaging where the clearest possible image is crucial. In addition to these features, another important consideration is the quantum efficiency of the sensor. Without much, quantum efficiency is the efficiency in which incoming photons are converted into electrons due to quantum effects. Historically, quantum efficiency has been higher for CCD sensors but due to technological advancements CMOS sensors have comparable or some cases higher quantum efficiency.

Both CMOS and CCD sensors can be configured to be monochrome in addition to chromatic. For both varieties of monochromatic sensor, the quantum efficiency is higher because the monochromatic versions do not have any color filters that would absorb a portion of the incoming light. Another difference between CMOS and CCD sensors is that CCD sensors are susceptible to smearing and bloom. When a pixel gets saturated on a CCD sensor, the charge from that pixel can “smear” into surrounding pixels creating unwanted visual artifacts. CMOS sensors on the other hand are immune to this effect because of its per pixel amplifier.

Another consideration for the type of sensor is whether it uses a rolling or global shutter. For a rolling shutter the image is captured line by line and for a global shutter the entire image is captured at once. The main difference between that two is that a rolling shutter can produce what’s called a “jello effect” which makes an image looked wavy or distorted. This effect only happens when either the camera is moving or the object you are trying to image is moving, like the fans of a helicopter. Considering we are imagining a static image that does not have any movement involved, the shutter type is largely irrelevant to our ability to image at a high quality.

### 3.1.1.1 Camera Technology Selection

**Table 3.1.1 Camera Sensor Type Selection**

Specification	CCD (Charge-Coupled Device)	CMOS (Complementary Metal-Oxide-Semiconductor)
Signal Conversion	Single amplifier converts charge to voltage.	Each pixel has its own amplifier.
Noise Level	Lower noise, uniform	Higher noise
Signal Output	Analog; needs external digitization.	Digital; converted on-chip.
Quantum Efficiency	Historically higher.	Now comparable or higher.
Dynamic Range	Slightly lower in modern sensors.	Generally higher dynamic range.
Power Consumption	High.	Low; better for portable devices.
Thermal Performance	Cooler operation; ideal for long exposures.	Generates more heat; usually cooling not required.

Smear/Bloom Effect	Susceptible to image artifacts.	Immune to smear and bloom.
Speed / Frame Rate	Slower readout.	Faster readout.
Manufacturing Cost	Expensive, complex fabrication.	Cheaper, standard process.
Cooling Needs	Often cooled for precision imaging.	Rarely needed; CMOS offers cooled option.
Typical Use	Scientific, medical, industrial imaging.	Consumer, mobile, automotive applications.
Market Trend	Declining use.	Dominant due to cost and efficiency.

After considering the advantages and disadvantages of both CCD and CMOS technologies, the CMOS sensor seemed to be the obvious choice for our system. Modern day CMOS sensors have similar noise floors, comparable quantum efficiencies, faster processing times, are less expensive. Considering that our imaging system operates with a limited amount of light, maximizing quantum efficiency and minimizing noise is ideal. Although CCD sensors technically beat out CMOS sensors on these aspects, the difference is negligible. While power consumption is not a primary concern, the inherently lower power requirements for a CMOS sensor gives us the best shot to be able to change to a battery-operated design later if desired. Another benefit to selecting CMOS sensors is that we are aiming for an imaging processing time, that is from the moment we send a signal to take a picture to when the computer is processing the image and finishes searching for contaminants, of less than 1 second if possible.

### 3.1.1.2 CMOS Camera Product Comparison

Selecting the appropriate camera is a critical part of ensuring our imaging system is both functional and efficient. Some key factors for consideration are camera dimensions, sensor size, resolution, pixel size, quantum efficiency, dynamic range, interface type, and ability to be integrated effectively. Considering that we have decided to use CMOS sensors rather than CCD sensors, we have identified three camera units for evaluation, each with differing features and capabilities.

While our camera unit is technically limited to a sponsor specified distance area, due to space constraints for the use of other optical components we have decided to split that area in half and try to find cameras that were smaller than a sponsor specified distance in thickness. Another consideration is that our sponsor wants all the components for our system to be off the shelf, meaning designing the camera ourselves or purchasing a board level camera is not ideal. We want the parts to be simple to implement into a final design, further restricting the cameras we can select.

Beyond these physical constraints, there are also a set of optical specifications we are looking for. The required magnification of our system is directly determined by the sensor size by the following equation.

$$M = \frac{\text{Sensor size}}{\text{Object size}}$$

We can likely make most magnifications work with standard sensor sizes by careful optical design, but it is still a fair consideration when selecting a camera. Another important factor that directly affects image quality is the size of the pixel. Since we are imaging particles in the range of 50  $\mu\text{m}$  – 300  $\mu\text{m}$  for detection and not features of those particles, our system is not diffraction limited. This means that when considering pixel size, we do not need super fine pixels because the optics are not the limiting factor for final image quality. Since we are imaging with relatively low light conditions, we want to have larger pixel sizes to collect more light and improve the signal strength per pixel, and in turn the signal-to-noise ratio of the system. The resolution of most cameras of the quality range we are looking for should be sufficient for capturing the level of detail required.

Another important consideration for maximizing the signal-to-noise ratio of our system is the devices quantum efficiency. Quantum efficiency measures the ratio of incident photons that are converted into a displaced electron. Considering we are expecting to image low amounts of light, A higher quantum efficiency means that our system will be more sensitive to lower amount of incoming light.

An important tradeoff we made here is whether to go for color or monochromatic cameras. While you can convert the output of a color camera into a monochromatic image, the quantum efficiency of a monochromatic sensor is higher than the quantum efficiency of its chromatic counterpart, allowing for better sensitivity. While this is technically a downside, we decided to use the color version. Choosing a color camera allows us to potentially reach our stretch goal of adding a brightfield function to our design. Should we decide to switch our green LED source for different color LED or even a white LED, our camera will be equipped to handle it.

Other important considerations for choice of the camera is the dynamic range, or the difference between the brightest and darkest regions a camera can image, and the exposure time, or the amount of time a camera sensor is exposed to light before the image is read. A larger dynamic range allows the system to capture bright reflections as well as low light without saturating the pixel. A flexible exposure time guarantees that we can image low light conditions while maintaining enough signal strength to be unaffected by the level of noise.

### **3.1.1.3 IDS U3-3880CP-C-HQ**

The IDS U3 3880CP C HQ is a color Progressive Scan CMOS camera from IDS Imaging Development Systems featuring a 1/1.8" (8.9 mm diagonal) Sony IMX178 sensor with a resolution of 3088  $\times$  2076 pixels (6.41 MP) and a pixel size of 2.40  $\mu\text{m}$   $\times$  2.40  $\mu\text{m}$  over a sensing area of 7.411 mm  $\times$  4.982 mm. The camera is designed for manufacturing, industrial, and automation applications. The camera can achieve a maximum full resolution frame rate of around 59 fps. The housing measures 29 mm  $\times$  29 mm  $\times$  29 mm (excluding connectors and lens mount) and weighs approximately 48 g. The camera uses USB 3.0 interface and is powered by the USB.

The shutter type is rolling, exposure time ranges from about 0.014 ms to 1060 ms, and the dynamic range is specified at about 71 dB. Power consumption is rated between roughly 1.4 and 3 W, and the camera operates in the 0 °C to +55 °C range. The quantum efficiency is not specified and the cost is listed as \$515.04 on Edmund Optics' website.

The IDS U3 3880CP C HQ seemed like a fair camera to consider given its specifications. This camera is very compact and easily fits within our size constraints. The Sony IMX178 has a strong dynamic range and the camera has a broad exposure range. The resolution of 6.4 Megapixels is more than fair for our application, but the pixel size of 2.4 microns is smaller than ideal, potentially affecting the signal-to-noise ratio for low light applications. The sensing area of 7.411mm x 4.982 mm is fairly large and would allow for a magnification of near 0.5x. The camera is around the same price as the CS165CU which makes the two cameras easier to compare. This camera also utilizes IDS Imaging's peak SDK platform that supports USB3 Vision standards. This will allow for relatively easy software development and integration. However, a major downside though is that the quantum efficiency is not listed making it more challenging to know how well the camera will function in that regard.

Overall, this camera is a viable option due to its size, dynamic range, and exposure range. However, given its smaller pixel size and comparable price point to the better documented Thorlabs CS165CU, the IDS U3 3880CP C HQ will likely rank below the other options.

#### **3.1.1.4 CS165CU**

The Thorlabs CS165CU is a color CMOS camera designed for general-purpose imaging applications such as brightfield microscopy and machine vision. It features a 1/2.9" (6.2 mm diagonal) sensor with a resolution of 1440 × 1080 pixels (1.6 MP) and a pixel size of 3.45 μm × 3.45 μm over an imaging area of 4.968 mm × 3.726 mm. The camera employs a global shutter and supports a maximum full-resolution frame rate of 34.8 fps. The read noise is specified as less than 4 e<sup>-</sup> RMS, with a full-well capacity of at least 11,000 e<sup>-</sup> and a dynamic range of up to 69 dB.

Exposure time ranges from 0.040 ms to 26,843 ms in around 0.025ms increments. The housing measures approximately 43.7 mm × 47.2 mm × 20.7 mm and includes internal SM1 threading with adapters for C- and CS-mount lenses. The camera consumes 1.17 W of power via USB and operates in ambient temperatures from 10 °C to 40 °C. The CS165CU includes a removable IR-blocking filter and relies on passive cooling. The quantum efficiency is around 0.65 at 535 nm light, around 0.625 for 620 nm light, and around 0.54 for 465 nm light. The cost is listed as \$531.45 on Thorlabs' website.

The Thorlabs CS165CU seemed to be the strongest contender for the camera we ultimately decide to purchase. The fantastic and dense documentation allows us to guarantee we understand exactly what we are getting. Furthermore, The larger pixel size allows us to achieve a better signal-to-noise ratio, considering we aren't diffraction limited. While the physical dimensions of the casing is slightly higher than expected, it still is possible to fit it into our design. The sensor size of 4.968 mm x 3.726 mm is smaller than both the other cameras we are comparing, leading to a higher needed magnification.

Although, this problem can be mitigated with intelligent optical design. The dynamic range as well as the exposure time should be sufficient to meet our purposes.

In addition to these benefits for selecting the CS165CU, an additional major benefit is familiarity. This camera is used for some of the undergraduate labs here at CREOL. Almost every UCF PSE student should be somewhat familiar with the software ThorCam. This ensure that our team will have the easiest possible time integrating this device into the rest of our system. Overall, the specifications of this device in addition to that familiarity should make this camera a simple top choice.

### 3.1.1.5 Teledyne DR-U3-50Y2C-C1-CS

The Teledyne DR-U3-50Y2C-C1-CS is a color CMOS camera from Teledyne FLIR’s Dragonfly S series designed for embedded and machine vision applications. It features a 1/2.5" (6.4 mm diagonal) onsemi AR0521 sensor with a resolution of 2592 × 1944 pixels (5.0 MP) and a pixel size of 2.2 μm × 2.2 μm over an imaging area of approximately 5.70 mm × 4.28 mm. The camera employs a rolling shutter and supports a maximum full-resolution frame rate of 49 fps. The housing measures 29.5 mm × 29.5 mm × 23 mm and weighs approximately 17 g. The camera uses a USB 3.1 Micro-B back connector and is powered through the 5 V USB interface. This camera uses the Spinnaker SFK software for operation.

Exposure time ranges from 9 μs to 0.5 s. The camera has a dynamic range of 69 dB. The noise is specified at 3.28 e<sup>-</sup> RMS, with a saturation capacity of 10,558 e<sup>-</sup>. The quantum efficiency is approximately 65 % at 470 nm, 74 % at 525 nm, and 48 % at 630 nm. The housing provides a CS-mount lens and has one auxiliary output rated for 4.3 V to 5.25 V up to 200 mA. The camera operates in ambient temperatures from 0 °C to +50 °C. Power consumption is rated at 2.4 W maximum. The camera does not include an optical filter. The cost is listed as \$199 on Teledyne’s and Edmund Optics’ websites.

The Teledyne DR-U3-50Y2C-C1-CS offers several different advantages, including its dimensions, high resolution, and competitive price point to the other two options. Its higher quantum efficiency in comparison is another attractive feature. However, the smaller pixel size may again limit the signal-to-noise ratio when compared to the CS165CU; although, this camera does have a stated noise which serves to counteract that fact slightly. Another downside is that this camera is limited to 0.5 seconds of exposure which may impact our ability to test our system. The USB 3 port being on the back may also serve to save some space in the final camera module design. Overall, this camera is a strong budget-friendly alternative that despite some of the tradeoffs, is a strong consideration.

### 3.1.1.6 CMOS Camera Product Selection

Table 3.1.2 CMOS Camera Part Selection

Specification	IDS U3-3880CP C HQ	Thorlabs CS165CU	Teledyne DR-U3-50Y2C-C1-CS
Sensor size / format	1/1.8"(8.9 mm diagonal)	1/2.9"(6.2 mm diagonal)	1/2.5" (6.4 mm diagonal)

	–Imaging area: 7.411 mm × 4.982 mm	–Imaging area: 4.968 mm × 3.726 mm	–Imaging area: 5.70 mm × 4.28 mm
Resolution (pixels)	3088 × 2076 (6.41 MP)	1440 × 1080 (1.6 MP)	2592 × 1944 (5.0 MP)
Pixel size	2.40 μm × 2.40 μm	3.45 μm × 3.45 μm	~2.2 μm × 2.2 μm
Shutter type	Rolling	Global shutter	Rolling
Max frame rate (full resolution)	~59 fps	~34.8 fps	~49 fps
Dynamic range	~71 dB	Up to ~69 dB	~69 dB
Quantum efficiency (peak)	Not specified	~65% at 535 nm	~74% at 525 nm, ~65% at 470 nm, ~48% at 630 nm
Housing / physical size	29 mm × 29 mm × 29 mm (excluding connectors/lens mount)	~43.7 mm × 47.2 mm × 20.7 mm	29.5 mm × 29.5 mm × 23 mm
Exposure time range	0.014 ms – 1060 ms	0.040 ms – 26,843 ms	0.009 ms – 500 ms
Interface type	USB 3.0	USB 2.0	USB 3.1 Micro-B
Software / SDK	IDS peak SDK (USB3 Vision compatible)	ThorCam	Spinnaker SDK
Power consumption	~1.4 – 3 W	1.17 W	2.4 W max
Cost	\$515.04	\$513.45	\$199.00
Comments	Very compact, good dynamic range, but small pixels and unknown QE is downside	Larger pixel size aligns well with low-light & SNR goal, significantly larger exposure time range, strong documentation & familiarity	Best price, good quantum efficiency, very compact, but smaller pixels

After considering these three options, the Thorlabs CS165CU was chosen as the most optimal camera for our system. Its larger pixel size improves signal-to-noise ratio under low-light conditions, its strong documentation, as well as its large exposure range ensures both predictable performance and the ability to optimize our system need anything change.

While slightly larger than our ideal dimensions, the camera still fits within our design parameters. In addition, our team’s familiarity with the ThorCam software simplifies its usability. Even though the cost of the CS165CU is much greater than the Teledyne DR U3 50Y2C C1 CS, we believe the benefits gained outweigh the difference. These factors and more make the CS165CU our most reliable and practical choice for both current needs and potential future stretch goals, such as brightfield imaging.

### 3.1.2 Lens Technology

Lens design can affect the final image quality of our optical system. Besides a lens’ geometry, the material or materials used to make a lens can also greatly affect the final caliber of our image. Considering light will pass forwards and backwards for parts of multiple 4f systems, minimizing aberrations ensures our system retains its quality while it propagates through our system.

#### 3.1.2.1 Singlet Lenses

A singlet lens is a lens, usually biconvex, planoconvex, or aspheric, that is made of one material. In comparison to achromatic and apochromatic lens, a singlet lens is much cheaper to manufacture but has strong chromatic aberration, leading to a situation where the rays will focus to different points for different colors. Singlet lenses are typically used for simple optical devices such as magnifying glasses or eyeglasses.

#### 3.1.2.2 Achromatic Doublet

Achromatic Doublets are lenses, usually biconvex, that are made of two different materials. The lenses are usually made from glasses with different refractive indices and varying amounts of dispersion. Often times the doublet lens is made from one positive lens, usually called the crown glass, and one negative, usually called the flint. These different glasses are designed to correct chromatic aberration for two colors, bringing two color to the focus. These two individual lenses can be cemented together or mounted together.

#### 3.1.2.3 Apochromatic Lenses

Apochromatic lenses are similar to achromatic doublets in the fact that they are designed to limit chromatic aberration. Apochromats are another level of quality above, are usually made of three or more lenses and naturally cost more. For our applications, apochromatic would be nice to have, but is strictly unnecessary.

#### 3.1.2.4 Lens Technology Selection

Table 3.1.3 Lens Type Comparisons

Property	Singlet	Achromatic Doublet	Apochromat
Chromatic correction	Poor	Moderate (2 wavelengths)	Excellent (3+ wavelengths)
Spherical aberration	Moderate	Moderate–Low	Low
Cost	Low	Medium	High

Property	Singlet	Achromatic Doublet	Apochromat
Complexity	Simple	Moderate	High
Alignment sensitivity	Low	Moderate	High
Common applications	Laser collimation, monochromatic imaging	General visible imaging, relay systems	Precision imaging, microscopy, broadband systems

For our imaging system we will be using a mix of achromatic doublet and singlet lenses. We will use a singlet planoconvex lens to collimate the green LED inside the camera module, other wise all other lenses, especially those that will be a part of a 4f system, will be biconvex achromatic lenses. Although, currently we are using a green LED, the ability to switch to a white LED later allows us to potentially add a brightfield option later. Overall, apochromats have stronger chromatic aberration correction corrections, and for their price point they are a lot more expensive.

For the vendor, we have decided to purchase all our achromatic doublet lenses and lenses from Thorlabs. We also considered Newport and Edmund Optics as well, although there was not a massive difference in quality or price, Thorlabs has a greater selection and better documentation which greatly aided our ability to design an optimized system.

### 3.1.3 Illumination Light Source Technology

Microscopy systems utilize illumination sources to enhance the brightness and contrast of an object. The most common microscopy illumination sources are halogen lamps, arc lamps, light-emitting diodes (LEDs) and lasers. Each illumination source has its advantages for different microscopy techniques and applications. These advantages are due to the coherence, collimation, optical power and power efficiency of each illumination source. Darkfield illumination requires a light source that emits bright and incoherent light. This is due to the scatter produced off an object can cause interference fringes at the image if a coherent light is used as the illumination source. We will also keep dimensions, cost and integration efforts in mind to meet our project specifications.

#### 3.1.3.1 Halogen Lamps

Halogen Lamps emit a broad spectrum of bright light with very low coherence, making them standard in classic darkfield illumination optical systems. They are an incandescent light bulb that emits light by running current through a tungsten wire, which glows when heated up. This glow emits a broadband of light in the visible and infrared range. As the tungsten evaporates off the wire over time, it reacts with the halogen gas atoms to create tungsten halide molecules, which condense back into the wire depositing into tungsten, completing the halogen regenerative cycle. This cycle lengthens the bulb's life, which made the halogen lamp the standard incandescent light bulb for decades.

Due to the nature of incandescent light bulbs, halogen lamps have poor power efficiency anticipated by a lot of power being wasted in the form of heat. This excess heat

raises concern with the way it would interact with our housing and neighboring components. Microscopes that utilize halogen lamps often equip them with intensive cooling systems, making the light source larger in size. This makes it more challenging for them to integrate into our system; however, they can be found at low cost due to the abundance of options on the market.

### **3.1.3.2 Arc Lamps**

Arc lamps emit a broad spectrum of the brightest light with low coherence, making them a solid option as a light source for darkfield illumination. They are mainly used in optical systems with a high NA condenser due to their high intensity. They are a gas-discharged light source that emits light from an electric arc (plasma) made that is the result of inputting a high voltage pulse into two electrodes a few millimeters apart housed in an easily ionized gas such as Xenon or Mercury. The gas used affects the optical spectrum of the emitted light, enabling more control which can decrease the signal to noise ratio of the detected image from our system.

Similar to halogen lamps, arc lamps have low power efficiency due to a lot of power being wasted into heat. Arc lamps also require more intensive cooling systems since they are the warmest in temperature out of all our options. This will make it more challenging to integrate into our system due to the same reasons stated earlier. Microscopy systems that utilize arc lamps are a lot larger in size with applications that require a high light intensity for illumination. They are also more expensive with shorter lifetimes than halogen bulbs and LEDs.

### **3.1.3.3 Light-Emitting Diodes (LEDs)**

LEDs emit a broad spectrum of bright light with low coherence, making them ideal for darkfield illumination. They have replaced halogen lamps and arc lamps in most modern applications due to their marginal power efficiency increase. This is due to LEDs emitting light straight from electron transitions rather than heating a material or producing plasma. The lack of needing an intensive cooling system allows them to be much more compact.

LEDs also have a narrower spectrum compared to halogen and arc lamps, allowing us to choose the best center wavelength for our optical system. Our research showed green would cause the least amount of chromatic aberrations since it is in the center of the visible spectrum, so most achromatic lenses perform best in this wavelength range. LEDs are compact, very bright and produced at a low cost, making them a standard in modern life.

### **3.1.3.4 Lasers**

Lasers emit a very bright narrow spectrum of coherent light, making them the worst option for our reflectance-based project. This would cause the interference fringes in the image mentioned earlier from the coherent beams scattering off the object. Lasers emit light from pumping energy into a gain medium which amplifies the optical properties due to an optical cavity reflecting the photons via mirrors. The gain mediums vary, but laser diodes are most common. This results in stimulated emission of a very controlled beam of light. Lasers are still used in modern day microscopy, in fluorescence excitation and laser scanning microscopy for example. They can emit light as bright as arc lamps, which makes

them very useful in high NA applications. To utilize this brightness in a darkfield illumination system is not impossible but would require an intensive optical setup to magnify and diffuse the beam which our dimensional constraints cannot afford.

### 3.1.3.5 Illumination Light Source Technology Selection

**Table 3.1.4 Illumination Light Source Technology Comparison**

Light Source Type	Collimation	Coherence	Bandwidth	Power Efficiency	Integration	Cost
Halogen Lamps	Low	Low	Very Wide	Very Low	Challenging due to dimensions	\$15-\$100
Arc Lamps	Low	Low	Wide	Low	Challenging due to dimensions	\$200-\$800
LEDs	Low	Low	Moderate	High	Ideal	\$1-\$100
Lasers	High	Narrow	Very High	Very High	Challenging due to dimensions	\$25-\$200

Clearly, the LED is the best option for our illumination system due to its power efficiency, size, optical properties and cost.

### 3.1.3.6 LED Product Comparison

As mentioned earlier, we will be looking at LEDs in the green wavelength range to limit chromatic aberrations in our detected image. This is because green is in the middle of the visible optical spectrum, so achromatic doublets are designed to perform best in this wavelength range. We will also compare high power LEDs since our system utilizes reflective dark field imaging techniques that lose a lot of the illumination power to the annular stop and only detects the scattered light. Our research showed that a 10 Watt LED with brightness between 500 to 1000 lumens should be sufficient for our project. We will compare the Green 10W High Power LED by Switch Electronics (311448), the 10 Watt Green LED by Vetco Electronics (VUPN1856) and the High Power LED Chip 10W Green by CHANZON. Since all these LEDs have very similar housing, we will primarily focus on comparing their viewing angle, luminous flux, power draw and price.

### 3.1.3.7 Green 10W High Power LED by Switch Electronics (311448)

The Green 10W High Power LED by Switch Electronics is a surface mount 525 nm LED with a luminous flux of 650 lumens. This LED requires an input current of 900 mA and an input voltage of 11 V, which is typical electrical power draw of high-power LEDs.

This LED also has a viewing angle of 140 degrees, which is similar to the other LEDs we are comparing.

The main negative about this LED choice is the lack of information and specifications on the switch electronics website. There is no information regarding dimensions or recommended LED drivers, making it more complicated to integrate into our system. Switch Electronics is also based in England, making it more costly to ship and replace if needed

### 3.1.3.8 10 Watt Green LED by Vetco Electronics (VUPN1856)

The 10 Watt Green LED by Vetco Electronics is very similar to the one by Switch Electronics . This LED emits 520 to 525 nm light with the luminous flux of up to 700 lumens. The LED requires an input voltage of 9-12V with an operating current of 900 mA. This and a 140 degrees viewing angle makes it very similar to the switch electronics LED.

Also, like the Switch Electronics LED, the main negative about this choice is the lack of information and specifications on the Vetco Electronics website. There is no information regarding dimensions or recommended LED drivers, making it more complicated to integrate into our system. Vetco Electronics is based in Bellevue, Washington, making shipping and the cost of replacement cheaper compared to the Switch Electronics LED.

### 3.1.3.9 High Power LED Chip 10W Green by CHANZON

The High Power LED Chip is very similar in specifications to the Vetco Electronics and the Switch Electronics LEDs. This LED emits 520 to 525 nm light with a luminous flux of up to 800 lumens, making it the brightest option of all three LEDs. The LED requires an input voltage of 9-12V with an operating current of 840 to 1000 mA. A 140 degrees viewing angle makes it very similar to the LEDs mentioned earlier.

What differentiates this LED is the additional information and specifications provided by the CHANZON amazon store. They provide engineering dimensional drawings and recommend an LED driver which will make it easier to integrate into our system safely

### 3.1.3.10 LED Product Selection

Table 3.1.5 LED Product Selection

Green LED	Input Voltage	Input Current	Wavelength	Viewing Angle	Luminous Flux	Integration	Cost
Green 10W High Power LED by Switch Electronics (311448)	11 V	900 mA	25 nm	140°	650 Lumens	Challenging due to lack of driver	\$8.12

10 Watt Green LED by Vetco Electronics (VUPN1856)	9 – 12 V	900 mA	520 – 525 nm	140°	700 Lumens	Challenging due to lack of driver	\$8.95
High Power LED Chip 10W Green by CHANZON	9 – 12 V	840 – 1000 mA	520 – 525 nm	140°	800 Lumens	Ideal	\$6.99

Due to the higher brightness and additional information provided with the LED by CHANZON, it is the LED we are selecting to be the illumination source for our darkfield illumination system. The ability to purchase them on Amazon also gives us reasonable shipping time in case we need to purchase another one if one fails during testing.

### 3.1.4 Collimating Lens Technology

The collimating lens in our project is used to transform the diverging rays of the LED to a set of parallel rays. This collimated beam enables the maintenance of our illumination source power and masks the annular stop onto the lens system. The divergence of the LED causes most of the rays to miss the lens following the annular stop if the illumination source is not collimated. We are looking for a lens with a short focal length and high beam quality. Cost can be a little higher since we only need one lens and low importance on aberration performance since we are planning to use a green LED.

#### 3.1.4.1 Aspheric Lenses

Aspheric lenses are designed with surfaces that are more complex than simple spherical curvatures. This design decreases chromatic and spherical aberrations and improves beam collimation properties. This high performance and small size make them effective for many applications in space, telecommunication and bio-medical industries where high precision collimation is critical. The complex manufacturing process of producing complex design makes them the most expensive option. As mentioned earlier, the aberration properties of the lens are not critical in our illumination sub-system, therefore we will not use aspheric lenses as our collimating lens due to their high cost.

#### 3.1.4.2 Best Form Spherical Lenses

Spherical lenses are bi-convex or plano-convex lenses that are usually used for simple focusing or collimating tasks. Best form spherical lenses keep the simplicity but are designed to minimize the effects of spherical aberrations. This improvement comes from special optimizations during the design process for a specific focal length and aperture, improving the collimated beam quality. These lenses are more expensive than traditional spherical lenses but are still relatively cheap, with most costing less than a \$100.

### 3.1.4.3 Plano-Convex Lenses

Plano-convex lenses are the most common lens used to collimate light sources. This is straightforward, with the plano side of the lens facing the light source, the convex side controls the divergence of the light into a collimated beam. This collimated beam quality is higher than traditional spherical lenses but lower than best form spherical lenses. This is due to the poor performance of spherical aberrations when collimating light sources with a plano-convex lens. These lenses can be found cheap, with some options costing as low as \$20.

### 3.1.4.4 Collimating Lens Technology Selection

**Table 3.1.6 Collimating Lens Technology Selection**

Lens Type	Beam Collimation Quality	Minimum Focal Length Available	Spherical Aberration Compensation	Cost
Aspheric Lens	High	20 mm	High	\$200 – \$2000
Best Form Spherical Lenses	High	40 mm	Medium	\$40 – \$100
Plano-Convex	Medium	25 mm	Low	\$20 – \$100

A best form spherical lens was chosen as the collimating lens for our project due to its high beam quality and low cost. We will now compare best form spherical lenses that are available off-the-shelf.

### 3.1.4.5 LBF254-040

Thorlabs' LBF254-040 is a one-inch best form spherical lens with the focal length of 40 mm. Even though this focal length is longer than we wanted, it is the shortest available and it will fit in the camera module's housing. This lens is uncoated, which can improve the durability of the lens since damaging the coating can be neglected. However, since we are losing a lot of power due to collimation and annular stop, anti-reflection coating is beneficial since it improves the collimated beam's power.

### 3.1.4.6 LBF254-040-A

Thorlabs' LBF254-040-A is a one-inch best form spherical lens with the focal length of 40 mm and an anti-reflection coating in the 350 nm – 700 nm. This is the same lens discussed earlier but has the AR coating. This coating is beneficial to our system since it increases the power of the collimated beam. The only negative about this lens is that the coating may burn in high power applications, but our 10 W LED should not cause this. This coating increases the price of the lens by only about \$15.

### 3.1.4.7 LBF254-040-B

Thorlabs' LBF254-040-B is a one-inch best form spherical lens with the focal length of 40 mm and an anti-reflection coating in the 350 nm – 1050 nm. This is the same lens discussed earlier but has the AR coating. This coating is not beneficial to our system since its AR coating is for wavelengths not emitted from the LED.

### 3.1.4.8 Best Form Spherical Lens Product Selection

**Table 3.1.7 Best Form Spherical Lens Part Selection**

Lens	Beam Collimation Quality	Focal Length	Coating	Cost
LBF254-040	High	40 mm	N/A	\$54.67
LBF254-040-A	High	40 mm	350 nm – 700 nm AR	\$67.06
LBF254-040-B	High	40 mm	650 nm – 1050 nm	\$67.06

Thorlabs' LBF254-040-A was chosen since it had the highest performance at a relatively cheap price.

### 3.1.5 Optical Mirror Technology

Our project utilizes a total of seven mirrors, all with the same purpose, bending the beams optical path by 90 degrees with the least amount of distortion possible. We use two mirrors in the periscope unit, two in the illumination path in the camera module and three in the detection path in the camera module. We compared different types of plano mirrors to identify the type that will work best for our system. Budget will play a bigger role in deciding which mirror we are going to use due to the high quantity of mirrors in our system

#### 3.1.5.1 Front-Surface Plano Metallic Mirrors

Front-surface metallic mirrors have their reflective coating on the top of the glass. This allows them to limit refraction and reflection that would occur if the reflective coating were on the back of the glass. The light is reflected from the free electrons in the metallic layer. This allows for a very broadband of wavelengths to be reflected. The reflectivity of the mirrors ranges from around 85% to 98% depending on the type of metal used as the coating. Typical metal coatings are aluminum, silver and gold. These coatings' biggest weakness is their durability, where they can scratch relatively easily, so scientists have developed protective coating, but they decrease the reflectivity percentage of the mirrors

These mirrors can be found relatively cheap, especially when they use cheap metals for their coating. This and the high performance of broadband reflectivity makes them a very common type of mirror used in many optical systems.

### 3.1.5.2 Front-Surface Plano Dielectric Mirrors

Similar to front-surface plano metallic mirrors, front-surface plano dielectric mirrors have their reflective coating on the top of the glass which limits the refraction and reflection that would occur if the reflective coating were on the back of the glass. The dielectric surface utilizes interference that occurs from multiple layers of dielectric materials that have different refractive indexes to reflect light. They have a way higher reflectivity compared to metallic coated mirrors, where they can reflect up to 99% of the incident light from 0° to 45° angle of incidence.

The main problem that dielectric mirrors have is that they have that high reflectivity for a narrowband of wavelengths and that they cost more than metallic mirrors. Luckily, this “narrowband” is still wide enough to fit most of the visual spectrum. This will allow us to integrate dielectric mirrors into our system to utilize the high reflection properties of them.

### 3.1.5.3 Back-Surface Plano Mirrors

Back-surface mirrors have their reflective coating on the back side of the glass. This type of mirror is not usually used in precise optical systems due to the refraction that occurs in the mirror glass before it hits the reflective coating. These types of mirrors are typical household or car mirrors. They are also used in optical systems that require partial transmission through the mirror, such as interferometry. Since the glass is on the outside, it protects the reflective coat from the environment making them more durable.

### 3.1.5.4 Right-Angle Prism Mirrors

Right-angle prism mirrors have a specialty where they can reflect light via total internal reflection or via metallic coating. The prism is composed of two uncoated surfaces on the side with the dielectric coating being on the hypotenuse of the “right triangle”. If light is incident on the hypotenuse from inside the prism, where the incident light angle is larger than the critical angle, almost all the light is reflected without flipping the polarization. If the light is incident on the outside of the hypotenuse, typical 90-degree beam fold occurs. This makes them a great contender to our project due to their large surface area, however, our budget is sensitive to mirror cost since our system needs seven mirrors in total, so a small premium will add up quickly.

### 3.1.5.5 Optical Mirror Technology Selection

Table 3.1.8 Mirrors Technology Selection

	Reflectance	Wavelength	Integration	Cost
Front-Surface Plano Metallic Mirrors	85% – 98%	250 nm – 20 μm	Low Effort	\$40 – \$200

Front-Surface Plano Dielectric Mirrors	>99%	400 nm – 750 nm	Low Effort	\$80 – \$200
Back-Surface Plano Mirror	>99%	450 nm – 20 μm	Low Effort	\$80 – \$130
Right-Angle Prism Mirrors	>99%	400 nm – 750 nm	Ideal	\$70 – \$320

We have decided to move forward with front-surface plano dielectric mirrors for their high reflectance at reasonably affordable prices. Even though it is only a few percent difference in reflection, our projects budget is used on this due to the amount of light our image and illumination are losing through the optical path.

### 3.1.5.6 Plano Broadband Dielectric Mirrors Product Comparison

We have decided to use front-surface plano dielectric mirrors with a one-inch diameter to fit into the periscope unit's and camera module dimension constraints. Higher reflectivity is needed due to the low NA of our system that is restricted to the dimension constraints. The main difference between the dielectric mirrors we are comparing is the substrate material.

### 3.1.5.7 Fused Silica Broadband Dielectric Mirrors (BB1-E02)

Fused silica broadband dielectric mirrors are composed of amorphous silica glass with a dielectric material coating. The properties of the silica glass give these mirrors high thermal stability with low thermal expansion. Silica glass is also very mechanically strong and hard. They are also cheaper compared to Zerodur broadband dielectric mirrors.

### 3.1.5.8 Zerodur® Broadband Dielectric Mirrors (BB111-E02)

Zerodur broadband dielectric mirrors are composed of a glass ceramic material that is produced by SCHOTT with a dielectric reflective coating on one side of the glass. This glass ceramic material has the lowest thermal expansion property that is descended from its best-in-class thermal stability. The main negatives of this type of dielectric mirror are that they are weaker mechanically and more expensive compared to the

### 3.1.5.9 Plano Broadband Dielectric Mirrors Product Selection

Table 3.1.9 Mirrors Product Selection

Mirror Type	Reflectance	Wavelength	Integration	Cost
BB1-E02	>99%	400 nm – 750 nm	Low Effort	\$86.91
BB111-E02	>99%	400 nm – 750 nm	Low Effort	\$143.64

Since our project does not include a high-power laser that might cause thermal expansion, we have decided to move forward with the fused silica broadband dielectric mirrors due to their high reflectance and reasonable price.

### **3.1.6 Optical Beamsplitter Technology**

Our project utilizes a beamsplitter to reflect the illumination beam out of the camera module into the periscope unit and transmit the detection beam from the periscope unit into the detection path of the camera module. This component must have a close to even reflectance and transmittance ratio to ensure there is enough light to illuminate the sample and enough light in the detection path. One of the reasons we chose such a high-power LED for illumination is that we would only detect about a quarter of the illumination source power. We also must ensure that the beamsplitter has the proper coating to be effective in the visual spectrum. We will choose a beam splitter with minimal wavefront distortion and beam deviation to ensure the highest quality image possible. We will also try to minimize the ghost reflections, which are the bright spots created when light reflects multiple times between surfaces. Beamsplitters are also sensitive to polarization of light, which we will keep in mind for our stretch goal of detecting smaller particles which can use polarization properties to enhance the contrast of opaque samples.

#### **3.1.6.1 Plate Beamsplitters**

Plate beamsplitters are made from flat glass disks with different coatings on each side. One side is coated with a partial reflector that controls the beamsplitters' reflection and transmission ratio. The other side has an anti-reflection coating to minimize ghosting reflections. They most commonly utilize a 45-degree angle on incidence. Since these plates are made from high quality glass such as BK7 or fused silica, there is relatively low image distortion. High levels of surface flatness and quality also ensure a high-quality image. However, these kind of beamsplitter still have lateral displacement of the transmitted beam due to the refraction that occurs inside the plate.

Plate beamsplitters are the cheapest option with an abundance of shapes and sizes options on the market. This is one of its strength since it will make it easier for us to integrate into our dimensions limited system. They often utilize non-polarizing coatings to overcome their shortcoming of having different reflecting properties depending on the polarization of the incident light.

#### **3.1.6.2 Cube Beamsplitters**

Cube beamsplitters are made from two right angle prisms that are cemented together on their hypotenuse side. There is a partial reflector placed between the two prisms that is similar to the one used in plate beamsplitter. To minimize ghosting, this type of beamsplitter must use a refractive index match cement between the two prisms. This homogeneity of the refractive index disables secondary reflection in the transmitted light beam. These beamsplitters also have high levels of surface flatness and quality that ensure a high-quality image

What makes cube beamsplitters unusable in our system is the fact that light only behaves as expected only when light is incident on one side of the cube. When light is

incident on the orthogonal side, there are altered reflective and transmission properties due to the partial reflector and anti-reflection coatings not being symmetric. This will not work in our design since our reflected image will be incident on the output side of the cube if our illumination beam is on the input side of the cube. This is unfortunate since cube beamsplitters have minimal aberrations with no beam offset with abundance of size options on the market. They are also the easiest to mount and align due to their simple geometry.

### 3.1.6.3 Pellicle Beamsplitters

Pellicle beamsplitters are made from multilayers of coated films with different refractive indexes that control the reflection and transmission of the beamsplitter. This partially reflective coating is applied to one side of the film. Adjusting the thickness and the refractive indexes of these layers enables precise design on the amount of light transmitted and reflected, however, this depends on the wavelength of the incident. They are also the thinnest beamsplitters available, which means minimal ghosting and beam displacement due to the lack of refraction inside the film. Combined with high levels of surface flatness and quality also ensures the best-quality image out of our beamsplitter options. The only negative of this ultra-thin design is the beamsplitter’s durability in harsh environments. The thin film is usually mounted on a metal or plastic frame to overcome some of these durability concerns.

Pellicle beamsplitters are more expensive than plate beamsplitters but around the same price as cube beamsplitters. They don’t have as many size options on the market but the dimensions and R:T ratio specifications for our project are available. Unlike cube beamsplitters, the reflected image in our detection path will undergo the same optical properties as the illumination path (R:T might cause slight differences but will maintain image quality), making them useful in our optical system. Pellicle beam splitters also have low polarization dependence, making them ideal for the stretch goals of our project.

### 3.1.6.4 Optical Beamsplitters Technology Selection

**Table 3.1.10 Beamsplitter Technology Selection**

Beamsplitter Type	Split Ratio (R:T)	Wavelength	Ghost Reflections	Beam Displacement	Polarization Dependence	Integration	Cost
Plate Beamsplitters	50:50	400 nm – 700 nm	Moderate	Moderate	High	Low Effort	\$90 – \$300
Cube Beamsplitters	50:50	400 nm – 700 nm	Low	Low	Moderate	None	\$190 – \$600
Pellicle Beamsplitters	45:55	400 nm – 700 nm	Low	Low	Moderate	Low Effort	\$200 – \$300

Since we have enough budget and want the best image quality possible, we wanted to move forward with the pellicle beamsplitter as our choice. However, we ended up using a beamsplitter for its moderate performance and high durability.

### 3.1.6.5 Plate Beamsplitter Product Comparison

To decide on which plate beamsplitter to use in our camera module, we will compare options on the market for their working diameter, overall size, split ratio and wavelength performance. Comparing these specifications will allow us to choose the ideal beamsplitter to our system.

### 3.1.6.6 Plate Beamsplitter, 25.4mm 50R/50T (BSW10)

The one-inch plate beamsplitter from Thorlabs meets a lot of our component requirements for the beamsplitter component. The beamsplitter has a one-inch clear aperture with housing that fits in our dimensional constraints.

### 3.1.6.7 Plate Beamsplitter, 45:55 R:T (BSW41-532)

The one-inch pellicle beamsplitter from Thorlabs meets a lot of our component requirements for the beamsplitter component. The beamsplitter has a one-inch clear aperture with housing that fits in our dimensional constraints. This beam splitter is designed to be used with ND:YAG lasers, making it have better performance but at a narrower working spectra.

### 3.1.6.8 Beamsplitter Product Selection

Table 3.1.11 Beamsplitter Product Selection

Beamsplitter Type	Split Ratio (R:T)	Wavelength	Integration	Cost
BSW10	50:50 at 525 nm	400 nm – 700 nm	Low Effort	\$122.03`
BSW41-532	50:50 at 532 nm	532 nm	Low Effort	\$373.85
Beamsplitter Type	Split Ratio (R:T)	Wavelength	Integration	Cost

We chose BSW10 for its high performance across the visible spectra and lower cost.

### 3.1.7 Opaque Surface

One fair consideration for our darkfield imaging system is the type of surface we are both imaging and detecting the presence of contaminants on. Depending on factors like surface roughness, reflectivity, contrast with the particle/contaminant we are measuring, and whether/ how much the surface is polished, our imaging system can produce varying qualities of image. We considered multiple types of surfaces for different reasons and attempted to find surfaces that would maximize the quality of the image. Our sponsor

specified that the selected surface should be reflective, but the actual choice of the surface was left to us.

### **3.1.7.1 Stainless Steel**

The first promising candidate for the realization of our darkfield contamination detection imaging system is stainless steel. Stainless steel is an alloy of iron that contains mostly iron (around 90%), chromium, carbon, and small amounts of other metals. Stainless steel is very durable and relevant in many lab and non-lab settings. For example, stainless steel is commonly used for surgical tools, pocketknives, kitchenware, and industrial manufacturing applications.

The reflectivity and usability of stainless steel greatly depend on the roughness of the surface. If you polish the surface of stainless steel, you can achieve reflectivity near 90%. If you choose to not polish the surface of the stainless steel or have stainless steel that has micro textures on it, the reflectivity becomes significantly lower, and the contrast of a contamination particle could lower, depending on the pattern. In addition, with the presence of micro textures on the metal the surface will reflect less light in a specular manner and more light in a diffuse manner. This potentially reduces bright spots, which was a consideration for the quality of our final image. In turn, the contrast between some types of particles and the background surface is lower.

Other than simply selecting a metal surface with high reflection, low surface roughness, and largely specular reflections, stainless steel is a very common material that is found in many situations. Using stainless steel as our objective surface allows us to test our system against a popular material type, expanding the use cases for our system. Stainless steel is also a widely available material. It is available to be purchased in relatively small amounts cheaply. Although, finding polished stainless steel is more of a challenge.

### **3.1.7.2 Aluminum**

Another promising candidate for the realization of our darkfield contamination detection imaging system is aluminum. Aluminum is a lightweight material that has high reflectivity across the visible spectrum, especially when properly polished. Compared to stainless steel, Aluminum has a higher reflectivity and lower surface roughness. Aluminum, like stainless steel, has largely specular reflections with a much smaller number of diffuse reflections in comparison.

When considering our reflective darkfield imaging system, aluminum has a strong reflectivity at the color light we are operating at. As a tradeoff though, the aluminum might be too reflective and cause strong glints if illumination isn't carefully managed. A fair benefit of using aluminum, other than it being very common to come across, is its ability to be purchased cheaply. I found more options for purchasing aluminum than I did searching for stainless steel.

If the polished aluminum proves to give more specular reflection than intended, It is always an option to acquire brushed aluminum, which has more surface roughness and more diffuse reflections than its polished counterpart. Because aluminum is a softer metal than something like stainless steel, aluminum plates are more prone to scratches and other

small damage. This could ultimately manifest itself as bright spots and scattering which will reduce our final image quality.

### **3.1.7.3 Silicon Wafer**

Silicon wafers are a strong contender for the reflective object surface. Since we are being sponsored by a semiconductor manufacturing company, it makes sense to test out our system on a material they manufacture. Polished silicon wafers, when compared to both stainless steel and aluminum, have a higher reflectivity. Polished silicon wafers have a higher specular reflection and lower diffuse reflectivity than aluminum.

Silicon wafers are far more expensive than a small square of stainless steel or aluminum. The cheapest single polished silicon wafer I could find (that was not on eBay) was going for \$13 but you needed to buy 25 of them. If I continued to dig, I could find someone or a company willing to send me one for cheaper, but this isn't really "off the shelf" like the sponsors of our project specified.

One of the largest advantages to using silicon wafers instead of another metal is that the surface contributes to very little scattering. Standard precision polishing is sub nanometer for some silicon wafers. Also, since the reflectivity and surface are very uniform throughout the entire silicon wafer, they serve as a good baseline and cause the potential contaminants on the surface of the wafers to have a strong contrast with the background.

In comparison to the strong stainless steel, and the slightly less strong aluminum, silicon wafers are very fragile. These wafers are far more susceptible to cracking or chipping than metal surfaces. While technically higher performance, reflectivity wise, and potentially more optimal for our configuration, silicon wafers are more difficult to buy singularly and with the difference in price being a multiple of the price of stainless steel or aluminum plates, the difference in performance is not worth the extra money spent.

### **3.1.7.4 Copper**

Copper plates were another consideration for a potential objective surface. While known for its thermal and electrical conductivity, when polished, copper can produce fairly high reflectivity in the visible spectrum. A polished copper plate can approach similar reflectivity to stainless steel, although not at the levels of aluminum and silicon wafers. Copper is a fairly malleable metal, like aluminum, and as a result a copper plate is prone to scratches and other artifacts which might affect our ability to detect contamination on its surface. Because darkfield imaging relies on controlling the amount of light that is scattered from a particular surface.

I wanted to consider another metal that wasn't a bright color as to potentially increase contrast and image quality, but it seems unlikely that copper will serve that purpose. In addition to the malleability of copper, copper is prone to oxidation. This means that after enough time the surface of the metal sheet would grow black and lose some of its reflectivity. It is possible to keep copper polished and restore it to a higher reflectivity state. Although, the fact that we must maintain the copper to keep consistent imaging makes it a less-than-ideal choice. Even worse than just being susceptible to oxidation, the layer of oxidation that forms is not uniform. In addition, copper is not a common material used for

applications such as these, so the amount of application for checking the contaminations on the surface of that material is low.

Another aspect which makes copper not our optimal choice is the price. The price of copper is somewhere between the cost of the aluminum and silicon wafers. While not nearly as expensive as silicon wafers. The slight cost of maintenance keeps me from choosing it. Another factor is the color of the copper itself. That natural red hue will reduce the contrast between red or brown contaminants.

**Table 3.1.12 Opaque Surface Selection**

Factor	Stainless Steel	Aluminum	Silicon Wafer	Copper
Reflectivity	High (polished)	Very high	Extremely high	Moderate-high
Surface Roughness	Medium-low	Low	Very low	Low (polished)
Specular Reflection	Moderate-high	High	Very high	High (fresh)
Diffuse Scattering	Medium	Low	Very low	Increases with oxidation
Durability	Very durable	Soft, scratches easily	Fragile	Soft, scratches
Cost	Low-Medium	Low	High	Medium
Availability	High	Very high	Limited	Moderate
Maintenance	Low	Low	Low	High
Color / Contrast	Neutral gray	Bright silver	Dark gray	Red-orange

Aluminum was chosen as the preferred surface for our dark-field contamination detection imaging system for multiple reasons. First off, aluminum offers high reflectivity across the visible spectrum. While the reflectivity is not as high as a polished silicon wafer, the difference in reflectivity is not so great as to make aluminum not suitable. Another reason aluminum is a solid choice is because it has a fair number of specular reflections when compared to diffuse reflections. While this quality of aluminum has the potential to cause bright spots, which can affect our final image quality, aluminum still seems like the sensible choice. Aluminum is the middle ground option when it comes to specular reflections.

A large reason why aluminum seemed to be the most appealing option is the low cost of the material in addition to the lack of maintenance, when compared to copper. While copper will oxidize given enough time, reducing the reflectivity of the surface, aluminum

will also tarnish but the difference in reflectivity is not great enough to dismiss the option of aluminum. Aluminum is also the cheapest option out of the four, with silicon being the most expensive, and copper and stainless steel being middle-tier options. In addition to the cost, Aluminum is also the most widely available on the internet. Since aluminum is malleable it is susceptible to scratches and other wear, but due the low cost of aluminum, getting a replacement plate if the one we purchase gets too damaged is no problem.

It is also worth noting that considering the relatively inexpensive cost of the objective surface materials it will be possible and even likely we will test out our system on another one of these materials in addition to the aluminum

### **3.1.8 Particle Selection**

One key consideration for our dark-field contaminant inspection imaging system is the type of particles or contaminants we will be inspecting. With the specification given to us by ASML, the particle size we are looking to inspect is around 200 microns by 50 microns. In this section we compare multiple different types of potential particles/contaminants to potentially be analyzed by our imaging system. These particles will be placed on our object surface and will scatter light allowing us to increase contrast at the edges of these contaminants. By analyzing how different contaminants behave under these conditions, we can determine the most effective particle for proving the viability of our imaging system

#### **3.1.8.1 Cotton Fiber Size**

One fair selection for a potential contaminant is cotton fibers. Cotton fiber can naturally vary in length from around 5 mm to 12mm, although this can be shortened using a sharp precision object for testing. Cotton fibers can also vary in diameter from around 12 micron to 20 micron. Cotton fibers have diffuse scattering as well as low reflectivity. The low reflectivity mentioned can cause a fairly significant issue when trying to image cotton fibers with a darkfield microscopy imaging system. The edges will not scatter light very efficiently and as a result the contrast for these particles will be low. Cotton fibers can be used to show the effectiveness of our system for low contrast edges.

On another hand, purchasing small amounts of fibers is not too expensive. If you make a error cutting the fiber to the desired length, it is always reasonable to go purchase more fibers and try again.

#### **3.1.8.2 Human Hair**

Another reasonable selection of potential contaminants is human hair. Human hair typically ranges from around 50  $\mu\text{m}$  – 300  $\mu\text{m}$  depending on the person. Human hair has a smooth surface and primarily has diffuse scattering. The reflectivity of a human hair, when compared to cellulose and cotton, is higher. A major benefit of using human hair instead of another contaminant is that we have hair in abundance and it serves as a quick contamination source that allows us to check whether our system works.

#### **3.1.8.3 Cellulose Fiber**

Cellulose fibers were also considered as a possible contaminant. This kind of fiber is commonly found in paper products, packaging materials, and cleaning wipes. Cellulose

fibers range from 10 micron to 20 micron. This contaminant. only weapons that range from a 0.5 mm to 3 mm. Like other organic fibers, cellulose has low reflectivity and produces primarily diffuse scattering, which can make its edges more challenging to detect in a dark-field configuration.

**Table 3.1.13 Particle Selection**

Criterion	Cotton Fiber	Human Hair	Cellulose Fiber
Typical Length	5–12 mm (cut to test size)	50–100 mm (cut to test size)	0.5–3 mm (can be cut)
Diameter	12–20 $\mu\text{m}$	50–100 $\mu\text{m}$	10–20 $\mu\text{m}$
Reflectivity	Very low	Low–moderate	Very low
Scattering Type	Mostly diffuse	Mostly diffuse	Diffuse
Surface Texture	Twisted, matte	Smooth, cylindrical	Irregular, fibrous
Relevance as Contaminant	Very common textile contaminant	Common biological/ environmental contaminant	Common paper/packageing contaminant
Why Useful for Testing	Soft diffuse signature	Thick, high-contrast fiber	Very small, low-contrast fiber

Human hair was selected as the preferred contaminant for evaluating our dark-field imaging system for several reasons. Hair offers a relatively consistent and easily measurable diameter, typically between 50 and 100 microns, which places it close to the 50  $\mu\text{m}$  – 300  $\mu\text{m}$  size specification provided by ASML. Its smooth, cylindrical structure produces predictable scattering behavior that allows the system to generate well-defined edges without the extremely low reflectivity issues seen in cellulose-based fibers. Human hair is also rigid enough to maintain its shape when placed on the reflective aluminum surface, reducing variability in how it appears under illumination. In addition, it is inexpensive, easy to source, and simple to trim to the required dimensions, making it a practical and reproducible contaminant choice. Using human hair as our benchmark contaminant allows us to reliably test detectability, contrast, and edge clarity across repeated imaging trials.

### 3.1.9 Camera Adjuster Technology

Through consultation with the team, sponsor, and project advisors, it was determined that the system may require a mechanism to finely adjust the position of either the object platform or the camera module. This adjustment will allow the user to bring the sample into sharp focus by moving the camera or stage a few millimeters along the optical axis.

The adjustment mechanism will be designed to provide precise, controlled motion either as a series of small incremental steps or as a smooth continuous movement. This fine control is essential to compensate for variations in component tolerances and to achieve optimal focus across different samples or surface heights.

Potential implementations include a manual micrometer screw, Grashof crank rocker, series of gears and a rack, or a spiral gear and a rack.

### **3.1.9.1 Manual Micrometer Screw**

A manual micrometer screw provides extremely precise and controlled motion, allowing the camera or object platform to be adjusted in very small, repeatable increments. This precision ensures that the optical system can achieve accurate focus without overshooting the desired position. The micrometer screw can be easily integrated into the housing, making it a practical option for fine mechanical adjustment. Its compact form factor also allows for a clean design that maintains the overall portability of the device.

The primary drawback of this approach is cost, as high quality micrometer stages and screws can be relatively expensive compared to simpler adjustment mechanisms. However, the precision and reliability they offer may justify the added expense, especially for ensuring consistent image quality and accurate optical alignment.

### **3.1.9.2 Grashof Crank Rocker or Regular Crank Rocker**

A Grashof crank rocker, or a standard crank rocker mechanism, offers a relatively simple and cost effective solution for achieving smooth motion. The linkage can be custom designed in CAD to meet the project's specific requirements, and 3D printing can be used to fabricate and integrate the components directly into the housing. The mechanism provides continuous and consistent motion when driven manually or by a small motor, which could enable smooth camera or platform adjustments.

However, one of the main challenges with this design is that it produces rotational motion, making it difficult though not impossible to achieve purely translational movement of the camera. Additionally, it can be challenging to precisely measure or control the exact displacement of the camera using this mechanism alone. As a result, while the crank rocker design is appealing for its simplicity and low cost, it may not provide the fine positional accuracy required for precise optical focusing.

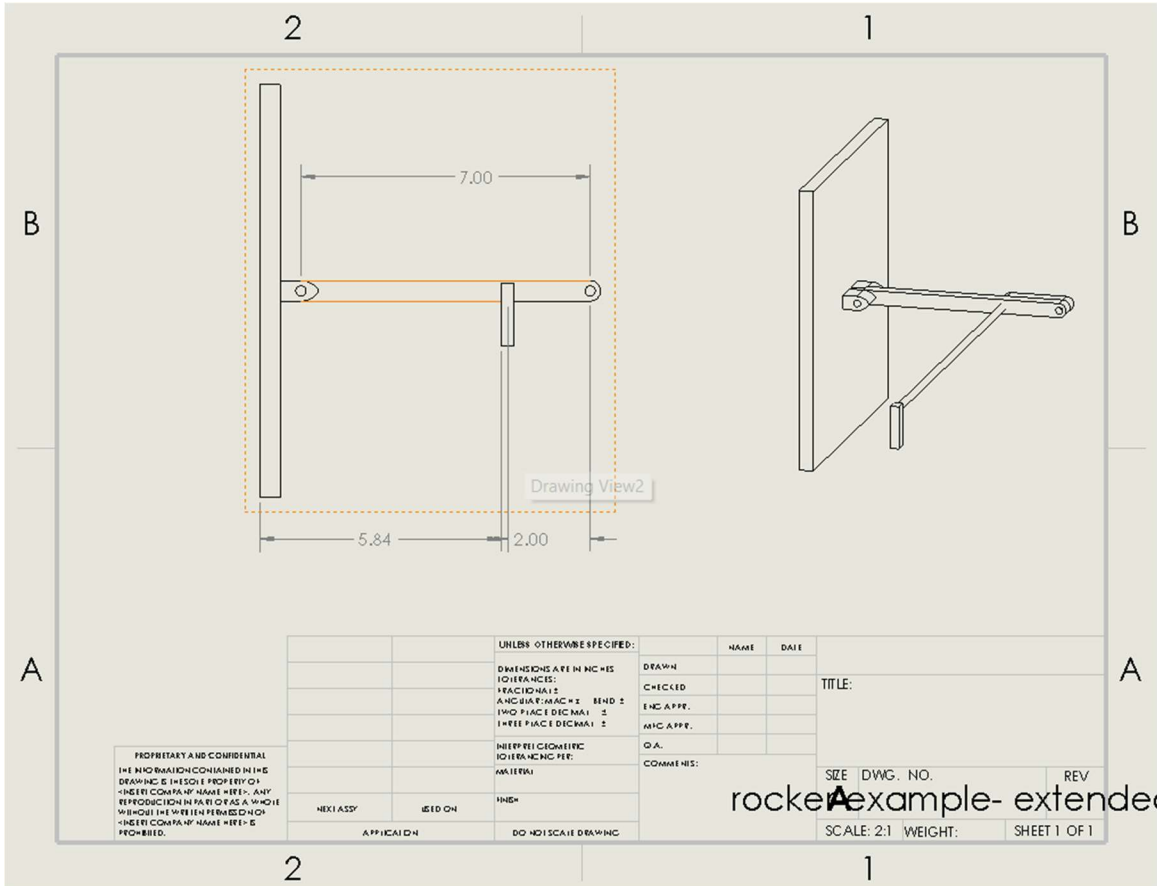


Figure 3.1.1 Crank Rocker Compressed

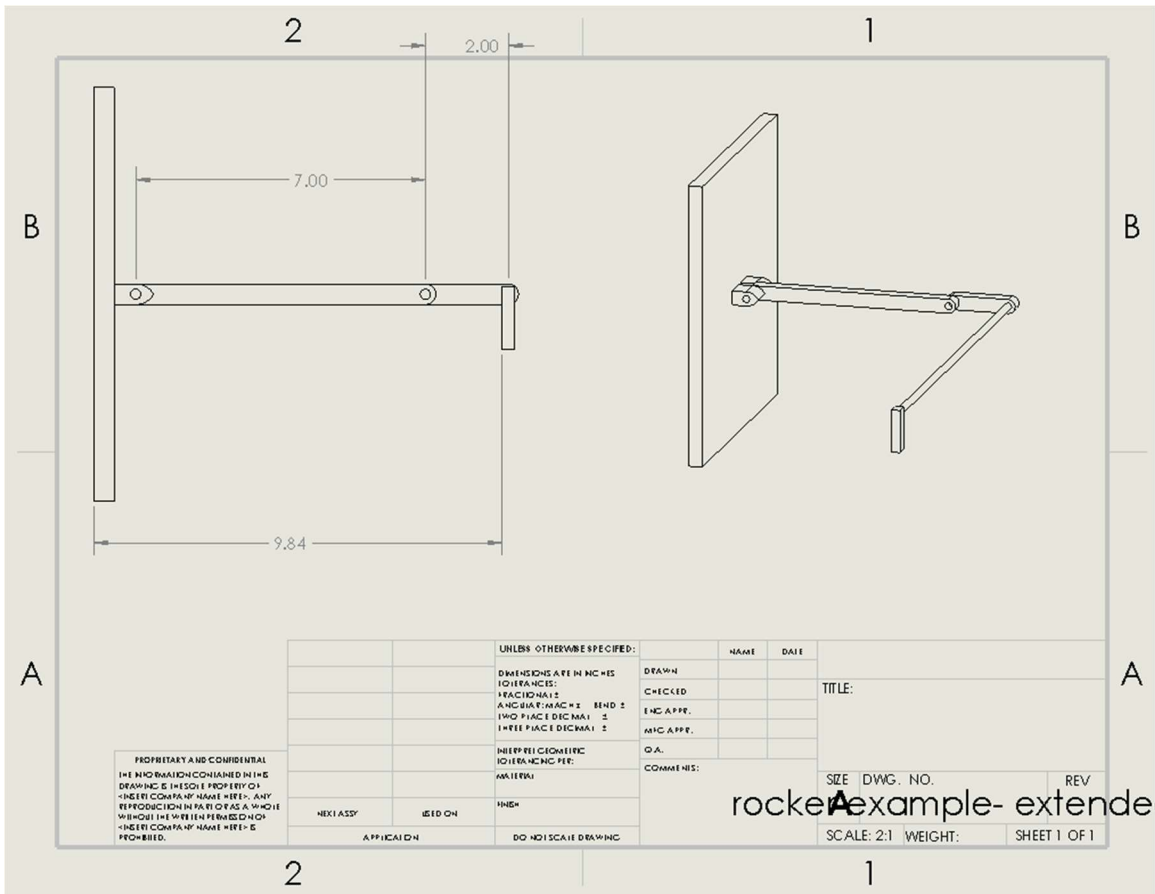


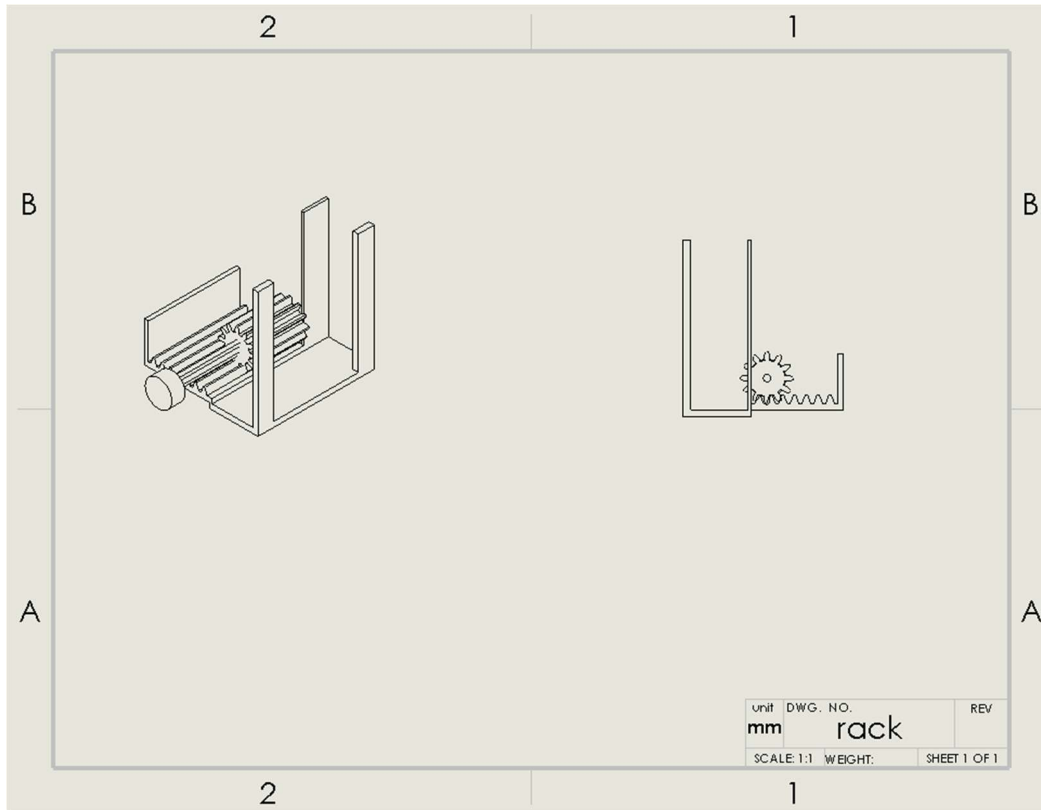
Figure 3.1.2 Crank Rocker Expanded

This figure presents a conceptual design of a crank rocker mechanism. The camera would be mounted on the flat platform, and rotating the crank would allow for approximately 4 mm of linear adjustment along the desired axis. The total range of motion can be increased by modifying the link lengths, though this adjustment would require additional space within the housing. This mechanism demonstrates a compact and mechanically simple approach to achieving smooth and repeatable camera positioning.

### 3.1.9.3 Series of Gears and a Rack

This option involves using a small set of gears connected to a rack attached to the camera or platform. Rotating the gears would translate the rack linearly, allowing for the fine and precise movements required to bring the camera into focus. The mechanism is straightforward to design and can be easily integrated into the housing.

A rack and pinion setup offers a good balance between precision and simplicity, providing smooth translational motion with minimal backlash when properly designed. It also enables manual or motor driven control, depending on the needs of the final prototype. The main drawback of this approach is that the gear components must be manufactured or purchased with very fine teeth to achieve the necessary resolution for small adjustments.



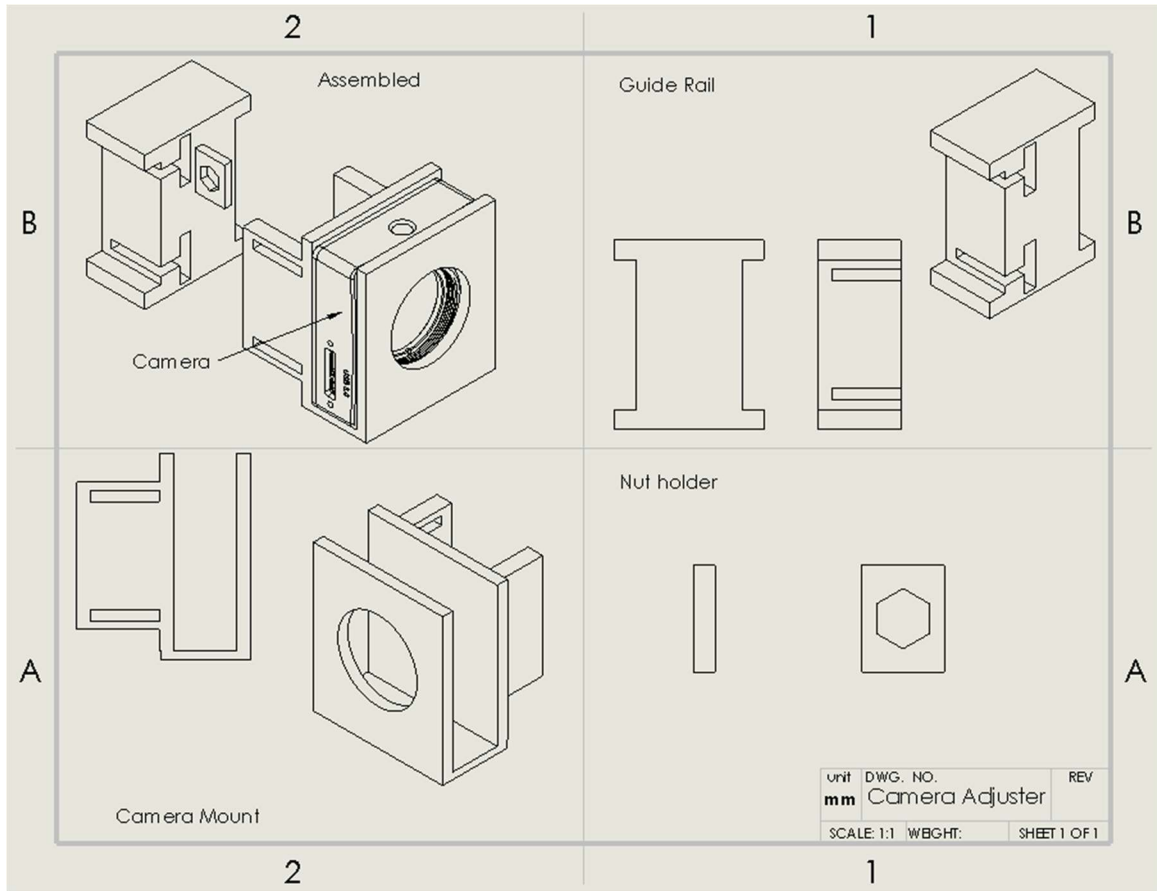
**Figure 3.1.3 Gear and Rack**

This figure illustrates the pinion gear and rack mechanism designed to provide precise linear adjustment of the camera. The camera would be mounted to the flat platform located at the end of the rack. The total adjustment distance is determined by the length of the rack, teeth size, as well as the diameter of the pinion gear. This setup allows for smooth, controlled translational motion and can be easily integrated into the housing, offering a compact and reliable method for fine camera positioning.

### **3.1.9.4 Rail and slider**

This option uses a rail and slider mechanism to guide the camera in a straight linear path while maintaining alignment during motion. The camera would be mounted directly to the sliding platform, which moves along fixed rails integrated into the housing. Motion can be adjusted manually using a lead screw, knob, or small motor depending on the final design requirements. This mechanism provides smooth and controlled translational movement, making it well suited for fine focusing applications.

A rail and slider setup offers a strong balance between simplicity, stability, and precision. The guided rails help prevent tilting or misalignment of the camera during adjustment, which is important for maintaining optical accuracy. It is also compact and can be custom designed to fit within the housing dimensions. The main drawback is that the rails must be manufactured or printed accurately to reduce friction and ensure smooth motion over the full adjustment range.



**Figure 3.1.4 Rail and Slider**

This figure illustrates the rail and slider mechanism designed for precise camera positioning. The camera would be mounted to the sliding platform, which travels along the fixed guide rails. The total adjustment distance depends on the rail length and housing space available. This design provides smooth, repeatable linear motion and reliable camera alignment, making it an effective solution for fine focus adjustment.

### 3.1.9.5 Camera Adjuster Technology Comparison

**Table 3.1.14 Camera Adjuster Technology Comparison**

Mechanism	Micrometer	Crank rocker	Series of gears and rack	Rail & slider
Cost	Relatively expensive	Minimal cost	Low cost	Low cost
Complexity	Simple construction	Moderately simple	Very simple	Moderately simple

Customizability	Limited	Highly customizable	Moderately customizable	Moderately customizable
Size	Large	Customizable but low	Customizable low	Customizable low
Availability	High	Custom Fabrication	Medium	Custom Fabrication
Accuracy	Very Precise	Dependent on Design Parameters	Sufficient for Application	Sufficient for Application
Smooth motion	Excellent	Acceptable	Consistent	Consistent

Based on all these criteria, the rail and slider mechanism will most likely be the first choice for this component, as it offers high customizability and can be easily adapted to fit within the limited space of the housing. If this design proves unsuitable during prototyping or testing, the series of gears and rack will serve as the secondary option, since it also allows for a degree of customization, though selecting appropriately sized gears may present additional challenges. The micrometer screw is considered the final alternative, primarily due to its higher cost, despite offering superior precision and smooth motion.

### 3.1.10 Optical Mounting Methods

Optical components are extremely sensitive to misalignment and must be positioned with high precision to maintain image quality and system performance. Any slight deviation in the alignment of lenses, mirrors, or beam splitters can significantly affect the optical path and result in image distortion or loss of focus. Therefore, careful consideration must be given to how these components are mounted within the housing.

Two primary approaches are being considered for securing the optical components:

- Designing custom mounts directly into the optical housing, or to fit directly into the optical housing
- Purchasing precision optical mounts from suppliers such as Thorlabs and integrating them into the design.

#### 3.1.10.1 Integrated Mounting Design

In this approach, the optical mounts would be designed directly into the 3D printed optical housing. This method allows for full customization of the component layout and reduces part count by eliminating separate mounting hardware. It also ensures that the geometry of each component seat is optimized for our specific optical configuration. However, this method requires extremely accurate design and 3D printing tolerances to

ensure proper alignment. Even small deviations in print accuracy could introduce angular or positional errors that affect optical performance.

### **3.1.10.2 Commercial Optical Mounts**

Alternatively, precision optical mounts can be purchased from suppliers such as Thorlabs, Edmund Optics, or Newport. These mounts are manufactured to with high tolerances and provide fine adjustment capabilities for both position and angle. Integrating such mounts into the design would ensure stable, repeatable alignment and long-term mechanical reliability of the optical system.

However, this approach introduces several challenges. The use of commercial mounts increases overall cost and can complicate the integration process, as additional design features may be required to secure the mounts within the custom housing. Additionally with our tight-sized constants some standard optical mounts may not fit within our system, potentially limiting their use in certain sections.

In summary, integrated mounts offer a lightweight and cost-effective solution suitable for a prototype, while commercial mounts provide superior precision and reliability. The final choice may involve a hybrid approach using integrated mounts where tolerances are less critical and commercial mounts for key alignment sensitive components such as the beam splitter.

### **3.1.11 Filament Types**

Different 3D printing filaments provide unique mechanical, thermal, and surface properties, making them suitable for specific components of the system. For this project, all parts will be printed using black filament to minimize internal reflections and light leakage that could interfere with optical performance. The primary materials being considered are PLA CF, PLA, ABS, and TPU.

#### **3.1.11.1 PLA CF (Carbon Fiber Reinforced PLA)**

PLA CF would be best used for the optical housings, where rigidity, dimensional stability, and light absorption are most critical. The carbon fibers enhance stiffness and reduce thermal expansion, helping the housing maintain alignment between optical components. The material's matte black finish also minimizes internal reflections, improving image contrast and accuracy. The main disadvantages of PLA CF are it has a higher cost and increased wear on the printer nozzle due to its abrasive nature.

#### **3.1.11.2 PLA (Polylactic Acid)**

Standard PLA will likely be used for the outer casing and noncritical structural components. It is cost effective, easy to print, and produces high quality surface finishes. While it lacks the heat and impact resistance of PLA CF or ABS, it is sufficient for the external enclosure, which experiences minimal stress and temperature variation.

#### **3.1.11.3 ABS (Acrylonitrile Butadiene Styrene)**

ABS offers excellent durability and higher temperature resistance than PLA, making it a viable alternative for any parts exposed to heat from the electronics or LED. However,

ABS can be more challenging to print due to warping and the need for an enclosed build area. It may be used selectively if added thermal robustness is required.

### 3.1.11.4 TPU (Thermoplastic Polyurethane)

TPU is a flexible filament that could be used for vibration damping mounts or protective pads. Its elasticity can help isolate sensitive optical components from mechanical vibrations without adding much complexity to the design.

### 3.1.11.5 Filament Type Selection

**Table 3.1.15 Filament Properties**

Filament Type	Strength	Density	Stiffness	Heat Resistance	Toughness	Adhesion
PLA	76MPa	1.20–1.30 g/cm <sup>3</sup>	2750MPa	57°c	26.6kJ/m <sup>2</sup>	13.8kJ/m <sup>2</sup>
PLA CF	89MPa	1.30–1.50 g/cm <sup>3</sup>	3950MPa	55°c	23.2 kJ/m <sup>2</sup>	7.8kJ/m <sup>2</sup>
ABS	62MPa	1.01–1.10 g/cm <sup>3</sup>	1880MPa	87°c	39.3 kJ/m <sup>2</sup>	7.4kJ/m <sup>2</sup>
TPU	N/A	1.10–1.25 g/cm <sup>3</sup>	N/A	69°c	124.2 kJ/m <sup>2</sup>	8.73kJ/m <sup>2</sup>

In summary, PLA CF will most likely be used for the optical housings to ensure high precision, dimensional stability, and effective light isolation. PLA will be used for the external casing to reduce material costs and simplify manufacturing, while ABS and TPU will be incorporated as needed for components requiring added thermal resistance or flexibility.

### 3.1.12 Processor Technology

The processor we decide to use is extremely important, as one of our constraints is how long it takes the for image-processing algorithm to run. Furthermore, with PCB requirements waived for our group, it means that we have access to high-power pre-made computing technology. This gives us the option to consider technologies more advanced than expected. The processor is one of the most important parts of our project since it involves image capture and analysis, and it is the heart of the electronic subsystem. This means that the processor will directly impact performance, latency, and overall system responsiveness.

### **3.1.12.1 Microcontroller Unit (MCU)**

Microcontrollers are traditionally the backbone of embedded systems. They are the parts with which the team has the most experience. Some advantages of MCU's include their power efficiency, low cost, and adaptability, making them ideal for projects that require low power or lower budgets. However, one of their biggest drawbacks is their limited computational resources, which can often create a bottleneck for compute-heavy applications.

For image-processing workloads, MCUs struggle to meet performance demands. Algorithms such as feature detection, filtering, or neural-network inference require high memory bandwidth, something which is outside the capabilities of most microcontrollers. While external DSPs or co-processors can be attached, that increases design complexity and partially defeats the MCU's simplicity. Moreover, MCUs generally lack hardware floating-point units or SIMD (single-instruction multiple-data) extensions powerful enough for heavy pixel-level computation.

Another limitation is software ecosystem maturity for vision applications. Although some MCUs support lightweight ML libraries, these are intended for small models or low-resolution sensor data, unlike the requirements of P.I.D.I. The biggest drawback is clearly the limits in processing and memory found in microcontrollers. This can cause camera interfaces, high-speed I/O, and large image buffers to quickly find the limit of their capabilities. As a result, even optimized implementations experience long processing times that make real-time performance unattainable.

The main advantage of MCUs would be simplicity, low cost, and power efficiency. However, given that our project emphasizes fast image processing and we are not constrained to ultra-low-power design, an MCU is not appropriate. The primary reason is that the performance and flexibility trade-offs would substantially limit the project's capability. Furthermore, we have been provided with a substantial budget, thus making processor selection not an area where we need to focus on cost savings

### **3.1.12.2 Single Board Computer (SBC)**

The second option we looked into was Single-Board Computers (SBCs). As the name indicates, SBCs are entire computers assembled on a single circuit board. This allows for more integrated peripherals than MCUs, such as CPU, memory, storage, and interfaces like HDMI, Ethernet, or USB. SBCs are powerful platforms capable of running full operating systems such as Linux, allowing the use of advanced programming environments (Python, OpenCV, TensorFlow) that simplify algorithm development.

The biggest advantage of SBCs is their use of multi-core, high-frequency application processors. The processors used in SBCs are similar to those used in smartphones and are coupled with GPUs. This means that our processing power is significantly high, and should be able to handle the image processing algorithm without much trouble.

The biggest drawback for SBCs, however, is the fact that integrating them into an embedded project is significantly more challenging due to their larger size, higher power consumption, and heat generation. This present a series of challenges not typically present in MCUs. Despite this drawbacks, SBCs remain a viable option for our projects, as our

main focus is on processing power. For our application, an SBC would easily meet processing demands.

### 3.1.12.3 System-on-Module (SoM) / Compute Module

A System-on-Module (SoM) or Compute Module bridges the gap between MCUs and SBCs. It encapsulates a powerful SoC (System-on-Chip), memory, and sometimes flash storage on a small board that connects to a carrier through standardized high-density connectors. This architecture combines the performance of an SBC with the integration flexibility of embedded hardware. Some examples include the Raspberry Pi Compute Module 5, ArmSoM CM5, and the LattePanda Mu.

The biggest benefit for us in regards to SoMs, is that we can still get the high computing performance while also being able to design a custom carrier board tailored to our peripherals. Essentially, SoMs merge the benefits of MCUs and SBCs by giving us access to high computing power found in SBCs, as well as more freedom of design, akin to MCUs.

The trade-off compared to SBCs is that SoMs often require a carrier board to expose connectors such as USB, HDMI, or Ethernet. This adds minor design effort and sometimes increases cost. However, for our project, since PCB requirements have been waived and we can use pre-made carrier boards, this drawback is insignificant. Therefore, the SoM architecture is the ideal balance between performance, form factor, and expandability.

### 3.1.12.4 Processor Technology Selection

Table 3.1.16 Processor Technology Comparisons

Technology	Computing Power	Integration Effort	Size / Power	Ecosystem	Image-Processing Suitability
MCU	Low	Very Easy	Minimal	Limited	Poor
SBC	Very-High	Moderate	Large / High	Excellent	Excellent
SoM	High	Slightly Easy	Compact / Moderate	Strong	Very Good

Ultimately, we have decided to go with a SoM. The choice was made due to it's performance being only slightly inferior to SBCs, while also providing a lot more freedom of design.

### **3.1.12.5 SoM Product Comparison**

Now that we have decided to go with there are a few different options to evaluate. Our biggest points of consideration are processing power, a need for our algorithm, and the ecosystem, useful for peripheral implementation.

### **3.1.12.6 ArmSoM CM5**

The first SoM we looked at was the ArmSoM CM5. This SoM is powered by a Rockchip RK3576 SoC, which is a chip combining a quad-core Cortex-A72@2.2GHz and a quad-core Cortex-A53@1.8GHz. This makes gives it high processing powers, ideal for the needs of our project. Furthermore, the ArmSoM CM5 can support up to 16 GB LPDDR5 RAM, which creates an environment with high speed flash memory.

In terms of performance, the ArmSoM CM5 has a good balance between efficiency and power. The inclusion of LPDDR5 memory and a 6 TOPS NPU makes it highly competitive for mid-to-high-performance embedded AI applications. Additionally, the small form factor design of the ArmSoM CM5 allows it to slot into compatible carrier boards, something that we will require in this project.

Although there are many advantages to the ArmSoM CM5, one of the biggest things to consider is the ecosystem. ArmSoM is a relatively new brand, so software and documentation are limited compared to more established brands. This brings into question potential issues with integration as well as reliability. Furthermore, kernel support, driver availability, and community resources are still developing. This can slow integration and debugging during development.

Despite these issues, the ArmSoM CM5 remains a strong contender from a purely technical perspective. It offers significant compute capability at a competitive price and supports modern interfaces. Nonetheless, the limited community support and uncertain longevity reduce its overall attractiveness compared to the Raspberry Pi Compute Module 5.

### **3.1.12.7 Raspberry Pi Compute Module 5 (CM5)**

The option that we were most interested in from the beginning was the Raspberry Pi Compute Module 5. This is the latest technology released by Raspberry Pi, and it has been designed specifically for embedded applications. The Raspberry Pi CM5 uses a Broadcom BCM2712 SoC with four ARM Cortex-A76 cores running at up to 2.4 GHz as its processor, while also offering support for LPDDR4X memory in configurations up to 8 GB.

The biggest advantage for the CM5 is its ecosystem. Raspberry Pi has a long standing and very established community. This means that there is extensive software support, documentation, and compatibility with existing tools and libraries. Furthermore, the CM5 supports modern I/O including dual 4Kp60 HDMI, PCIe Gen 3, MIPI CSI for cameras, and USB 3.0. This is crucial for our project, as our peripheral components require these types of connections. Additionally, it still remains in a small form factor, and keeps its power consumption moderate.

From a design standpoint, the Compute Module ecosystem is ideal for embedded integration. There are countless peripherals and carrier boards from numerous

manufacturers that are compatible with the Raspberry Pi ecosystem as a whole. These would allow us to rapidly prototype, while also maintaining performance.

Given its high performance, broad software support, and reliability of supply, the Raspberry Pi Compute Module 5 is the most balanced choice for our project. It offers ample processing capability for image processing algorithms while minimizing integration risk.

### 3.1.12.8 LattePanda Mu Compute Module

Another strong option for our project was the LattePanda Mu Compute Module. It has high computing power as well as offering strong I/O support necessary for our components. Furthermore, it utilizes x86 architecture, which might be helpful if required by our overall system’s environment.

Even though performance wise the LattePanda Mu Compute Module is comparable to the other options, it does come with significant trade-offs. Firstly, it has a less friendly environment than the Raspberry Pi. Furthermore, power consumption, heat dissipation, and integration details might be higher/harder compared with a compact ARM-SoM.

Ultimately, although it appears to be a good alternative, there are trade-offs to consider with this technology.

### 3.1.12.9 SoM Product Selection

**Table 3.1.17 SoM Products Comparisons**

Technology	CPU	Memory	Cost	Ecosystem	Image-Processing Suitability
ArmSoM CM5	Rockchip RK3576: 4× Cortex-A72 @ 2.2 GHz + 4× Cortex-A53 @ 1.8 GHz	Up to 16 GB LPDDR5	Moderate/High	Limited	Good
LattePanda Mu Compute Module	Intel N100 (or N305) x86; quad-core up to 3.4 GHz (N100)	8–16 GB LPDDR5	Moderate/High	Limited	Good
Raspberry Pi CM5	Broadcom BCM2712, 4× Cortex-A76 @ up to 2.4 GHz (Armv8)	LPDDR4X 2/4/8/16 GB	Moderate	Strong	Good

Our choice was to originally go with the Raspberry Pi Compute Module 5. The reasoning was its lower price point, extensive ecosystem, and familiarity with the brand offered plenty of flexibility for our design. However, we encountered a compatibility problem between the Raspberry Pi CM5's ARM architecture chip and the proprietary software from Thorlabs for camera control.

Given those issues, the LattePanda Mu was then selected as our new SoM. The x86 chip architecture was compatible with Thorlabs' code. This also gave us the benefit of running either Windows or Linux as our operating system, whereas with the Raspberry Pi we could only run Linux based OSs.

### **3.1.13 I/O Carrier Board**

The off-the-shelf component requirement for our project means that we have to buy an IO carrier board the LattePanda Mu. This will allow us to have access to I/O ports which will be necessary for peripherals as well as the camera itself. Fortunately, LattePanda offers two separate options for the LattePanda Mu.

#### **3.1.13.1 (DFR1142) Lite Carrier Board**

The (DFR1142) Lite Carrier Board is the lower end model of carrier boards offered by LattePanda. However, it still offers a comprehensive development platform and full I/O functionally necessary for our project.

The Lite Carrier Board includes two USB 3.2 and two USB 2.0 ports, as well as an HDMI port. This would be more than enough ports for our peripherals. Additionally, the Lite Carrier Board offers two options for power supply, a USB Type-C port rated for 15V with a max of 3A, and a DC 5.5x2.5mm port rated for 12-20V with a max of 10A.

The Lite Carrier Board also offers a myriad of expansion slots including one PCIe 3.0 x4 expansion slot and Dual M.2 expansion slots. This offers a lot of flexibility when it comes to possible future improvements. The price for the Lite Carrier Board is \$39 without taxes and shipping fees.

#### **3.1.13.2 (DFR1141) Full-Function Evaluation Carrier Board**

The DFR1141 Full-Function Evaluation Carrier Board is an expanded version of the Lite Carrier Board. It offers four USB 3.0 ports, two USB 2.0 ports, a USB Type-C port, as well as two HDMI 2.0 ports.

The DFR1141 also offers more options when it comes to expansion slots, including one PCIe 1.0 x1 slot, two SATA 6Gb/s interfaces, as well as the options offered by the Lite Carrier Board.

Additionally, the DFR1141 offers full access to all the pins from the LattePanda Mu. This includes the GPIO pins, as well as UART and I2C pins. This is a significant advantage over the Lite Carrier Board, as having access to the GPIO pins is very beneficial for the design of our project. The price for the Full-Function Evaluation Carrier Board is \$89 without taxes and shipping fees.

### 3.1.13.3 Carrier Board Product Selection

Table 3.1.18 Carrier Board Products Comparison

Technology	USB Ports	HDMI Ports	GPIO Pin Access	Power Supply Input	Price
Lite Carrier Board	2x USB 3.2 2x USB 2.0	1x HDMI 2.0	NO	USB Type-C: 15V MAX 3A  DC 5.5x2.5mm: 12-20V MAX 10A	Low
Full-Function Evaluation Carrier Board	4x USB 3.0 2x USB 2.0 1x USB Type-C	2x HDMI 2.0	YES	DC 5.5x2.5mm: 12-20V MAX 10A	High

We opted to go with the Lite Carrier Board over the Full-Function Evaluation Carrier Board. Although the access to the GPIO pins was a significant point of consideration for the Full-Function Evaluation Carrier Board, the price difference was also significant, as the Lite Carrier Board is more than half the price. Furthermore, the Lite Carrier Board offers enough I/O ports for our project.

### 3.1.14 Touch Screen Display Selection

A touch screen display was chosen to be a part of the project as it serves multiple functions. Primarily, it offers a display panel necessary for the system, but it also offers input controls through the touch inputs. In order to select the right product, a few points had to be considered.

#### 3.1.14.1 Panel Size Selections

One of the first decisions is selecting the panel size and resolution. A larger panel (10” to 12”) might provide more screen real-estate, but increases cost, physical size/weight, power consumption, and may require more robust mechanical support. A smaller panel (5” to 7”) stays compact and embedded-friendly. Resolution is also key: 1024x600 is a common resolution for 7” panels in the maker space, giving sufficient detail for many UI tasks but not ultra-high definition.

Table 3.1.19 Screen Size Selection

Technology	Resolution	Interface & Power	Cost
Large Panel	1920×1080 or higher	HDMI + power >5 V	High

Medium Panel	1024×600	HDMI + USB + 5 V	Moderate
Small Panel	800×480	HDMI or SPI/DSI	Low

A medium panel was the best compromise as it allows for better viewing of the image than a small panel, while not taking as much space as a large panel. Also, the price difference between medium and small panels not substantial enough to sway the decision.

### 3.1.14.2 Display Technology Selection

The next decision to be made was for the display technology. We looked into the three major technologies available: LED, LCD, and OLED.

#### 3.1.14.3 LED Display

LED (Light Emitting Diode) displays are composed of numerous individual diodes that emit light directly. Since each pixel emits its own light, LED panels offer excellent contrast, brightness, and color depth. Furthermore, LED panels have no need for a backlight, making them more energy efficient. However, LED panels can be more expensive, and they run the risk of suffering burn-in over time if static images are displayed continuously.

#### 3.1.14.4 LCD Display

LCD (Liquid Crystal Display) panels, on the other hand, use a backlight. This backlight is often made up of white LEDs, which then illuminate pixels that modulate light through liquid crystal layers. LCDs are cheaper and perform well under bright lighting conditions, however, they consume slightly more power. IPS (In-Plane Switching) LCDs, in particular, provide wide viewing angles and accurate color reproduction, making them ideal for embedded displays that need consistent visuals.

#### 3.1.14.5 OLED Display

OLED (Organic Light Emitting Diode) displays are similar to LED displays in that each pixel produces its own light. OLED can simply be considered to be an upgraded version of LED technologies. This means that, just as with LED, OLED displays do not require a separate backlight, allowing them to achieve very high contrast ratios, deep black levels, and excellent color quality. OLED panels are also typically thinner and lighter than LCD panels, which can be useful in compact embedded systems. Additionally, OLED displays often have fast response times, making them suitable for smooth user interfaces. However, many of the same drawbacks from LED displays remain, while also increasing cost.

**Table 3.1.20 Display Technology Selection**

Technology	Image Quality	Durability	Suitability	Cost
------------	---------------	------------	-------------	------

LED	Very high contrast, deep blacks	Can suffer burn-in	Premium, high-end visuals	High
LCD	Excellent color accuracy, slightly lower contrast	No burn-in, longer lifespan	Reliable, cost-effective embedded use	Moderate
OLED	Highest contrast, deepest blacks	More susceptible to burn-in	Top of the market visuals	Very High

For our project, an IPS LCD display was chosen. It offers reliable image quality, stable color reproduction, and affordability. The consistent performance of LCD panels under varying lighting and temperature conditions makes them ideal for embedded systems that prioritize functionality and readability over absolute visual contrast.

### 3.1.14.6 Touch Input Technologies

The next decision is touch input type, which comes down to two technologies, capacitive (multi-touch, more modern, smooth) vs resistive (cheaper, may support stylus). For interactive systems, capacitive is preferred since it gives more natural touch experience and supports multi-finger gestures, though it may raise cost slightly.

Resistive touchscreens detect input through pressure. They consist of multiple flexible layers separated by a small air gap. When touched, the layers make contact, and the system registers a voltage change at that point. This technology works with any object, and is resistant to electrical noise. However, it supports only single-touch input, tends to have lower optical clarity due to extra layers, and wears out faster after repeated use.

Capacitive touchscreens rely on the electrical properties of the human body to sense touch. When a conductive object, like a finger, alters the local electrostatic field on the surface, the system calculates the touch position. Capacitive screens support multi-touch gestures, provide excellent image clarity, and have smooth, glass-like surfaces. Their main drawbacks are higher cost and reduced responsiveness with gloved hands or non-conductive materials.

**Table 3.1.21 Touch Input Technologies Comparison**

Technology	Input Type	Multi-Touch Support	Clarity	Durability	Cost
Resistive	Pressure-based	No	Moderate	Moderate	Lower
Capacitive	Conductive	Yes	Excellent	High	Slightly High

### 3.1.14.7 7” IPS Touch Screen Panel 1024x600 HDMI

Specifically, the module from BuyDisplay (ER-TFT070A2-6-5782) is a 7” IPS panel with resolution 1024x600, HDMI driver board, optional touch panel. Features include full viewing angle, high brightness, HDMI input. Because it uses generic driver board, this type of screen gives good flexibility and is often lower cost. A possible drawback: touch interface may require additional USB driver, documentation may be less polished, and mechanical support (mounting holes, enclosure) may be minimal. For our project this screen is quite capable, but we should verify that the touch-controller is plug-and-play in our OS and that the module includes a stable touch driver board.

### 3.1.14.8 Waveshare 7” HDMI LCD IPS 1024x600 Touch Screen

The Waveshare version offers 7” IPS, resolution 1024x600, HDMI input, capacitive touch, back-light control. The advantages: well-known brand, good documentation (Waveshare), community usage, stable driver support. The form factor and interface are ideal (HDMI + USB touch). For our project, this is a very strong candidate: good brand support, robust community, known compatibility with Raspberry Pi and other SBCs.

### 3.1.14.9 7” HDMI/USB Capacitive Touch Screen Monitor for Raspberry Pi / PC

This option (for example from 52Pi, SKU EP-0177) offers 7” IPS 1024x600, 60 Hz refresh, HDMI input, capacitive touch, driver-free (USB) for Raspberry Pi/PC. This kind of monitor is built more like a small monitor (with frame/stand) and is very plug-and-play. The benefits: very easy setup, includes speakers in some variants, well-targeted for Raspberry Pi. The trade-off: it may be bulkier/integrated for a portable or embedded custom enclosure, and mechanical integration (mounting holes) may be less specialized for production use. For our project, if we’re planning an enclosure and mounting, we might prefer a board-only version rather than a monitor style.

### 3.1.14.10 Touch Screen Display Part Selection

Table 3.1.22 Touch Screen Product Comparisons

Technology	Brand / SKU	Pros	Cons
BuyDisplay 7” IPS 1024x600 HDMI	Generic / BuyDisplay	Lower cost; flexible module	Possibly minimal mounting support; touch driver may require more setup
Waveshare 7” HDMI LCD IPS 1024x600 Touch	Waveshare	Strong brand; known SoM compatibility; good documentation	Slightly higher cost; may need manual configuration tweaks

52Pi / iPistBit 7” 1024x600 HDMI Touch	52Pi / iPistBit	Ready for SoM; includes frame and stand	Possibly less optimized for custom embedded enclosure; may add size/mass
--	-----------------	---	---

After comparing our options, we decided to go with the Waveshare 7” HDMI LCD IPS 1024x600 Touch. Despite its potential higher cost, we know we are getting a reliable component that is compatible with the rest of our peripherals.

### 3.1.15 PSU

**Table 3.1.23 Power Supply Requirements**

Component	Working Voltage (DC)	Working Current	Working Power
LattePanda Mu	9-20 V	3 A	Minimum 10W  (Minimum 45W for Lite Carrier Board)
Waveshare 7” HDMI LCD IPS1024x600 Touch	5 V	2.5-3 A	~15 W
Arduino Nano ESP32 [ABX00083]	5V	2.5-3A	~15 W
Thorlabs CS165CU	5V	2.5-3A	~15 W
High Power LED Chip 10W Green by CHANZON	9-12 V	840-100mA	10W

The power supply requirements for the components of our project are very straightforward, with the exception of the LED. The LED is crucial to the performance and functionality of our project, thus it is a high power LED that is going to require a driver. Thus, this will be the main point to take into account when selecting our PSU.

#### 3.1.15.1 Battery Technology

Our project is going to consist of two separate power systems. The LattePanda Mu is going to be powered by the AC/DC power adapter that is standard for these types of technologies. Furthermore, the display, camera, and Arduino are all powered through a USB cable. This means that they can all be connected to the Lite Carrier Board and be powered by the standard 5V/3A output from these ports. That means that the LED is going to require its own dedicated power supply. In order to accomplish this, we will look at a few battery technologies available to us.

### **3.1.15.2 Lithium-ion**

Lithium-ion batteries have the highest energy density of any common rechargeable chemistry, typically 180 to 250 Wh/kg, and are used widely in laptops, phones, and drones. They offer long runtimes per weight and can easily deliver high continuous currents. The biggest problem, however, is the safety measures required for them. To safely operate Li-ion batteries, they require protection and balancing circuitry. Integrating Li-ion safely means including a battery management system and charger circuit. These are added costs and time to our project for not much of an advantage. Furthermore, The added complexity of this system would go against the nature of the off-the-shelf component requirements.

Despite the drawbacks, lithium-ion batteries perform well when prioritizing energy efficiency. They are ideal for systems that must run for long periods or where compact size matters. However, the added complexities required for running this type of system make them a less attractive option.

### **3.1.15.3 Nickel-Metal Hydride (NiMH)**

NiMH cells represent a middle ground between older NiCd and modern Li-ion technologies. They are inexpensive and can tolerate modest over-charging and deep-discharge better than Li-ion. A typical AA NiMH cell has a nominal voltage of 1.2 V and capacities between 2000 and 2500 mAh. NiMH chemistry also allows direct charging with simple constant-current chargers, making it easy to manage in a lab environment. They deliver currents of 1-2A continuously without damage.

Another major benefit of NiMH batteries is their predictability. They are very forgiving in testing environments, and even if slightly over-discharged or left unused, they typically recover without significant loss in capacity. Their ability to deliver steady voltage throughout most of the discharge cycle makes them reliable for electronics that need consistent power levels. This reliability, along with the simplicity of charging, makes NiMH a practical option and a leading candidate for the technology we will choose.

### **3.1.15.4 Sealed Lead Acid (SLA)**

Lead-acid batteries provide high surge current and are very rugged but have low energy density (around 30–40 Wh/kg) and are heavy. They are often used in backup or automotive applications but are poorly suited to portable embedded systems where weight and form factor matter. Charging times are long, and efficiency is lower than other chemistries.

Despite these drawbacks, SLA batteries remain popular for applications where cost and durability are more important than weight. They are extremely tolerant to abuse, can operate in a wide temperature range, and are easy to recharge with simple constant-voltage chargers. For stationary test setups, an SLA battery could easily power our circuit without issue. However, its bulk and limited capacity per weight make it an impractical choice for a handheld or mobile device, especially when portability is part of the system's design goal.

### 3.1.15.5 Battery Technology Selection

Table 3.1.24 Battery Technology Selection

Technology	Nominal Cell Voltage	Energy Density (Wh/kg)	Recharge Cycles	Safety/ Complexity	Typical Cost per Wh	Suitability
Li-ion (18650)	3.6–3.7 V	180–250	500–1000	Requires BMS & protection	Moderate	Excellent but complex
NiMH (AA)	1.2 V	60–120	500–800	Very safe & simple	Low	Very Good for moderate load
SLA	2Vper cell	30–40	200–300	Safe but heavy / bulky	Low	poor for portable use

Considering weight, safety, cost, and availability, NiMH emerged as the most practical technology for our project. It can be configured easily into a 9.6 V pack (8 × AA cells), providing sufficient voltage for our LED.

### 3.1.15.6 Battery Product Comparison

After choosing NiMH chemistry, three main AA battery product types were evaluated: Panasonic Eneloop Pro, Amazon Basics High-Capacity NiMH, and Energizer Recharge Power Plus. The criteria considered were capacity, current handling, cycle life, and cost.

#### 3.1.15.7 Panasonic Eneloop Pro (BK-3HCCA)

Eneloop Pro cells are premium NiMH batteries rated at 2500 mAh with low self-discharge (retain ~85% charge after 1 year). They deliver up to 2.5 A continuous discharge without noticeable voltage sag. Cycle life is around 500 charges. They have excellent reliability and consistency between cells, making them ideal when stable voltage under load matters. The main drawback is higher cost, around \$25 for a pack of 8, but for critical systems this reliability is worthwhile.

#### 3.1.15.8 Amazon Basics High-Capacity NiMH AA

These offer 2400 mAh at a lower price point (around \$15 for 8). Their internal resistance is slightly higher than Eneloops, so they show more voltage drop under high load (>1.5 A). Rated for 500 cycles, they're good value for general use. For a project where cost sensitivity matters and loads are 2 A average, they are an acceptable option.

### 3.1.15.9 Energizer Recharge Power Plus AA

Energizer’s rechargeable AA cells are rated at 2000 mAh (min) and are widely available. They handle ~1 A continuous loads well but heat up under heavier currents. Their self-discharge is higher, after 1 year, typically ~70% charge remains. They are affordable and convenient but less ideal for sustained high-current operation.

### 3.1.15.10 Battery Product Selection

**Table 3.1.25 Battery Product Comparisons**

Product	Capacity (mAh)	Max Continuous Current	Cycle Life (cycles)	Price (8 pack)	Self-Discharge (after 1 year)	Suitability
Panasonic Eneloop Pro	2500	≈ 2.5 A	~500	~\$25	~85% remaining	Excellent – high current, stable voltage
Amazon Basics High Capacity	2400	≈ 2 A	~500	~\$15	~80%	Very Good – cost-effective
Energizer Recharge Power Plus	2000	≈ 1 A	~400	~\$20	~70%	Good – for light loads only

We selected 8 × Panasonic Eneloop Pro AA NiMH cells connected in series, producing a nominal 9.6 V output. Their high capacity (2500 mAh) and ability to supply up to 2.5 A continuously satisfy our system’s requirements. When calculating for battery life, this configuration yields roughly:

$$9.6V \times 2.5Ah \approx 24Wh$$

At an average draw of ≈ 10 W the runtime is ~2.4 hours. We also will require a constant-current driver for the LED.

### 3.1.15.11 Constant Current Driver Technologies

Given the requirements of the LED, as well as the PSU we have selected for the LED, a ~900mA constant current driver is required. This driver has to be able to output ~900mA constantly while also providing 9-12V. Furthermore, dimension constraints prevent the addition of a heatsink to the LED, this means that the LED must remain off except for when toggled, to keep it cool. This means that the driver has to be able to be controlled.

All of these conditions made it challenging to find an off-the-shelf driver that would suit our needs.

### 3.1.15.12 LD24AJTA Constant Current LED Driver

The first option we found was the LD24AJTA Constant Current LED Driver. This driver is able to output a constant current between 30mA to 1200mA, adjustable via a resistor. This falls within the range required for the LED. The working current for the driver is 6-25V, which our estimated 9.6V PSU falls inside. Finally, the driver contains a PWM pin, allowing for toggling of the LED.

### 3.1.15.13 Chanzon LED Driver 900mA (Constant Current Output)

The second option we looked at is the driver recommended by the LED manufacturer. This driver outputs the required 900mA, and we have confidence that it works for our LED, as it is made by and recommended by the same manufacturer as the LED. However, this driver works off AC voltage and it is not programmable, thus not making it the ideal fit for our system.

### 3.1.15.14 B0F1K97V61 Constant Current Led Driver

The last option we considered was the B0F1K97V61 Constant Current Led Driver. This driver is more similar to the LD24AJTA, as it works off of DC voltage, and outputs anywhere from 28mA to 1500mA constant current. However, this driver also has multiple drawbacks. Firstly, the driver is not programmable, so the LED will not be able to be toggled directly from the driver. Also, the working voltage is only 3-6V, which is lower than what our batteries produce, and is only able to output 8W, also lower than the expected ~10W required by the LED.

### 3.1.15.15 Constant Current Driver Product Selection

Table 3.1.26 Constant Current Driver Product Comparisons

Product	Working Voltage	Output Current	Output Power	Programmable
LD24AJTA Constant Current LED Driver	DC 6-25 V	30-1200mA	Up to 20W	Yes
Chanzon LED Driver 900mA	AC 120 V	900mA	10W	No
B0F1K97V61 Constant Current Led Driver	DC 2.9-6.1V	28-1500mA	Up to 8W	No

Given that we could only find one driver that met all of our requirements, the LD24AJTA Constant Current LED Driver was our selection.

### 3.1.15.16 LED Microcontroller

Since the LED will require toggling, a microcontroller is going to be introduced into the LED's PSU. This will allow us to control when the LED is turned on and off, helping

with cooling and power efficiency. Fortunately, these our requirements are simple there are many viable options that can be integrated into our project as long as they have access to GPIO pins.

### 3.1.15.17 Arduino Nano ESP32 [ABX00083]

The first option we looked at was the Arduino Nano ESP32. This Arduino works off a USB Type-C that must supply 3-5V, which is typically the standard output of any USB Type-C port. Additionally, that same USB Type-C cable functions as data transfer, allowing us to program the Arduino. Its size is very small, about 43.1mm x 17.78mm, making it easy to integrate into the overall system. Furthermore, it is compatible with Python, making it easy to integrate with our software. Most importantly, it comes with 13 Built-in LED Pins; 14 Digital I/O Pins; 8 Analog input pins; 5 PWM pins, which is more than enough to control our driver. The prize is ~\$20, and comes with the benefit of brand name trust and ample documentation and support.

### 3.1.15.18 MCP2221A USB to GPIO

The second option we looked at was the MCP2221A. This microcontroller shares a lot of similarities with the Arduino Nano, as it is powered by a USB Type-C that must supply 3-5V and that same USB Type-C cable functions as data transfer, allowing us to program it. This microcontroller comes with 4 GPIO pins, and is even smaller at 27mm x 17.7mm. It is also cheaper, costing around ~\$15. The MCP2221A is another viable option for our project, although we are more comfortable with the ample support and documentation available for the Arduino Nano, as well as our team's familiarity with the Arduino family of products.

### 3.1.15.19 Arduino Uno REV3 [A000066]

Lastly, we considered another Arduino microcontroller, the Arduino Uno REV3. This can be considered a bigger version of the Arduino Nano, although it comes with some added drawbacks. The Uno REV3 also works off a USB Type-C that must supply 3-5V, and can be programmed via that same connection. It also includes access to more pins than the other options, those being 14 digital I/O pins (6 of which support PWM output) and 6 analog inputs. However, it does have a few drawbacks when compared to the previous options. Firstly, it is bigger and more expensive than the other options, measuring 68.6mm x 53.4mm and costing around \$27.60. Furthermore, the Uno REV3 does not support Python directly.

### 3.1.15.20 LED Microcontroller Product Selection

Table 3.1.27 Microcontroller Product Comparison

Product	Size	Operating Voltage	Ease of Integration	GPIO Pins	Price
Arduino Nano ESP32 [ABX00083]	Small	3.3V	High	Yes	Medium

MCP2221A USB to GPIO	Smallest	3V	Moderate	Yes	Low
Arduino Uno REV3 [A000066]	Medium	5V	High	Yes	Medium High

All 3 options were viable options for our project. Ultimately, we selected the Arduino Nano ESP32 as we are familiar with the brand, and it cost less than the Arduino Uno REV3 while also having Python support.

## 3.2 Communication Protocol Selection

Communication protocols are the backbone of data exchange between digital components in an embedded system. Choosing the right protocol determines how efficiently sensors, microcontrollers, and peripherals share information. Since our design involves multiple devices, including a camera module, LED driver, and touchscreen display, selecting an appropriate communication interface is essential for reliable and timely data transfer.

However, most of our peripherals come with integrated communication protocols due to their USB cable connections to the SoM. That means that the communication protocol we select will refer only to the Arduino and the LED controls.

### 3.2.1 UART

UART is one of the simplest and most common serial communication protocols, using only two lines to send and receive data between two devices. It is asynchronous, meaning no shared clock is needed, and both devices simply agree on a data rate. This makes it ideal for straightforward point-to-point communication, such as between a microcontroller and a computer. However, UART does not scale well for multiple devices and is slower than SPI or I<sup>2</sup>C. It is most useful for debugging, serial monitoring, or interfacing with low-speed modules.

### 3.2.2 SPI

SPI is a high-speed, synchronous communication protocol that provides full-duplex data transfer between a master and one or more slave devices. It uses four main lines: MISO, MOSI, SCK, and a slave-select line for each device. SPI is much faster than UART or I<sup>2</sup>C and is well-suited for peripherals that handle large amounts of data, such as displays or sensors. The main drawback is wiring complexity, since each slave requires its own chip-select line. Overall, SPI is the best option when performance and transfer speed are priorities.

### 3.2.3 Communication Protocol Selection

Table 3.2.1 Communication Protocol Selection

Protocol	Type	Typical Speed	Wires	Multi-Device Support	Ideal Use Case
UART	Asynchronous	Up to ~1 Mbps	2	No	Debugging or direct serial links
SPI	Synchronous	Up to ~50 Mbps	4 + CS lines	Limited	High-speed peripherals

The requirements we have for a communication protocol are very forgiving, so using either protocol was viable. We decided to use UART in this case, as the simple connection between the Arduino Nano and the LED driver didn't demand more than it can provide.

## 3.3 Software Selection

### 3.3.1 Data Analysis Software

Our design process requires a lot of repetitive calculations that can be intensive when done by hand, however, software with strong calculation and plotting abilities improves our time efficiency on design while providing professional plots that visually describe characteristics of the P.I.D.I system. Accessibility, prior experience, resources and cost will play a role in the tool we chose.

#### 3.3.1.1 Python

Python is an open-source programming language that is known for its simplicity and ease of use. Due to its simple syntax and extensive scientific, it is used heavily in the engineering industry and academia to help professionals, and students focus on data analytics rather than trouble shooting complex programming languages. This makes with the open-source accessibility Python the first coding language learned by many people. There are also many resources available due to its extensive usage and large community size.

The main reason we are looking at Python rather than other open-source programming languages is its superior integration of visualization tools via its extensive libraries. The only negatives about python is that we have higher experience level for data analysis utilizing different software and slower processing speed due to its age.

#### 3.3.1.2 MATLAB

MATLAB is a proprietary tool that uses simple syntax for numerical and engineering computing. It is developed by MathWorks and requires a paid license, but free educational versions are available. The University of Central Florida gives its students free MATLAB access with full features, making the price negligible for the P.I.D.I. project. MATLAB is

used extensively in industry and academia for many different applications. It is used extensively in imaging simulations due to its expertise and efficiency in matrix-based operations.

Due to its high uses in niche applications, the MATLAB specialized tool library enables specialized visualization techniques that are very useful in imaging and other applications. The team members also have a higher level of expertise in MATLAB compared to python due to the amount of usage through classes at CREOL. Overall, MATLAB is a simple data analysis tool that is fast and efficient with the best visualization tools out of the data analysis tools compared in this section

### 3.3.1.3 Microsoft Excel

Microsoft Excel is a spreadsheet software application that is the most common data analysis tool out of the tools being compared. This is due to its simplicity and versatility. Similar to MATLAB, it requires a paid license, but the University of Central Florida gives its students free Microsoft Office access with full features, making the price negligible for the P.I.D.I. project. Due to its spreadsheet nature, excel is superior at collecting data but not ideal for advanced calculations and specialized plots. Most advanced calculations can be done; however, it would require an advanced network of cell references which takes a lot more time compared to MATLAB and Python.

This makes Excel a powerful tool for simple data analysis and collection. Since we are looking for an advanced data analysis tool with advanced visualization techniques, excel would is not sufficient for the P.I.D.I. project.

### 3.3.1.4 Data Analysis Software Selection

**Table 3.3.1 Data Analysis Software Comparisons**

Data Analysis Tool	Calculation Efficiency	Data Visualization	Resources	Prior Experience	Cost
Python	High	Moderate	High	Moderate	Free
<b>MATLAB</b>	<b>High</b>	<b>High</b>	<b>High</b>	<b>High</b>	<b>Free as student</b>
Excel	Low	Low	High	High	Free as student

MATLAB was selected mainly due to its high capabilities of visualizing data and high amount of prior experience from the team members. Many raytracing and magnification calculations were verified via MATLAB. Due to its high calculation efficiency, it made it easy to design interactive simulations that allowed us to test different lens apertures and focal lengths to experiment with alternative design choices.

### **3.3.2 Optical Design and Simulation Software**

The optical design and simulation software will help ensure the optical system meets the system specifications before ordering the components. They do full analysis of the optical system as they treat light utilizing wave optics techniques, providing information that would require a very intensive MATLAB code to produce.

The main information we will be using the optical simulation software is to calculate the back reflections of the illumination, which would cause a high signal to noise ratio in the detected image. We will also use optical simulation to gain knowledge about the chromatic aberration in our system. . Accessibility, prior experience, resources and cost will play a role in the tool we chose.

#### **3.3.2.1 Ansys Zemax OpticStudio**

Ansys Zemax OpticStudio is the most common professional optical design and simulation software. This is due to its intuitive layout and visualization focus tools, making it more accessible compared to other optical design software. Zemax features include but are not limited to sequential raytracing, non-sequential raytracing, physical optics simulations, tolerance analysis and is able to integrate CAD models into the optical system. Optimizing optical design is one of the biggest uses of optical design software. System optimization works from the user's input as variable parameters such as lens radius or thickness, then set goals for the optimization from the merit function. This could be as simple as the focal length of the lens being designed or the spot size of a system. If you are optimizing multiple system parameters, weight of each parameter will set the order of priority of what to optimize for.

Our team is most familiar with Zemax compared to other optical design and simulation software. This is due to CREOL at the University of Central Florida providing licenses to their students. Without this academic license, it would be very challenging to learn software that is used in industry. Most of these software's are expensive due to their complexity and relatively niche application.

There are a lot of resources for Zemax. A simple google search may provide models with similar mechanics and optical techniques as the optical system being designed, which can be a great starting point when the user is still learning the software. This large number of already made models is the result of the worldwide community of users. There are also plenty of free video tutorials available on YouTube.

#### **3.3.2.2 Code V**

Code V is a professional optical design and analysis software that is developed by Synopsys. It has many features similar to Zemax such as sequential raytracing, physical optics simulations, tolerance analysis, wavefront analysis and automation via the included Macro Command Language (MCL) that automates the analysis and optimization of the optical systems. This emphasis on precise and advanced optimization is what sets Code V apart from Zemax. These features enable the design and simulation of high-end optics such as cutting-edge applications in the aerospace and defense industries.

Code V used to be much more expensive than Zemax due to its better optimization solutions. However, when Zemax switched its licensing model into a subscription, both

options became a lot closer in price. The official prices are not available on either company’s website, making it challenging to compare official prices. This limits the team members’ exposure to Code V since it does not provide accessibility to students.

The resources for Code V are not as abundant as the resources for Zemax. There are resources that are included when the software is purchased and some YouTube tutorials. The number of models online is also much less than the models available for Zemax.

### 3.3.2.3 3DOptix

3DOptix is a modern approach to optical design and simulation software. It is cloud based which makes it more collaborative compared to Zemax and Code V. It also has similar basic features as traditional optical design and simulation software such as sequential raytracing, non-sequential raytracing, physical optics simulations, tolerance analysis and can integrate CAD models into the optical system.

The software is the cheapest option to access as a student if the university does not have an optical design software. It also combines the power of the cloud with the user’s computer GPU to achieve fast simulation results.

The main negatives regarding 3DOptix is that it doesn’t have as many resources as Zemax and Code V due to its young age. It is also not as powerful in terms of the analysis it is capable of performing.

### 3.3.2.4 Optical Design and Simulation Software Selection

Table 3.3.2 Optical Design and Simulation Software Comparisons

Data Analysis Tool	Calculation Efficiency	Data Visualization	Resources	Prior Experience	Cost
Zemax	High	High	High	Moderate	Free as student
Code V	Ideal	High	Moderate	None	Need to request quote
3DOptics	High	Moderate	Moderate	None	\$40/month for students

Ansys Zemax OpticStudio is chosen as the design and simulation software of choice due to its accessibility, resources and the team members’ prior experience with the software.

### 3.3.3 Optical Schematic Software

To draw the schematic of the P.I.D.I’s optical system, we must pick software that has great compatibility with svg type files. We found a set of optical components in a svg file that are free of copyright that are mentioned in the appendix. The main requirement we

are looking for is software that enables efficient manipulation of a svg that also enables measurements to ensure the proper placement and spacing between the optical components

### 3.3.3.1 Microsoft PowerPoint

Microsoft PowerPoint is presentation software that is slide based. We got recommended to use it by students that have designed optical schematics with svg files in the past. The great thing about PowerPoint is that it is designed for presentation development, which means a lot of automatic alignment of pictures or svg files.

A convenient thing about PowerPoint is the ability to duplicate slides. This makes it very easy to make alternate designs within the same file rather than different files for each design, which is what Inkscape does. All the team members have been using PowerPoint since elementary school, making it a lot more familiar to work in compared to Inkscape. The University of Central Florida provides the Microsoft office suite for free to its students, making the price for a Microsoft office license negligible.

The problem with Microsoft PowerPoint is that it can get very unorganized very fast when the optical system utilizes as many optical components as the P.I.D.I optical system. The horizontal alignment makes it not ideal for drawing systems that are longer vertically compared to horizontally.

### 3.3.3.2 Inkscape

Inkscape is a an open-source graphics editor that is used in graphic design applications rather than presentation slides. This means that Inkscape is designed for designing svg files, which gives it a lot more features for this application compared to PowerPoint. The only negative is that the team members have not used Inkscape before. This makes Inkscape less efficient to do work in compared to Microsoft PowerPoint, but the extra tools and features enable the students to produce the highest quality schematic possible.

Another positive regarding Inkscape is that it is completely free with similar features as Adobe Illustrator and other similar expensive graphic editing software, which gives it the edge which developed a large community with many learning resources.

### 3.3.3.3 Optical Schematic Software Selection

Table 3.3.3 Optical Schematic Software Selection

Data Analysis Tool	Graphic Design Features	Resources	Prior Experience	Cost
PowerPoint	High	High	High	Free as student
Inkscape	Ideal	High	None	Free

Inkscape is chosen as the software the group members are going to utilize to compose optical schematics that are up to scale with real life models. Even though the group members don't have experience using Inkscape, the software's tools being intuitive helped with the initial learning curve of the software. The team members also wanted to learn the software for this project to make it easier to compose high quality optical schematics in the future.

### **3.3.4 CAD/3D Modeling Design Software**

For this project, a three-dimensional computer aided design (CAD) model will be created to design the housings and verify all optical, electrical and mechanical components fit within the designated size constraints (40mmx40mmx200mm for the periscope unit and 80mmx80mmx200mm for the camera unit) and that sufficient space exists along the designed optical path. The CAD model will also help visualize the layout of the system, ensure all components are properly aligned, and help facilitate the mechanical design as well as the manufacturing process.

There are many CAD programs capable of meeting the requirements of this project; therefore, the team will focus on software that members are already familiar with and have access to. The options being considered include Fusion360, SolidWorks, and AutoCAD.

#### **3.3.4.1 Fusion360**

Fusion360 provides all the essential features and functions for designing the 3D model of the periscope/camera unit and has a straightforward, user friendly interface. It also allows for direct and easy generation of G code, which would simplify the process if the team decided to manufacture the final housing using CNC machining or similar methods. fusion360 also supports exporting models as STL files, which will allow the housing and internal components to be easily 3D printed for prototyping and testing.

A limitation of fusion360 is that component suppliers such as Thorlabs where most optical parts will be sourced do not always provide native Fusion360 part files. While it is possible to convert compatible files, using software with direct support files is generally preferred to reduce the risk of file corruption or geometry alteration during conversion.

#### **3.3.4.2 SolidWorks**

SolidWorks is another strong candidate due to its easy to use assembly tools and extensive compatibility with optical and mechanical component libraries. Thorlabs and similar suppliers provide SolidWorks files directly, which makes importing components seamless and completely accurate. Additionally, SolidWorks built in simulation tools can assist with verifying mechanical tolerances and clearances. One of the limitations is that SolidWorks requires a licensed installation, which may limit accessibility outside of the university environment.

Like fusion360, SolidWorks can export STL files, enabling straightforward transition from design to 3D printing.

### 3.3.4.3 AutoCAD

AutoCAD is well suited for creating precise 2D drawings and layouts, making it a useful companion for early dimensional planning and technical documentation. However, its 3D modeling capabilities are less intuitive than those of fusion360 or SolidWorks. For this reason, AutoCAD will likely be used for drafting schematics or laser cut patterns rather than full 3D assembly modeling.

### 3.3.4.4 CAD Software Selection

Table 3.3.4 CAD Software Comparison

Software	Ease of Use	3D Modeling Familiarity	Simulations Familiarity	3D Printing Compatibility	Native Manufacturer File Support	Cost
Fusion 360	Moderate	Moderate	Basic	Full	Limited	Free
SolidWorks	Very intuitive	High	Strong	Full	Extensive	Free
AutoCAD	Requires experience	Limited	None	Full	Good	Free

In summary, while all three CAD programs can be used effectively, SolidWorks offers the most direct integration with optical component libraries, SolidWorks is also the software the team is most familiar with and for these reasons will most likely be the primary software used to model and export parts for 3D printing the final system.

### 3.3.5 3D-Printing Software

We will most likely be 3D printing the final housing for our prototype, as additive manufacturing allows for the creation of highly customized and complex geometries at a relatively low cost. Compared to CNC machining or other subtractive manufacturing methods, 3D printing enables rapid prototyping, easy design iteration, and reduced material waste making it ideal for developing our optical housing.

The 3D printing software considered for this project includes Ultimaker Cura, Slic3r, and PrusaSlicer.

#### 3.3.5.1 Ultimaker Cura

Ultimaker Cura provides one of the most comprehensive and beginner-friendly interfaces available for 3D printing preparation. It supports a wide range of 3D printers,

giving our team flexibility to use almost any printer accessible in university labs or personal setups. Cura makes it simple to adjust parameters such as infill density, build orientation, layer height, print speed, and support structures all essential for producing durable, light tight housing with high dimensional accuracy. Its visualization tools also help identify potential issues before printing, reducing failed print attempts and material waste.

### 3.3.5.2 Slic3r

Slic3r is an open-source slicing software known for its advanced customization and fine control over print parameters. It allows detailed modification of perimeter counts, infill patterns, extrusion widths, and other factors that influence part strength and surface quality. These features are particularly useful when optimizing prints for specific properties. Slic3r also offers detailed G code previews and supports multi material printing, making it an excellent tool for testing complex parts or experimental configurations. However, it has a steeper learning curve compared to Cura, making it better suited for experienced users who require maximum flexibility.

### 3.3.5.3 PrusaSlicer

PrusaSlicer, developed from the Slic3r platform, combines the customization capabilities of open-source software with a more intuitive and user-friendly interface. It includes preset profiles for various materials and printers, detailed support generation, and advanced options such as variable layer height and customizable modifiers. PrusaSlicer is compatible with most 3D printers and provides robust visualization tools for optimizing print quality and print time.

PrusaSlicer’s balance of power and usability makes it a strong alternative for users seeking precision without sacrificing convenience.

### 3.3.5.4 3D-Printing Software Selection

Table 3.3.5 3D Printing Software Comparison

Software	Ultimaker Cura	Slic3r	PrusaSlicer
User Interface	Very intuitive and beginner-friendly	Technical and less intuitive	User friendly with advanced options
Customization Level	Moderate, sufficient for most applications	Extensive, fine control over all parameters	High, advanced settings with guided options
Printer Compatibility	Extremely broad, supports most printers	Broad, but requires manual configuration.	Broad, supports most printers

Feature Set	Comprehensive basic and intermediate tools	Highly detailed parameter control and scripting	Comprehensive basic and intermediate tools
Visualization Tools	Strong, clear previews and error detection	Moderate, basic G-code visualization	Advanced, detailed layer and support visualization
Learning Curve	Very easy to learn	Steep, suited for experienced users	Moderate, accessible for most users

In summary, all three slicers can be used effectively depending on user preference and the specific requirements of each print. Ultimaker Cura offers the most straightforward workflow and dependable results, making it ideal for routine printing and rapid prototyping. Slic3r and PrusaSlicer, while more advanced, will be valuable for specialized parts that demand greater control over print parameters or custom material profiles. However, Ultimaker Cura will most likely be the primary software used for this project due to its accessibility, printer compatibility, and ease of use for all team members.

### 3.3.6 Embedded Programming Language

For the P.I.D.I project, there are two main pieces of software that will need to be programmed. First, we'll take a look at the various languages available and then decide.

#### 3.3.6.1 C

The most notable and common language when it comes to embedded systems is C. C offers direct, low-level access to hardware while still maintaining portability across microcontrollers and processors. It is a very lightweight language that allows for integration in low-level and low-memory systems. Additionally, most vendor-supplied SDKs, HAL libraries, and peripheral drivers are written in C, making it the natural choice for configuring timers, interrupts, communication interfaces, and memory-mapped I/O. C provides a great balance between performance, portability, and direct register-level access, which makes it an appropriate choice for both safety-critical and resource-constrained environments.

Besides performance aspects, C's popularity is further cemented through its extensive toolchain ecosystem that has been established over the years. Standardized subsets such as MISRA-C are commonly followed by organizations in regulated domains to avoid undefined behavior and non-deterministic constructs. Its predictable compilation model, ease of integration with other existing codebases, and myriad applications-including microcontroller firmware, sensor interface development, and control algorithms-ensure continued research support for C as the most common language of embedded development.

### 3.3.6.2 C++

The ability of C++ to provide abstraction, modularity, and object-oriented design has secured it a place of increasing popularity in embedded systems research. Long-term maintainability studies of embedded projects often note that compared to C, C++ offers better component encapsulation and code reutilization, particularly for projects with complex systems composed of layered software architectures or heterogeneous subsystems. Modern versions of the standard for C++ (C++11 onwards) offer improved features relating to type safety, template metaprogramming, and compile-time evaluation, all of which are documented to offer reliability improvements and reduced boilerplate code patterns in embedded application domains.

Despite these strengths, a number of constraints are present in the language. For example, certain language features such as dynamic allocation, exceptions, and runtime type information can introduce unpredictability in real-time contexts. Consequently, a number of embedded-focused C++ subsets have emerged that constrain high-overhead features but retain many benefits of abstraction. Overall, while C++ offers long-term maintainability gains, it requires disciplined design practices in order to guarantee predictable execution.

### 3.3.6.3 Python

Python has become increasingly relevant in embedded systems research, especially in domains that include rapid prototyping, data processing, and high-level system integration. Some of the biggest advantages of Python include accessibility, extensive scientific libraries, and compatibility with embedded Linux platforms like Raspberry Pi. The language is particularly fit for research contexts that require fast iterations: computer vision, machine learning inference, automation, and sensor data analysis, where development speed and algorithmic flexibility are more important than strict real-time control. This is highly advantageous in our case, as we are dealing with high algorithmic speed requirements.

However, the usage of Python also presents limitations in environments that require deterministic execution or minimal latency. The presence of an interpreter, dynamic typing, and garbage collection introduces nondeterministic behavior, which is incompatible with classical real-time embedded constraints. MicroPython and CircuitPython try to avoid these issues; however, they still present significantly higher memory footprints and slower execution speeds compared to C-based implementations. All in all, Python is best suited for embedded systems with operating systems, computational headroom, or hybrid architectures where microcontrollers handle real-time tasks while Python manages high-level coordination.

**Table 3.3.6 Embedded Language Selection**

Language	Strengths	Limitations	Prior Experience
----------	-----------	-------------	------------------

C	<ul style="list-style-type: none"> <li>• Highest performance</li> <li>• Tiny memory footprint</li> <li>• Direct register-level access</li> <li>• Vendor HAL/SDK support is ubiquitous</li> </ul>	<ul style="list-style-type: none"> <li>• Development is slower</li> <li>• More boilerplate</li> <li>• Harder to maintain at scale</li> <li>• Fewer high-level libraries for vision/ML</li> </ul>	High
C++	<ul style="list-style-type: none"> <li>• Abstraction and modularity for complex systems</li> <li>• type safety,</li> <li>• Good performance near C with careful use.</li> </ul>	<ul style="list-style-type: none"> <li>• Some features complicate real-time behavior without strict guidelines</li> <li>• More complex toolchains</li> <li>• Potential code size growth.</li> </ul>	High
Python	<ul style="list-style-type: none"> <li>• Fast prototyping</li> <li>• Rich ecosystems for CV/ML/data (NumPy, OpenCV, PyTorch/TFLite)</li> <li>• Excellent on embedded Linux like CM5</li> <li>• Easy integration with C/C++ via bindings.</li> </ul>	<ul style="list-style-type: none"> <li>• Not deterministic</li> <li>• Interpreter and GC add latency</li> <li>• Heavier memory footprint than C/C++</li> <li>• Pure-Python loops are slower.</li> </ul>	High

After listing each language’s strengths and weaknesses, we decided on Python as the language to be used for the software. Although there may be drawback with performance, it’s easy integration with the Raspberry Pi environment, as well as the breadth of libraries available, make Python the clear choice for our project.

### 3.3.7 Development Environment

#### 3.3.7.1 Eclipse

Eclipse has been one of the longest-standing environments in embedded development, especially when coupled with plugins like CDT for C/C++ work. It has great project-structure tools, integrates well with Makefiles, and can be used for debugging out of the box. Its plugin ecosystem is huge and includes support from microcontroller vendors, which can be advantageous for projects relying a lot on low-level C or C++ firmware. Eclipse is resource-heavy, and on modern systems, this indeed can give it the feeling of being a bit sluggish or cumbersome. Its UI is highly outdated, and the learning curve is substantially higher compared to more modern IDEs.

Where Eclipse really shines is in large-scale enterprise or classical embedded workflows. For fast iteration cycles or for multi-language developments, it's less efficient. Due to its configuration overhead and slower responsiveness, it is not ideal for this project, as it often switches between pure Python-based algorithms, C/C++ extensions, and hardware debugging on a Linux-capable platform like the Raspberry Pi CM5.

### 3.3.7.2 Visual Studio Code (VS Code)

VS Code has become one of the most widespread development environments throughout software and embedded systems because of its flexibility, performance, and wide plugin library. With support for Python, C, and C++-and dozens of other languages with extensions-it is well-suited for dealing with all the code components of the P.I.D.I. project from a single environment. Git integration, virtual environments, SSH remote editing, and container-based development make it perfect for working directly on the Raspberry Pi CM5 or on a remote server.

Furthermore, VS Code is the environment with which we have the most experience using. This is particularly important when it comes to a development environment as it allows for the user to work more efficiently thanks to the sense of familiarity and previous knowledge.

**Table 3.3.7 Development Environment Selection**

Environment	Strengths	Limitations	Prior Experience
Eclipse	<ul style="list-style-type: none"> <li>• Strong debugging</li> <li>• Large plugin ecosystem</li> </ul>	<ul style="list-style-type: none"> <li>• Heavy</li> <li>• Outdated UI</li> <li>• Steep learning curve</li> </ul>	Moderate
VS Code	<ul style="list-style-type: none"> <li>• Lightweight</li> <li>• Highly customizable</li> <li>• Supports Python + C/C++</li> <li>• Remote development</li> <li>• Strong extensions</li> </ul>	<ul style="list-style-type: none"> <li>• Requires extension setup</li> </ul>	High

Thus, primarily due to the lengthy experience already using VS Code, that was our final choice for development environment.

## **Chapter 4 – Standards and Design Constraints**

### **4.1 Industry Standards**

#### **4.1.1 I2C Standards**

The I2C protocol refers to a standard used by embedded systems. Most embedded systems contain multiple processors and peripherals that must transfer data between each other. In order to minimize the wiring required to make this happen, a serial bus is implemented and it is here where the I2C protocol defines the specifications of the bus.

Following I2C standards, only two wires are utilized, a serial clock wire (SCL) and a serial data wire (SDA). Furthermore, every component connected through the data bus is designated as either a slave, master, or both. The master devices generate bus clock and initiate the communication, while the slaves simply respond to the commands. Additionally, in order to facilitate communication, each component is given a unique address so that data transfer can be efficient.

Evidently, it is important that we follow the I2C standards in the P.I.D.I. project as we do have multiple components connected to each. Aside from the MCU that will be embedded in the system, we have to ensure proper communication between the camera and the touch-screen display. These devices will be in constant communication with each other as the user interface will control the camera, and the camera will feed the image to the MCU in order to run the image processing algorithm.

The I2C standard also establishes electrical and timing constraints that must be met to ensure reliable data transmission. Both the serial clock (SCL) and data (SDA) lines are open-drain, meaning they require external pull-up resistors to maintain proper voltage levels and prevent signal contention on the bus. The standard also limits total bus capacitance to 400 pF and specifies distinct speed modes, defined as 100 kbit/s for Standard-Mode, 400 kbit/s for Fast-Mode, and up to 3.4 Mbit/s for High-Speed Mode. These constraints are particularly important in compact systems like P.I.D.I., where wiring length, electromagnetic noise, and multiple devices on a single bus could otherwise cause communication errors or data loss.

Adhering to the I2C protocol not only ensures electrical compatibility but also promotes scalability and maintainability in the system. By following the standardized addressing and timing structure, future sensors or peripherals can be added to the design without requiring new communication hardware. This compliance allows the P.I.D.I. system to remain flexible for future upgrades, as well as potential modularity, while maintaining robust communication between the MCU, camera, and display modules. Conforming to the I2C standard ultimately guarantees that all components regardless of the different manufacturers conform to consistent data transfer, which is critical for accurate image processing and responsive user interaction.

#### **4.1.2 Battery and Power Standards**

The P.I.D.I. system is powered by alkaline batteries, chosen for their low cost, wide availability, and stable voltage output suitable for embedded systems. To ensure safe and reliable integration of these batteries, the design follows standards that govern the

performance, safety, and environmental handling of primary cells. The main reference is IEC 60086, Primary Batteries, which defines international requirements for dimensions, labeling, safety, and performance characteristics of alkaline cells. Specifically, IEC 60086-1 covers general requirements such as size and discharge characteristics, while IEC 60086-5 outlines safety tests including short-circuit resistance, incorrect installation, over-discharge, and temperature exposure. Adhering to these standards ensures the P.I.D.I. power system operates reliably and safely under typical laboratory and demonstration conditions.

## **4.2 PCB Design Standards**

### **4.2.1.1 Manufacturer's Design Standards**

Printed Circuit Board (PCB) design standards define the physical and electrical criteria that we must follow to ensure that the PCB can be manufactured and assembled reliably. We have chosen to use JLCPCB for the PCB fabrication required for the P.I.D.I. project. JLCPCB provides standardized design and production specifications that we must follow to ensure the quality of our product. These include requirements for minimum trace widths, copper thickness, via and hole diameters, clearances, and edge spacing. If we follow the specifications provided by the manufacturer, we can also guarantee that the design files will be compatible with the manufacturer's process. This means that we can be sure the process will be completed without running into any preventable issues. Thus, we can guarantee that the finished PCB will meet both functional and mechanical requirements.

Looking into JLCPCB's published manufacturing standards, we found the following standards:

- The thickness of the copper may be selected from either 1 oz or 2 oz per square foot.
- Any finished hole must have a minimum size of 0.3 mm.
- All via pads must have a minimum diameter of at least 0.5 mm.
- The clearance between the hole and the pad must be at least 0.5 mm, and vias must be placed a minimum of 0.254 mm away from other tracks or vias.
- Minimum trace width and spacing are both 0.127 mm.
- The minimum distance from any copper feature to the edge of the PCB must be 0.3 mm.

These dimensional standards ensure proper electrical isolation and mechanical robustness during etching, drilling, and plating processes. Files must be provided in Gerber format, and drill data should be in Excellon format, as required by JLCPCB for fabrication.

### **4.2.1.2 Industry's Design Standards**

In addition to the manufacturer's specifications, we must ensure that the PCB design also adheres to the IPC-2221 generic standard for printed board design. IPC-2221 provides detailed guidance for parameters such as trace width, conductor spacing, and dielectric thickness based on operating voltage, current density, and thermal constraints.

Following the standards set by the IPC-2221, helps ensure that critical aspects of the P.I.D.I. system function as required. Primarily, this refers to thermal management as well

as power distribution. It is important that we follow such specifications, since there are multiple components and peripherals attached to the project, and the best way to ensure everything works accordingly is by following these standards.

Among the many standards and requirements set by the IPC-2221, we can find the following categories:

- **Electrical Properties:** These standards dictate electrical performance, power distributions, and conductive material requirements.
- **Component and Assembly:** These standards dictate accessibility, clearance between components, and requirements for through-hole and surface mount components.
- **Thermal Management:** It involves standards related to heat transfer, heat dissipation, and cooling mechanisms
- **Holes:** As expected, it dictates hole dimensions, placements, and spacing as well as broader general requirements.
- **Material:** It involves the restrictions on certain materials, as well as dictating the spacing and thickness of materials used to fabricate the PCB.
- **General Requirements:** Lastly, general requirements involve regulations for things like testing considerations as well as general layout and design.

Thus, our group endeavors on following these standards and requirements as to ensure that our PCB is designed and manufacture in a way we are compliant. This is to our benefit as well, as we can be sure that we comply with safety measures and we facilitate the use of a variety of components from multiple manufacturers.

### **4.3 Mechanical Hardware standards**

The fabrication of 3D printed components for the P.I.D.I. system will follow established additive manufacturing standards to ensure that all parts meet functional, dimensional, and mechanical quality requirements. The two main standards used for this purpose are ISO/ASTM 52910:2018 Additive Manufacturing Design Guidelines and ISO/ASTM 52902:2023 Additive Manufacturing Test Artefacts for Geometric Capability Assessment.

ISO/ASTM 52910:2018 provides design principles and recommendations for creating parts through additive manufacturing. It includes guidance on geometry optimization, material choice, and print orientation to help minimize warping, reduce internal stress, and improve consistency between prints. By following this standard, the optical housing, periscope, and camera casing can be designed with appropriate wall thicknesses, tolerances, and support structures that suit the FDM printing process. These guidelines also help limit post-processing and improve overall print reliability.

ISO/ASTM 52902:2023 focuses on evaluating and verifying the geometric accuracy of additive manufacturing systems by using standard test artifacts. These printed models are used to check a printer's dimensional accuracy, repeatability, and resolution. Applying this standard ensures that all 3D printed parts stay within acceptable tolerance limits so that the optical alignment and overall assembly fit remain consistent. Regular calibration using

these test artifacts helps confirm that the printers maintain the accuracy required throughout the project.

## 4.4 Design Constraints

Our sponsor designated the dimensions of the device we were going to build, and we found the dimensions to be very limiting. We are designing a modular common illumination and detection path demagnifying reflective darkfield imaging system used to detect contamination particles 50  $\mu\text{m}$  – 300  $\mu\text{m}$  in size. We will have two modules, The camera module, which will house all the illumination optics and electronics for darkfield as well as some of the optics and electronics for the camera, and the periscope module, which will relay the light both to and from the object surface we are detecting contaminants on to the camera module. Our sponsor gave us the following volume and practical constraints for the system:

- The Camera Unit must be a sponsor specified size.
- The Periscope Unit must be a sponsor specified size.
- The distance between the camera unit and the periscope unit must be a sponsor specified size and may not have any optical components between it.
- The distance between the periscope unit and the object surface must be a sponsor specified size.
- The light coming in and out of the camera unit must follow the same path (Common illumination and detection path).
- The surface we are imaging should be a sponsor specified size for one image.
- The system should be able to resolve particles.
- The total cost of the system should be less than 5000\$
- The system should be built only using off the shelf components.

### 4.4.1 Initial System Design Constraints

Our initial research given these design constraints led us to consider using an infinity corrected objective lens and associated tube lens design. This combination of optics is a standard for most lowlight microscopy imaging systems such as fluorescent or darkfield for many reasons. The main reason is because of the functionality of an infinity corrected objective. The infinity part of that name basically means that the rays coming through the objective lens come out with a theoretically infinite radius of curvature or simply put parallel, creating a collimated beam. A collimated beam simplifies the design of a microscope greatly, allowing for easy substitutability for any optics within the collimated beam path; Aswell as allowing you to keep any mirrors or additional elements to a standard size without having to worry about changing the mirrors later.

While we say it is collimated, it is probably more accurate to say it is functionally collimated. In real life as light goes through an opening, based on the size of that aperture the beam will diffract or bend around the edges. Due to this effect, the beam will still spread very slowly and due to that spreading there is a functional limit to how far that beam will remain collimated. Infinity corrected objectives though can remain collimated over meters of length which is extremely large for most microscopy applications. Considering the shortest optical beam path we could have used to create our particle inspection system is

around 0.5 meters from the object surface to our camera, this technology would have been very functional.

The other component of this design that increases simplicity for microscopy imaging systems is the tube lens. This is a finite focal length lens that theoretically takes a collimated beam and sends all light to a single point, the focus. In real life though as an effect of geometry and other properties of light, rays don't always focus evenly leading to a series of factors that can majorly affect the output image quality; These factors are called aberrations. Generally, there are methods for combating certain types of aberrations, but the calculations involved take an extended period of time to solve and consider. Ultimately the most efficient method for calculating and mitigating aberrations is putting your system into a sophisticated lens design software such as Ansys Zemax and finding how each lens contributes to total aberrations. This is an intensive process that takes time, careful design, and intuition to do well. The tube lens has had this anti-aberration workflow done already. It is designed specifically alongside the many lenses that make up the infinity corrected objective lens to mitigate aberrations. This significantly decreases the amount of work necessary to have a functional, aberration free optical system.

In our first month of senior design, this configuration seemed like a very promising candidate but after signing an NDA and speaking to our coordinator with our sponsor we ran into an issue. The object surface we are imaging needs to be sponsored specified distance away from the edge of the periscope and the periscope can only be sponsored specified distance thick. A sponsored specified distance from the object surface to the edge of the periscope translates to having the objective lens having a minimum working distance of 5 mm -100mm, which is very lengthy for objective lenses. Also considering we need all our components to be purchasable "off the shelf," We found that all long working distance (LWD) objective lenses with working distance over sponsored specified distance were both expensive, over 2000\$, and larger than a sponsored specified distance in size. Without having the front end of the objective lens fitting out the periscope, this design became impossible to implement while keeping the dimension specifications given to us by our sponsor. Ultimately, we had to reconsider the entire design of our system.

#### **4.4.2 Current Iteration Design Constraints**

After further research into potential methods of achieving our goals with the given design constraints, we discovered another very promising design option that allows for better control of the required magnification and better modularity, which gives us the ability to change our design quickly if we run into any problems or necessary changes down the line. As it is inevitable that we will come across unforeseen issues with our design, this was a very desirable characteristic.

We decided to use a series of 4f system relays. A 4f system is an optical configuration where you have an input plane, a lens placed at one of its focal length away from that plane, a second lens placed at a distance of the focal length of the first lens plus the focal length of the second lens, and an output plane at a distance of one focal length of the second lens to the output plane. This could be modeled by looking at the following figures:

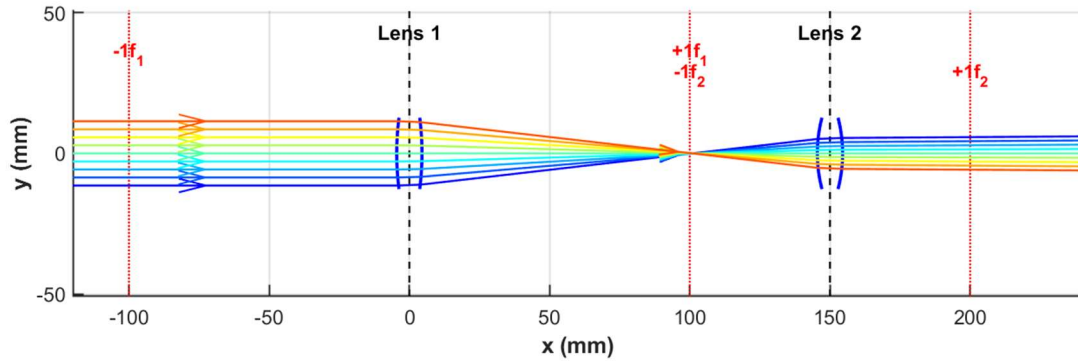


Figure 4.4.1 Input Rays Parallel

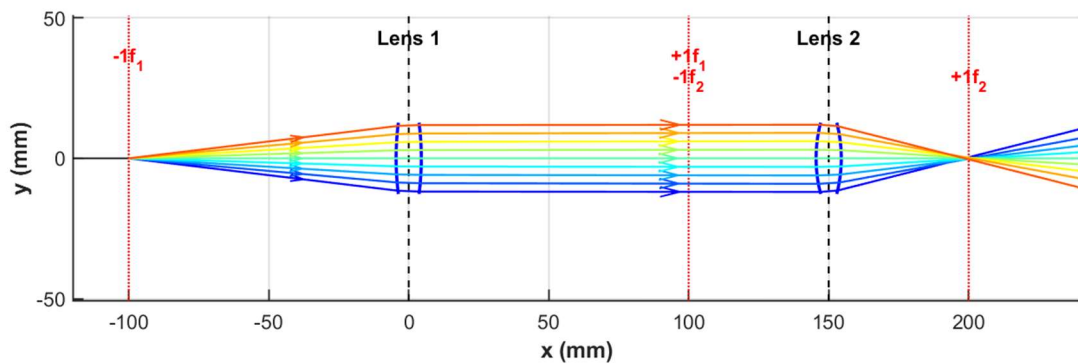


Figure 4.4.2 Input Rays come from Focus

As seen in Figure 4.2.2.1 the light rays emerge parallel to the optical axis, hit the first lens, get focused to its focal point, hit lens 2 and reemerge parallel. Although the set of rays that emerge are more tightly focused. You can think of this configuration as the mapping of one point at the object plane to one point at the image plane. The total size of a 4f relay is two times the focal length of the first lens plus two times the focal length of the second lens. The magnification for this example system in both cases could be found by the following equation:

$$M = \frac{f_2}{f_1}$$

By placing a series of these optical configurations concurrently we could achieve the required magnification while also allowing for better control over the placement of the lenses which will be critical in ensuring a functional design that fits in the desired

specifications. You could consider a single 4f system as almost a “base unit” in which we can structure our optical system around.

For our system the target magnification calculated by equation is between 0.23x and 0.29x magnification. It is important to note that for magnification values between zero and one the system will demagnify the object or make a larger object at the input, smaller at the output.

## 4.5 Size Constraints

### 4.5.1.1 Lens Diameter

Without considering any calculations, the largest size of lens available to us throughout the system is beneficial. This is due to the fact that our system will be imaging low amounts of light and the more light we can get through, the higher likelihood the system will work. If we have too much light hitting our sensors there are both image processing techniques and physical options for lowering the amount of light. We could for example lower the exposure time, decreasing the period of time in which we collect light for an image, or add an aperture stop which will physically block a portion of the beam, decreasing the total amount of light hitting our sensor.

Our smallest dimension concerning lens diameter can be found in the periscope unit. The periscope unit is a sponsored specified distance thick, meaning that the largest lens we can make is sponsored specified distance minus the space needed to hold the lens. Commercial lenses are made at standard sizes of 3 mm, 5 mm, 6 mm, 9mm, 1/2” (12.7 mm), 1” (25.4 mm), 1 1/2” (38.1 mm), and 2” (50.8 mm), with varying amount of focal length options per size. The largest selection of differing focal lengths lenses is found for the 1” and 1/2” size lenses. Functionally, this sponsored specified distance thickness limitation combined with the off the shelf lens condition means that the largest size of lens we can choose is 1” (25.4 mm). Due to these size limitations, using official concomitant lens mounts would be impossible, meaning all the lens mounts will have to be 3D printed.

### 4.5.1.2 Focal Length Limitations

Using the given size constraints and a series of 4f relays, we developed a set of equations for which we can fit our set of solutions for finding focal length around. It should be noted that our design philosophy was to design the detection path magnification first. Initially we were hoping we could limit our design to two 4f relays but due to the following set of conditions, the size constraints were too stringent. Please refer to our diagram in chapter 2 to find the names of lenses.

$$f_1 > A \text{ sponsor specified distance}$$

$$f_3 + f_4 > A \text{ sponsor specified distance}$$

$$2f_1 + 2f_2 + f_3 \leq \text{Optical Path length at periscope end}$$

$$2f_1 + 2f_2 + 2f_3 + 2f_4 < \text{Total available path length}$$

$$M_{total,detection} = M_{d1}M_{d2} = \frac{f_2 f_4}{f_1 f_3}$$

The first condition comes from the fact that we need to be a sponsor specified distance between the object surface and the camera module. Design-wise this means that our first lens (from object to camera) needs to have a minimum focal length of 40 mm. Including lens thickness and the fact that we can not have any of our parts sticking out of the periscope, this ultimately means that the shortest off the shelf lens we could purchase would be a sponsor specified distance in length.

The second condition, the most limiting, comes from the fact that the camera module and the periscope unit need to be placed is a sponsor specified distance. Design-wise, considering that we are unable to place any optical components between the camera module and the periscope unit, this means that the shortest distance we can have between the two lenses for the second 4f system is a sponsor specified size, again thicknesses of the lenses should be considered increasing this value slightly.

The third condition is due to optical path length limitations. To not have any of our optical components sticking out between the camera unit and the periscope unit, we will need the placement of our first three lenses to fall within the periscope unit. Mathematically, this means that to place all three of those lenses physically we would need to have the full required space for the first two lenses plus half the required space for the third lens to be less than the total path length. This is also not considering that we will need to place two mirrors in the periscope unit which limits further the solution space for which we could fit these conditions.

The fourth condition is the least limiting condition for this problem, but still a consideration. The full distance of the 4f relays need to fit within the total optical path length available. This can be a limiting factor but due to clever use of 3d space the total optical path length available can be much higher than what these lenses will normally require.

The fifth condition written is the required detection magnification. We calculated that for an object size of near a sponsor specified distance and sensor size of around 3.5 mm we would need a magnification around 0.2x- 0.5x. While all the other conditions are limitations regarding the placement of our lenses, this condition ensures our optical system has the ability to image objects of a certain size.

#### **4.5.2 Two 4f System Design Constraints**

These set of limitations proved to be a huge problem for our two 4f relay detection path design. To have a magnification of around 0.2x-0.5x, only including two 4f relays and having the first lens be sponsored specified distance minimum, we would need the magnification of both 4f systems to be less than 0-2x. Considering that the smallest magnification possible for the first 4f system while preserving lens size is 30/45, or 0.666, the required magnification for the second 4f is around 0-2x. This fact combined with condition two made the number of solutions available to us tiny. Using only off the shelf lenses, we only found a single configuration that met all these properties, and this was not considering the possible issue of having to place our mirrors.

Further research and design also opened up another glaring issue with the two 4f relay design. Any demagnification made over the common detection and illumination path

would have to be counteracted for the illumination path to guarantee spot sizes were appropriate throughout the entire system. The system was designed to demagnify in one direction. Reversing the direction of the system would result in the reciprocal of the demagnification, or in other words a magnification. Additional optics are then required to demagnify the illumination prior to reaching the common path. This proved to be impossible given the dimensions of the camera module.

## **4.6 Time Constraints**

One main constraint for this project is time. As this is a senior design project, we only have a certain amount of months to plan, design, redesign, implement, and evaluate our system. This limitation greatly reduces our ability to perfect our design and create our most refined final product. Having a time limit also forces our group to favor certain goals over others. For our project, this means we will prioritize image quality, modularity, size, simplicity for means of use, and reliability over auxiliary features such as adding a brightfield illumination option or aesthetics.

As is true with any senior design project, there are aspects of our design that will fail and require time to diagnose and correct. While this is a natural limitation, it is in our best interest to get as far ahead in designing and testing as possible, as to either catch these errors earlier or to give us more time to react to those unavoidable problems.

To mitigate some of these unforeseen issues, our group has implemented a few strategies in an attempt to optimize our time management. One major strategy we used, for the Photonics Science and Engineering portion of our team, was a Gantt chart. Detailing the expectations for each period of time made it simple to organize our schedule and manage progress. While is true for any extrapolation of future events, there were time expectations for certain goals that were either too ambitious from the jump or did not get met in the way we were expecting. Regardless, this was an invaluable asset and lesson in trying to manage our time efficiently.

Another fantastic strategy we used to assist quality time management is effective communication. On top of our bi-weekly meetings with our sponsor, our group also met weekly to discuss our findings and review our progress for current and upcoming goals. In addition to this, anytime there were any semi-significant changes to a part of the project that affected other people, we promptly made each other aware. Another useful strategy for maximizing our success was keeping each other accountable. We would message each other in the group chat; updating each other with how we were progressing or any roadblocks we were facing.

One final aspect that is worth considering is the amount of time it will take to receive the parts. Shipping delays could limit our time to implement or assess components. In addition, due to the fact that this is a sponsored project and we are being entirely funded by our sponsor, we will have to go through our sponsor to make any final decisions about purchasing parts. In reality, this does not have to be a huge loss of time. With careful planning and good time management we can send a parts list early and ensure we stay on track. Although as there will inevitably be unforeseen issues, we may end up in a position where we discover a problem late on a Friday night and we will be unable to update our

sponsor on the situation until the following Monday morning. Though, this is a tiny factor and potentially an afterthought.

While time remains a main constraint of our project, proactive planning, consistent communication, and timely coordination with our sponsor will allow us to manage this limitation and ensure we deliver a quality refined final product without being hindered by time restrictions.

## **4.7 Economic Constraints**

Economic constraints are an important consideration for any engineering project. Thankfully in our case, ASML has provided adequate funding to allow our team to work freely without too much concern. Although optical components can get expensive very quickly, the budget of \$5000 should be more than sufficient in ensuring we can deliver on their expectations for the project. Frankly, not having to worry about funding allows our team to spend time we would have spent on fundraising on designing and testing. In addition to saved time, this budget gives us the flexibility to make better design decisions, leading to less choices where we must sacrifice cost for quality.

Although we have a larger budget than most senior design groups, this does not mean that we are spending it wantonly. We are making a deliberate and continued effort to keep costs down and make frugal design choices. We have considered multiple vendors and compared prices to make sure we are getting the best value without compromising the system's reliability.

A key part of the proposal made to us by ASML was the modularity of the system. By keeping costs low, as well as only using off the shelf components, we ensure that our design is reproducible and able to be scaled efficiently. Maintaining as cost-effective a design as possible allows ASML to upgrade or adapt our system with minimal additional investment.

## **Chapter 5 – Application of ChatGPT**

Since ChatGPT's release to the general public on November 30th, 2022, its utility and ability to solve complex problems and demonstrate solid reasoning in difficult fields like mathematics, physics, chemistry, and many more has increased dramatically. Previously, I had used ChatGPT in some classes to help me correct errors or inefficiencies in my code or get a base jumping off point when learning and searching for arcane material. The effectiveness and prevalence of LLMs like ChatGPT are hard to ignore and have become a serious subject of debate for multiple classes I have taken. Some teachers advocated its use and consideration as a serious tool for upgrading your efficiency in learning and completing work and some teachers viewing it more as a tool for cheating and avoiding work. While both opinions have truth to them, coming into senior design one and with the release of GPT-5, I expected ChatGPT to be one of the most useful tools at our disposal, but my experience has both supported and disproved that assumption.

At a base level, LLMs function like an advanced search engine. They are able to find and explain information quickly that previously was only available by combing through website after website or subjects typically reserved for textbooks or other academic literature. At a higher level, it's able to solve complex problems and work through drawn

out derivations but with a significant catch. LLMs will “hallucinate” or make up information confidently if it doesn’t fully know the answer to your prompt. Even despite these are very powerful tools if used correctly.

Though, not all LLMs are trained the same and as a result do not behave the same. Throughout this section, we use both Claude and ChatGPT for varying tasks and to varying degrees of success. Overall, I would describe Claude, as opposed to ChatGPT, as having a more conversational tone and being more willing to admit when it does not know something. Whereas ChatGPT tends to be more structured in its formatting. ChatGPT also can browse the web in real time, helping this model take an edge over competitors for the moment.

Another interesting tidbit is that the most recent version of ChatGPT, for whatever reason, is extremely fond of pushing its image generating capabilities, despite the fact that its ability to generate images that are correct is downright horrendous. I have had many instances where I ask a series of technical questions to ChatGPT and it asks if it can make an image based on my question. Every single time I have said yes to that request, ChatGPT takes around a minute to load the resulting image, which in my experience; has always been incorrect. Not once have I had it generate an image and explain information in that fashion faster than it would have taken me to either look it up or just ask ChatGPT for more insight in word form. Another ChatGPT fun fact is that every single time you ask a question ChatGPT will read the entire conversation to answer. This also explains why it is necessary, for efficient use of ChatGPT, to open new conversations often. Longer conversations means longer thinking times as well as more inconsistent results.

Finally, in terms of our senior design project, we have considered the following uses for it and more:

- Base level research
- Coding software
- Drafting reports
- Explanation of technical concepts
- Suggestions for solving engineering problems.

## **5.1 Case Study 1**

Please refer to Appendix, for the Case Study 1 given prompt and outcomes from ChatGPT

Initially, I used ChatGPT as a resource to figure out how I approach learning how to ray trace. Although it was able to provide some insight, I found ChatGPT’s response to be dangerously inefficient. ChatGPT eventually gave helpful correct information, but first it spewed out many small irrelevant caveats and superfluous pieces of information, especially considering the length of question I prompted. As an advanced search engine, ChatGPT worked great, confirming information I already knew about ABCD and geometric ray tracing. Though as a tool for actually coding this system in MATLAB, I found it to be very inefficient.

Having been familiar with ChatGPT's limitations, I first attempted to learn the geometry of accurate geometric ray tracing myself instead of asking it to code my system in one go. After learning the math behind it, I attempted to direct ChatGPT to solve one specific part of the code at a time, with varying degrees of success. Initially, it gave me a decent baseline for starting a ray tracing code but after reading it closer it was clearly missing critical information about the problem at hand. ChatGPT also immediately made syntax level errors in the code, even when asking extremely limited scope coding questions.

While ChatGPT is usually strong with logic and code, I found it to work with mixed results. One reason for this is due to the fact that ChatGPT codes at an advanced level compared to a rookie like me. I found that often when asking ChatGPT to code something for you, the structure of the code was complicated and difficult to understand, frequently using coding structures and commands I had not been exposed too. I had to spend a lot of extra time combing through the internet looking for specific functions. Even though I have the ability to understand the logic and math behind the code I wanted to program, actually converting base logic into code is a time-consuming task. As a result, anytime ChatGPT gave me a nonfunctional code, I immediately sent it the error message and asked it to diagnose the problem. For simple syntax errors this process usually fixed the code in less than one or two iterations, but for more complicated errors it took ChatGPT upwards of ten separate times asking before it solved what issues I wanted. There were lot of times throughout this process where I would feed ChatGPT its own code and it would randomly change small features forcing me to prompt it yet again.

To actually model geometric ray tracing correctly, even using the strategies mentioned, it took over ten different conversations of varying length, as well as two days of work to finally achieve accurate ray tracing behavior for a single lens. Using ChatGPT to scale it further was not too complicated and took few iterations in comparison.

While ultimately I got ChatGPT to write the intended code, the time investment was almost not worth the effort spent. Although, I would argue, not spending the time learning how to syntax my logic in MATLAB did harm my learning experience for senior design in one way. In another, it freed up time to work on designing or study other topics more pertinent to my skill set.

## **5.2 Case Study 2**

Check appendix for the Case Study 2 given prompts and outcomes from ChatGPT.

At the beginning of senior design one, I had no clue how to even approach understanding potential designs for our imaging system. ChatGPT was a great resource in getting a general overview of the technologies that existed. While the design decisions it tried to inform me, one were outright incorrect for the specific project we were developing, ChatGPT revealed to me a lot of the then unknown terminology and common configurations for solving the problem at hand. Beyond that, I learned enough about the specifications I was looking for to be able to iterate on further designs.

Beyond that initial design phase, ChatGPT was used to verify design information I only understood partially. After considering a new design, ChatGPT became the fastest method in which I can verify whether my understanding of a certain design or fact was

correct. Rather than spending countless hours scouring papers or combing forums, only to find people explaining their work at a higher level than I was currently at, ChatGPT allowed me to expedite this process by either providing sources, and therefore close to direct access to the information I need, or allowing me to ask as many stupid question as I wanted; a luxury that previously required a lack of shame and a patient teacher.

ChatGPT can enable someone to do the bare minimum if they want to. Although, when used effectively it can be an amazing tool. It is also important to not expect too much out of ChatGPT. Naturally, ChatGPT did not get everything right with our project and what kind of design needed to be done. The type of project we are working on is not something that has been extensively covered on the internet or exactly in textbooks. As a surface level researching tool, it led me in the right direction; Allowing me to later ask more specific and relevant questions about our project.

### **5.3 Case Study 3**

Refer to Appendix B.3 for case study 3 given prompt and outcomes of ChatGPT and Claude AI

To help improve our detection path raytracing, we utilized ChatGPT and Claude AI to give us a starting point of MATLAB code. There was a big difference in the quality of code provided.

ChatGPT 5.0 evolved to have a PhD students' level of knowledge, however, it is not the best AI tool for coding applications. When it tried to write the code of the provided prompt, it did not plot it properly while also missing information on the plot such as lens placement. It also automatically assumed that we wanted the ray trace of the marginal and chief ray. This is not bad quality of a LLM assuming that we need the marginal and chief ray for ray tracing, however, this is not useful in this application since the object is the illumination source. This makes the raytracing different due to raytracing the rays as if it is emitting from a point source. Further prompts allow us to adjust this, but the original code is a lot worse compared to the code provided by Claude.

Claude was recommended to us to use by one of the graduate students to ray trace in MATLAB. This made a big difference in the quality of code provided while using the same prompt. This code included many essential features of a well-rounded ray trace plot such as lens placement and a plot key. This was made a superior option as a starting point since it made it easy for us to include the lens apertures and ray trace the illumination through the annular stop as a point source This was essential in our raytracing since we wanted to ensure our rays were not missing the working diameter or our chosen lenses. We had to adjust our lens selections focal length and distances multiple times to achieve this.

### **5.4 Case Study 4**

Refer to appendix for the Case Study 4 given prompts and outcomes from ChatGPT.

At the beginning of this semester, I had limited experience with 3D printing, particularly for extremely small and precise objects like those required for this project. To deepen our understanding of additive manufacturing and explore strategies for high-precision printing, we utilized ChatGPT as a learning and research tool.

ChatGPT provided guidance on printer settings and features that could be adjusted to improve potential print accuracy and reduce the likelihood of errors. It helped explain concepts such as layer height, print speed, nozzle size, support structures, overhang angles, and cooling. This allowed us to understand how small changes in these parameters could influence the quality of a print, even before attempting any physical prints.

Through these discussions, we were able to identify the limitations and considerations relevant to designing small, detailed components, including the minimum wall thickness and feature sizes that the Bambu Lab X1 series printer can reliably produce. ChatGPT also helped us plan which settings and strategies would likely be most effective when we eventually move to actual printing.

Overall, using ChatGPT allowed us to explore the finer details of 3D printing in a comprehensive and understandable way, providing a helpful foundation for future experimentation and design decisions. It highlighted the value of combining AI guidance with critical evaluation and hands on practice for understanding and applying additive manufacturing techniques.

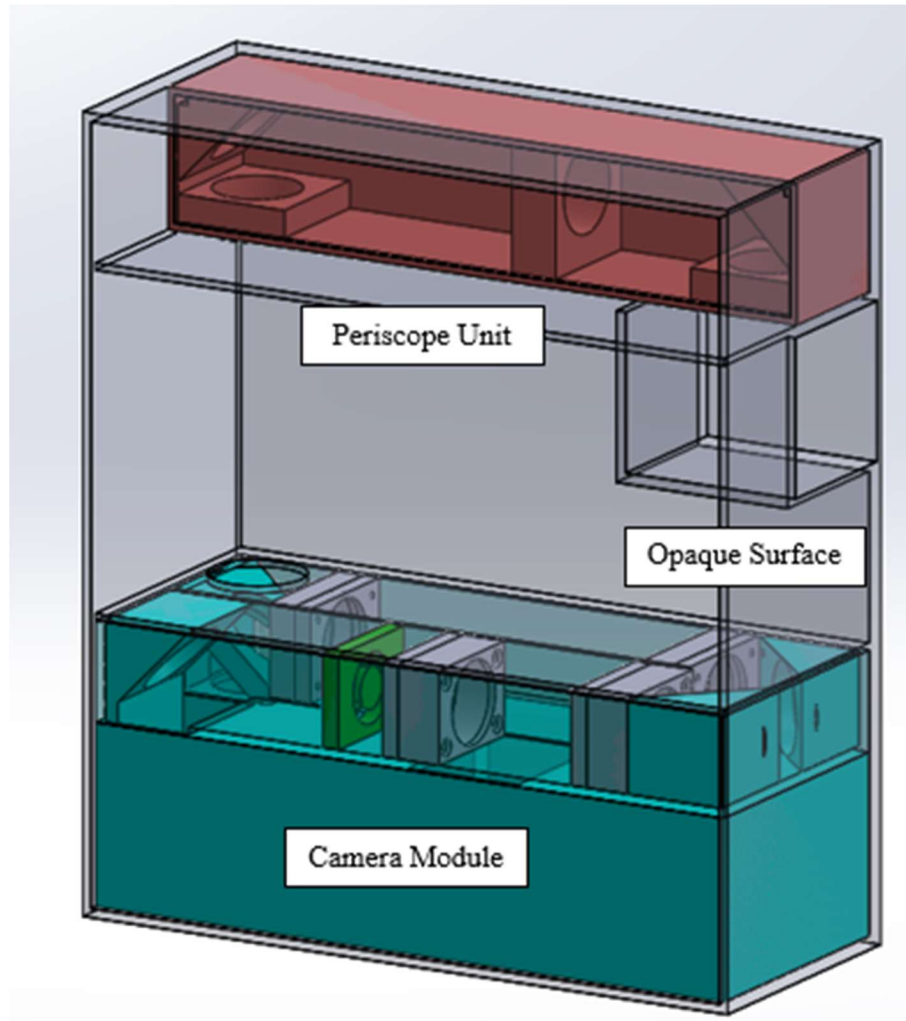
## **Chapter 6 – Hardware Design**

### **6.1 CAD Model**

#### **6.1.1 Case – Current Design**

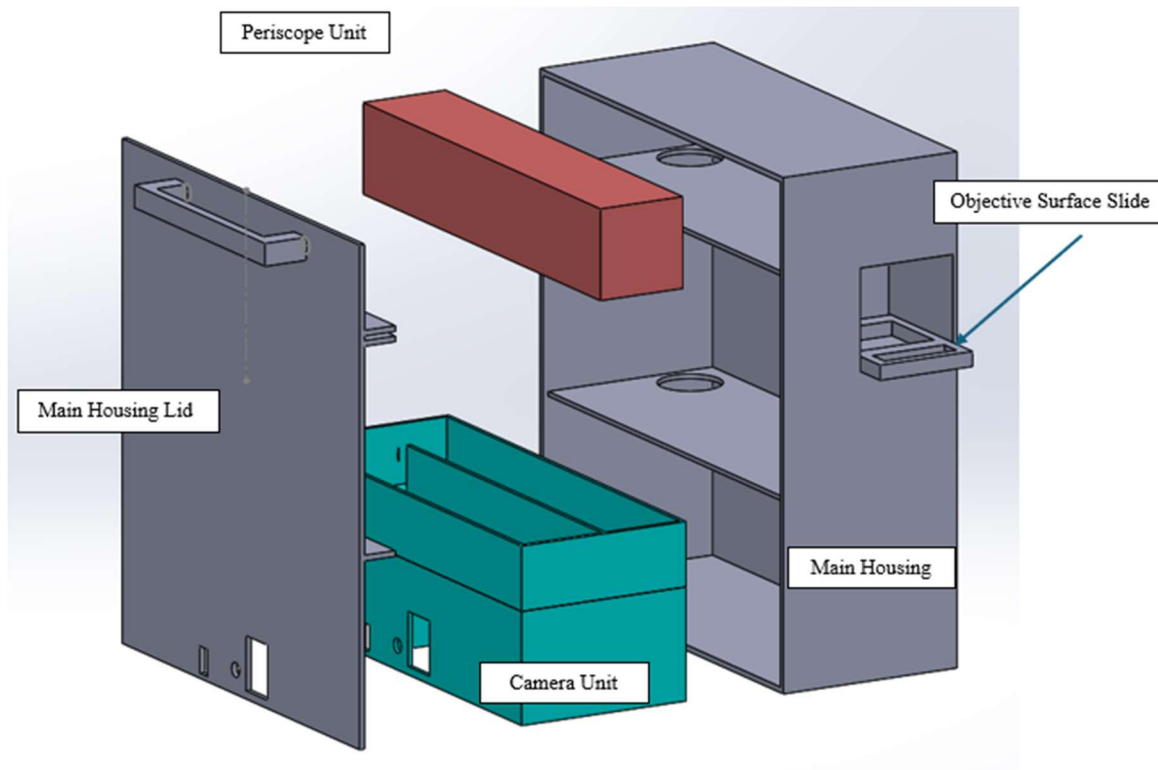
The case's primary function is to enclose both the periscope and camera units, ensuring a light tight seal necessary for accurate optical measurements. The design maintains the proper distances between the camera, periscope, and object surface, which is essential for requirements we have been provided with.

In addition to providing darkness, the case also contributes to the mechanical stability and alignment of the system. The geometry is structured to hold each optical component securely in its intended position, minimizing misalignment or vibration during operation. The current design emphasizes simplicity, rigidity, and modularity, allowing for straightforward assembly while ensuring that all internal spacing requirements are consistently maintained.



**Figure 6.1.1 Case closed**

This view illustrates the assembled case in its closed configuration, with all dimensions given in millimeters, showing the overall form of the housing. The blue sections represent the upper and lower portions of the camera unit, with the upper portion shown as transparent to display the internal lens and mirror mounts. The red section in the upper portion represents the periscope unit. Lastly, the transparent gray section represents the case itself as well as the opaque sample surface location.



**Figure 6.1.2 Case Expanded**

This view displays the case in an open configuration, revealing the internal structure and component layout. The blue blocks represent the upper and lower camera units, while the red block corresponds to the periscope unit. The small gray object in the upper right is the objective surface slide, which is used to move the surface in and out of view. This exploded view helps visualize how the internal assemblies fit within the housing and

illustrates the spatial relationship between the major optical components.

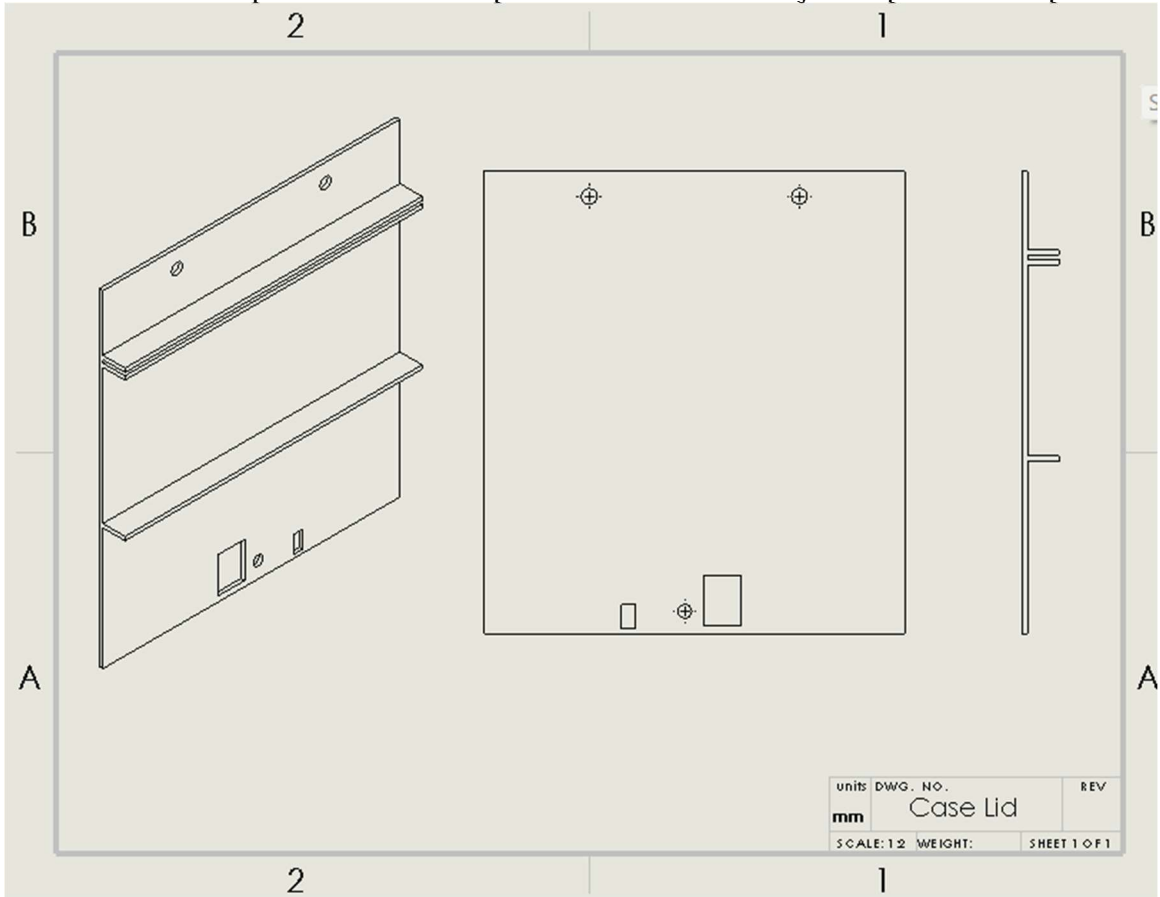
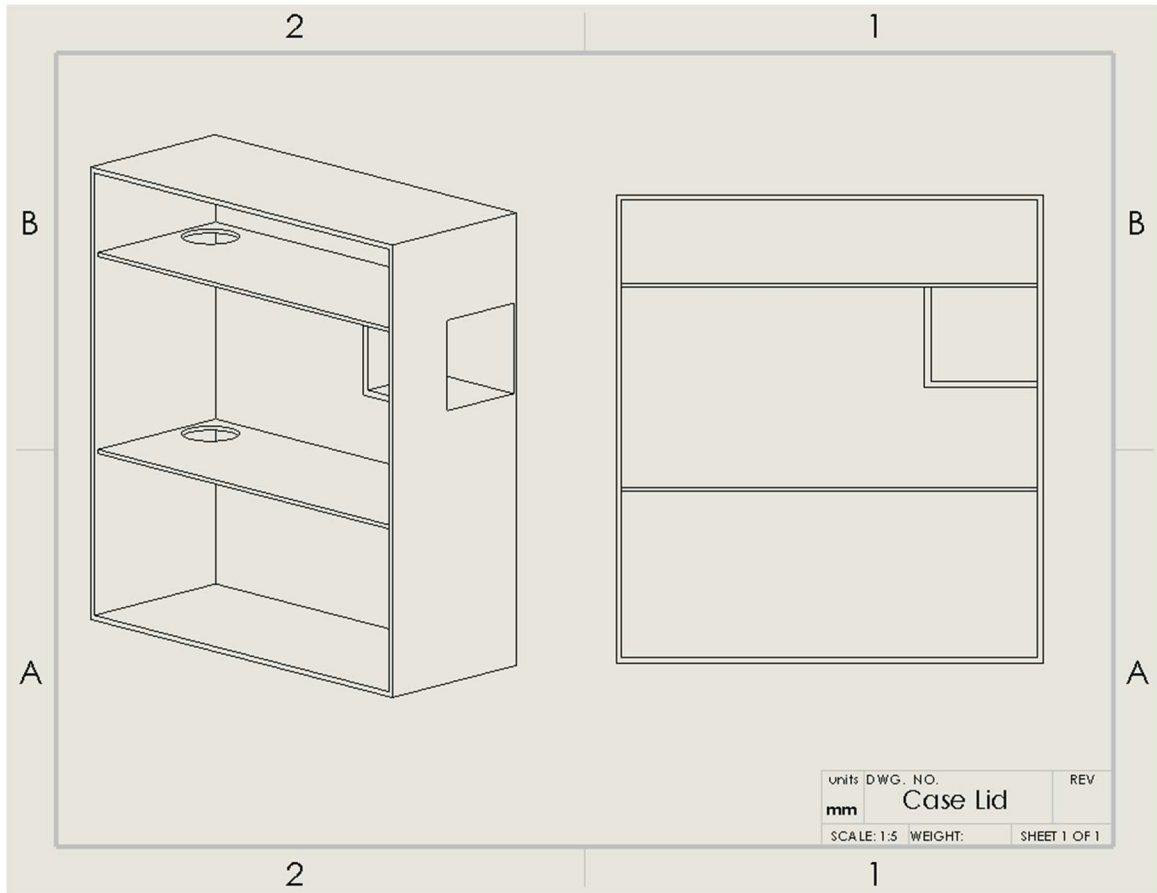


Figure 6.1.3 Case Lid Drawing (mm)

This drawing illustrates the dimensions of the casing lid, with all measurements provided in millimeters. The lid measures xx mm in length and xx mm in width. It is designed to slot securely into the main housing using 3 xx mm rectangular lips.

The extended portion of the lid serves a dual purpose: it allows the lid to slide into position after the periscope and camera units have been installed, ensuring precise alignment of all internal components, and it helps to maintain a light tight seal, preventing external light from entering the system.



**Figure 6.1.4 Main Case Drawing (mm)**

This drawing illustrates the design and layout of the main case. The case features a “L” shaped platform to hold the objective surface as well as 2 platforms that allow the lid and units to slid into place, ensuring proper alignment and a light tight seal. A large rectangular cutout is incorporated to accommodate the camera unit on the bottom (measuring xx mm × xx mm) and the periscope unit on the top (measuring xx mm × xx mm). These openings are positioned exactly xx mm apart, meeting the specified optical and geometric requirements.

The casing is designed with a minimum wall thickness of xx mm between the outer edge and any interior components to prevent light leakage and enhance structural rigidity.

## 6.1.2 Periscope Unit

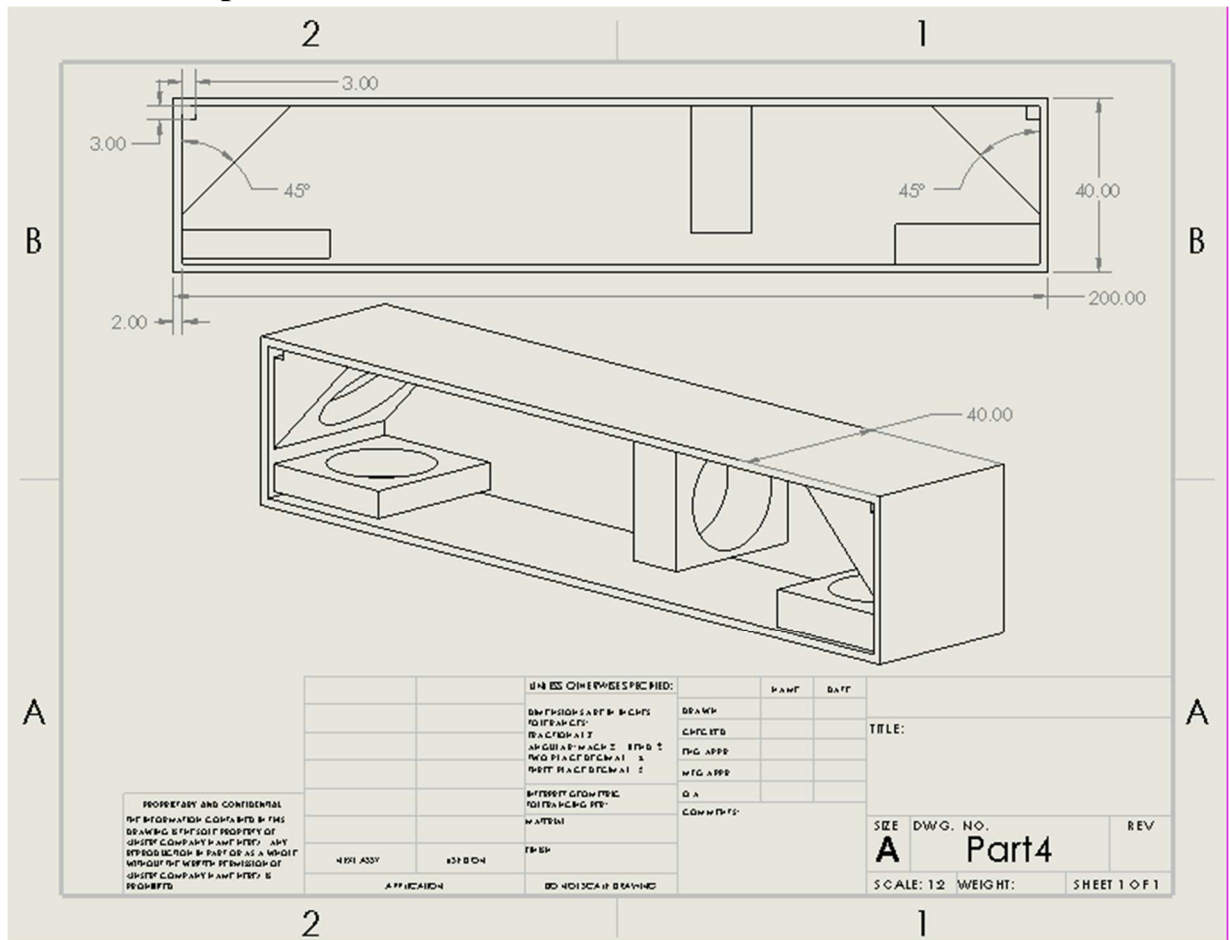


Figure 6.1.5 Periscope Drawing (mm)

This figure presents the main housing of the periscope unit, which is modeled as a  $xx \text{ mm} \times xx \text{ mm} \times xx \text{ mm}$  rectangular prism with  $x \text{ mm}$  thick walls. In the upper left and right corners, there are  $x \text{ mm} \times x \text{ mm} \times xx \text{ mm}$  slots designed to allow the periscope lid to slide securely into position, ensuring proper alignment and light tight assembly.

Additionally, the top corners feature  $xx^\circ$  angled holders that support the  $x \text{ mm}$  thick mirrors, positioning them precisely while maintaining sufficient clearance for the optical path. The design also includes three rectangular mounting blocks, each dedicated to securing one of the system's lenses in its correct position. Together, these features ensure structural rigidity, precise optical alignment, and ease of assembly within the periscope housing.

This figure illustrates the methods used to verify component fit and lens selection within the periscope unit. The process involved evaluating the spatial compatibility of each optical element to ensure that both the lenses and optical paths could be properly accommodated within the design constraints of the unit. By modeling each component in CAD and analyzing the optical layout, the team determined which lenses provided the required performance while still fitting within the available space. This approach ensured

efficient use of the internal volume and maintained precise optical alignment throughout the system.

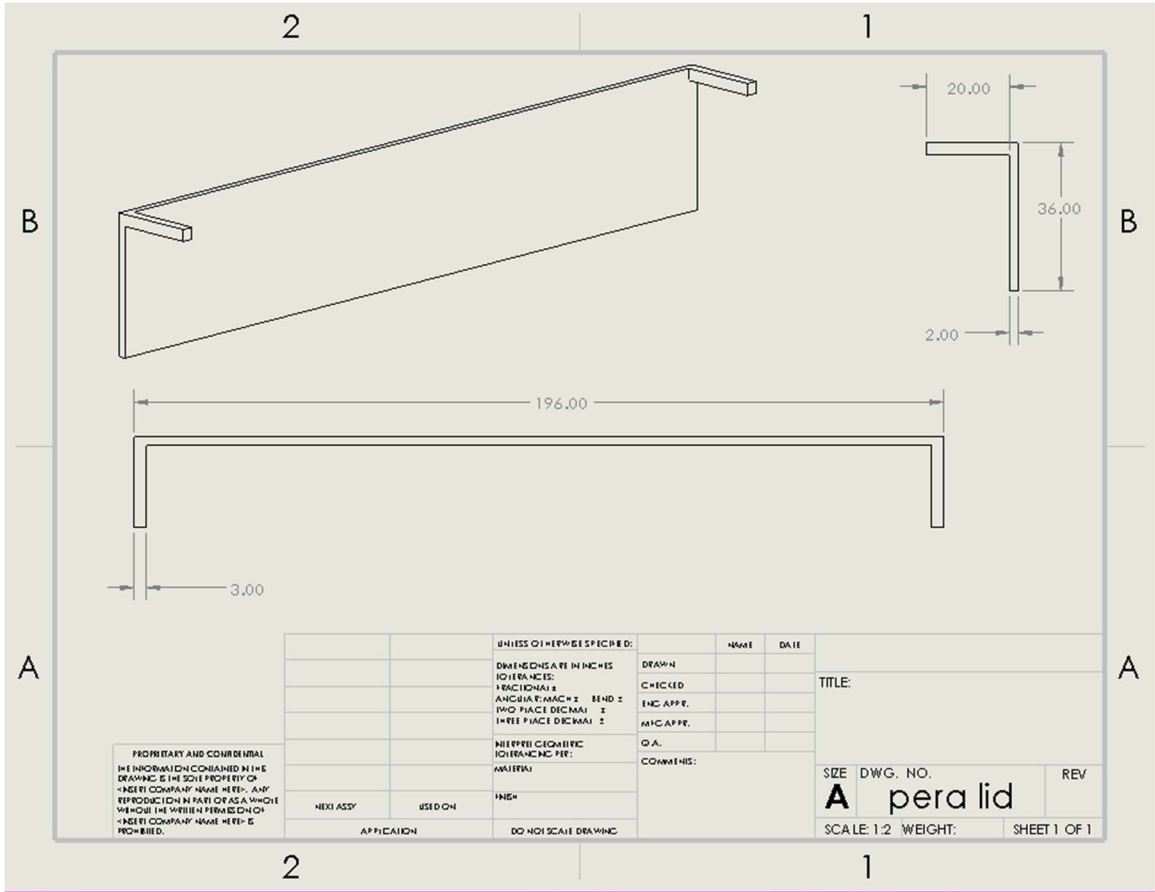


Figure 6.1.6 Periscope Unit Lid Drawing (mm)

This figure shows the periscope lid, which is modeled as a 196 mm × 36 mm × 2 mm rectangular prism. In the upper left and right corners, there are 3 mm × 3 mm x 20mm rectangular protrusions designed to slot into the corresponding openings in the main housing. These features ensure that the lid aligns accurately with the periscope structure, creating a secure fit that prevents unwanted light from entering the system.

### 6.1.3 Camera Unit

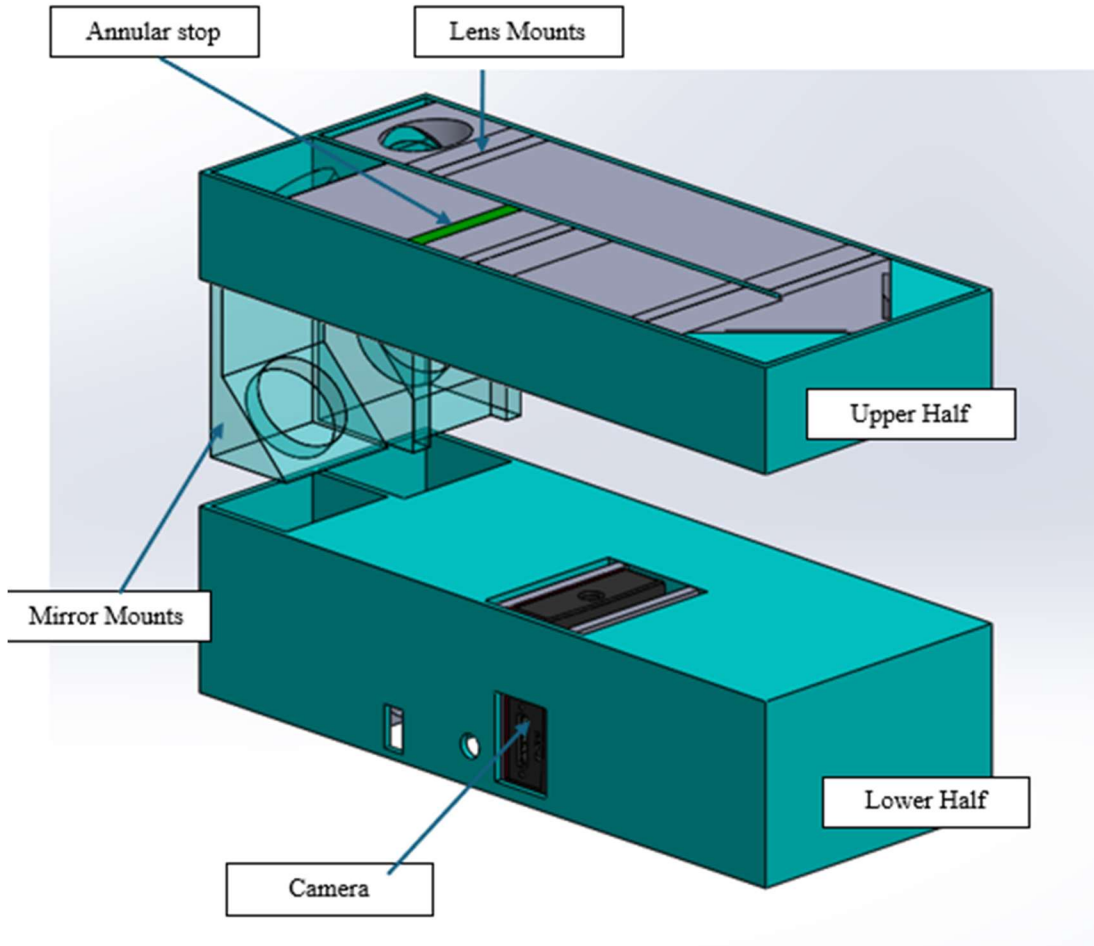


Figure 6.1.7 Camera Unit Top and Bottom View Drawing (mm)

This figure presents the top and bottom views of the mm x mm x mm camera unit. Like the periscope unit, the  $^{\circ}$  angled walls are designed to house the mirrors, while the rectangular mounting blocks serve as supports for the optical lenses. In the top view, a 36 mm  $\times$  5 mm slot is in the lower left corner to secure the annular stop right in front of the LED. Right above the LED is the beam splitter angled at negative degrees to the current view, the team would have liked to use the provide mount for the beam splitter but due to the size constraints it wasn't possible to fit. In the bottom view, a mm  $\times$  mm slot is provided in the upper left region to mount the mm tick camera with mm of adjustment room, while the adjacent area to the left will be used for the camera adjustment mechanism. This layout ensures proper optical alignment, compact integration, and ease of assembly for the entire imaging system.

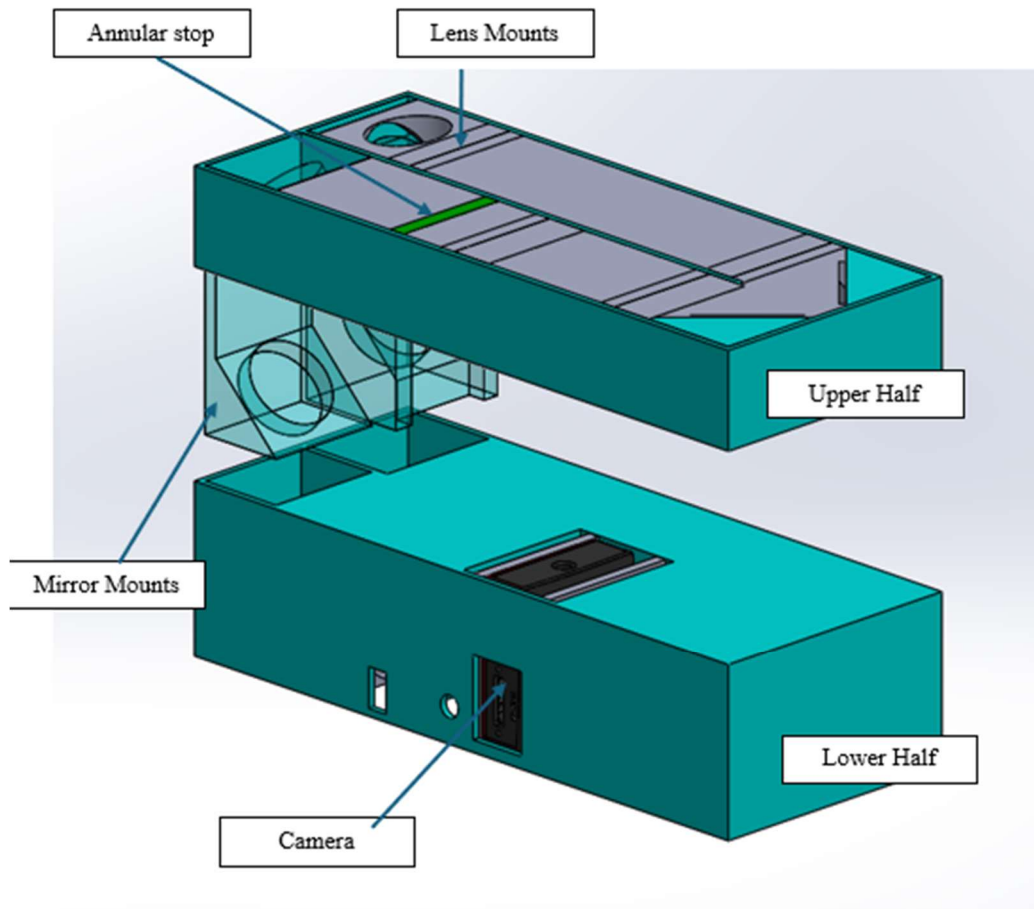


Figure 6.1.8 Camera Unit Isometric View (mm)

This figure provides an alternate view of the camera unit's bottom and top, offering a clearer visualization of the internal component arrangement. From this angle, the positions of the camera, annular stop, and beam splitter can be more easily identified. It also highlights the mounting locations and orientations of the lenses and mirrors, providing a more comprehensive understanding of how each optical element is securely supported and aligned within the housing.<sup>2</sup>

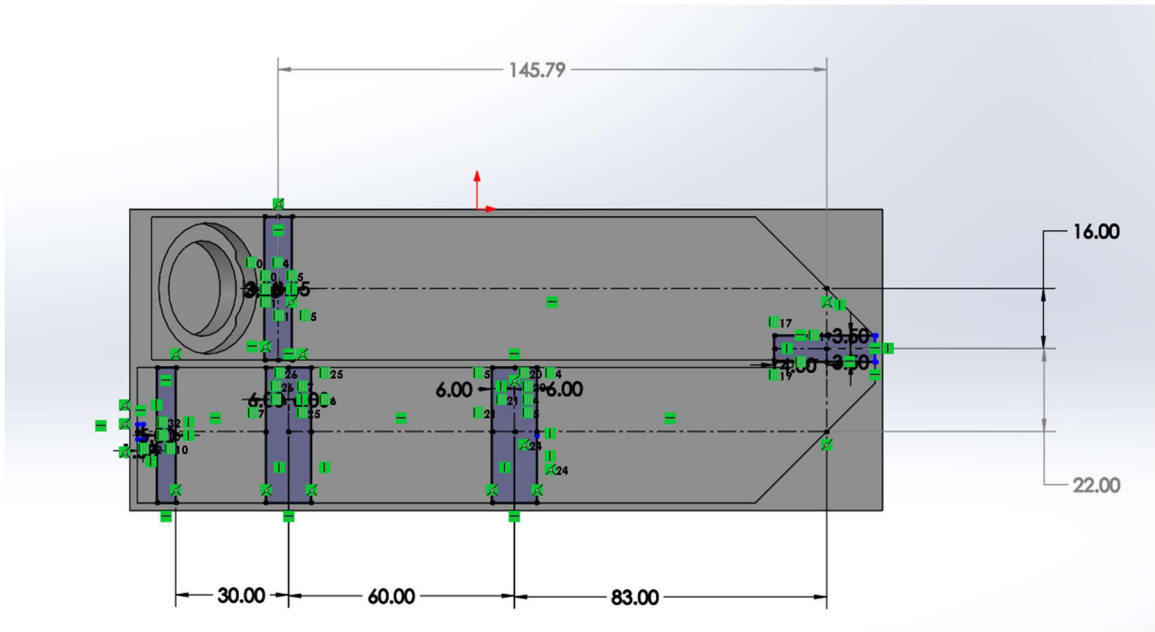


Figure 6.1.9 Camera Illumination Section Optical Measurements (mm)

This figure shows the SolidWorks sketch used to verify the distances and placements of the lenses and annular stop for the led path. Similar to the periscope design, this sketch was developed to ensure the optical components could fit and accurate optical alignment by accounting for factors such as lens thickness, mirror positions, and spacing requirements.

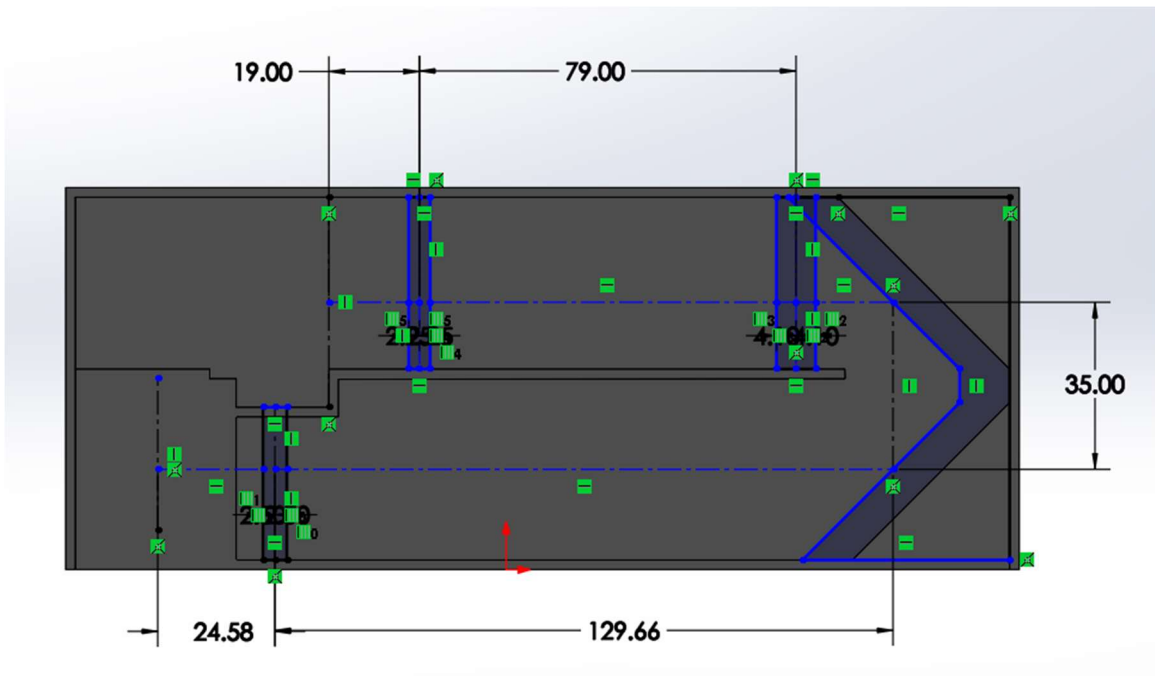


Figure 6.1.10 Camera Unit Detection Section Optical Measurements (mm)

The next figure presents a similar SolidWorks sketch, created for the camera path. Like the previous model, it was used to validate the positioning and spacing of optical

elements, ensuring that the camera, lenses, and mirrors were correctly aligned for a clear and focused image within the constraints of the housing.

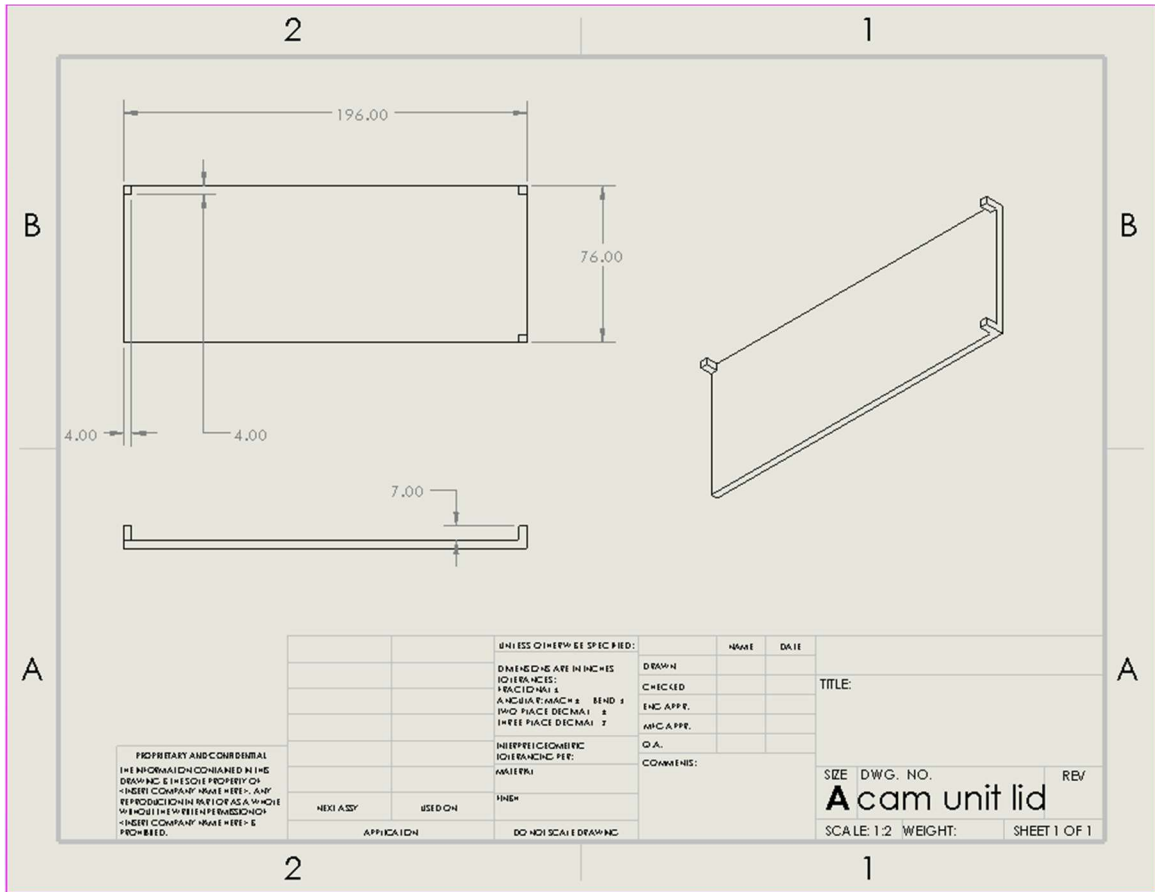
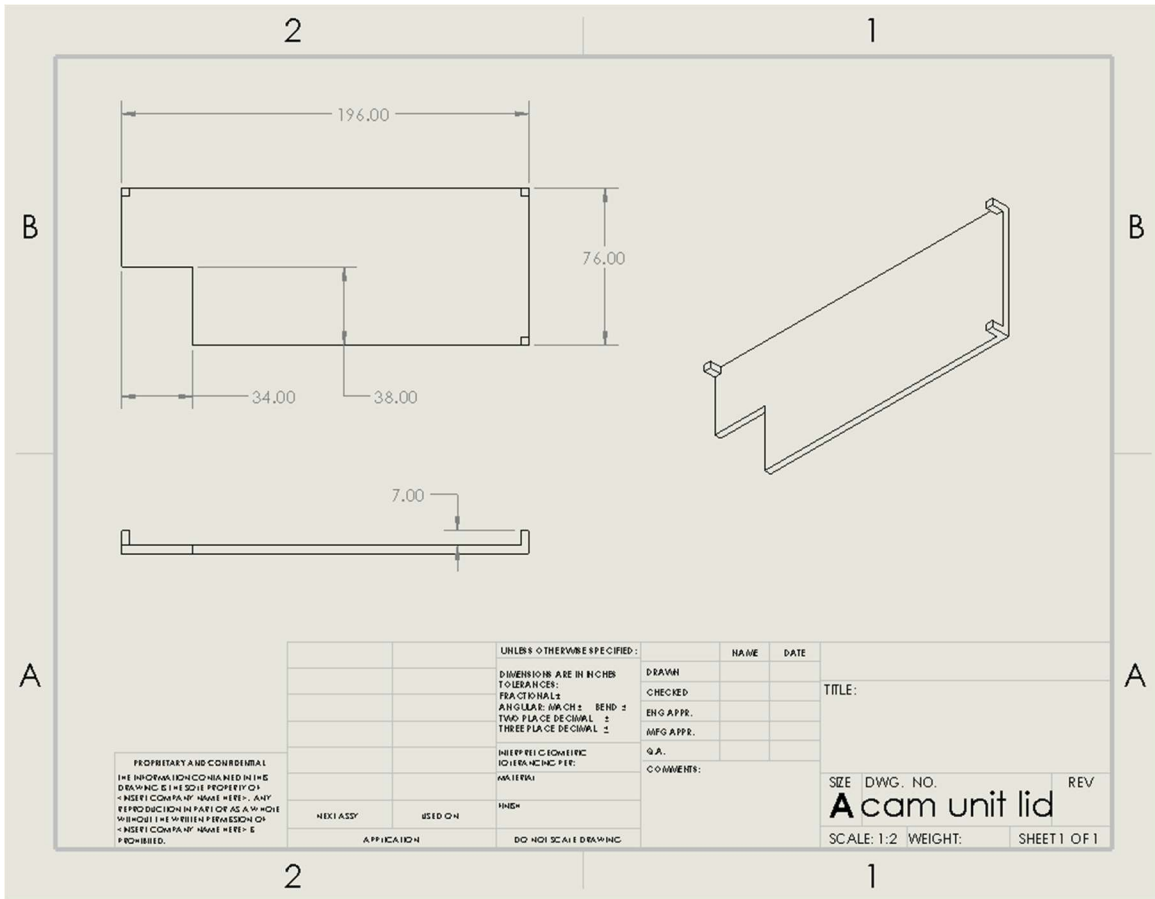


Figure 6.1.11 Camera Unit Bottom Lid (mm)

This figure shows the lid for the camera side of the housing. It is designed to slot securely into place using three 4 mm × 4 mm × 7 mm rectangular prisms, which ensure proper alignment and create a sealed fit between the lid and the main housing. This design helps maintain internal darkness and prevents misalignment during assembly or operation



**Figure 6.1.12 Camera Unit Top Lid (mm)**

This figure shows the lid for the LED section of the camera housing. It is identical in design to the camera side lid, except that it includes a 38 mm × 34 mm opening positioned directly above the beam splitter. This opening provides a path to the periscope unit.

### 6.1.4 Annular Stop

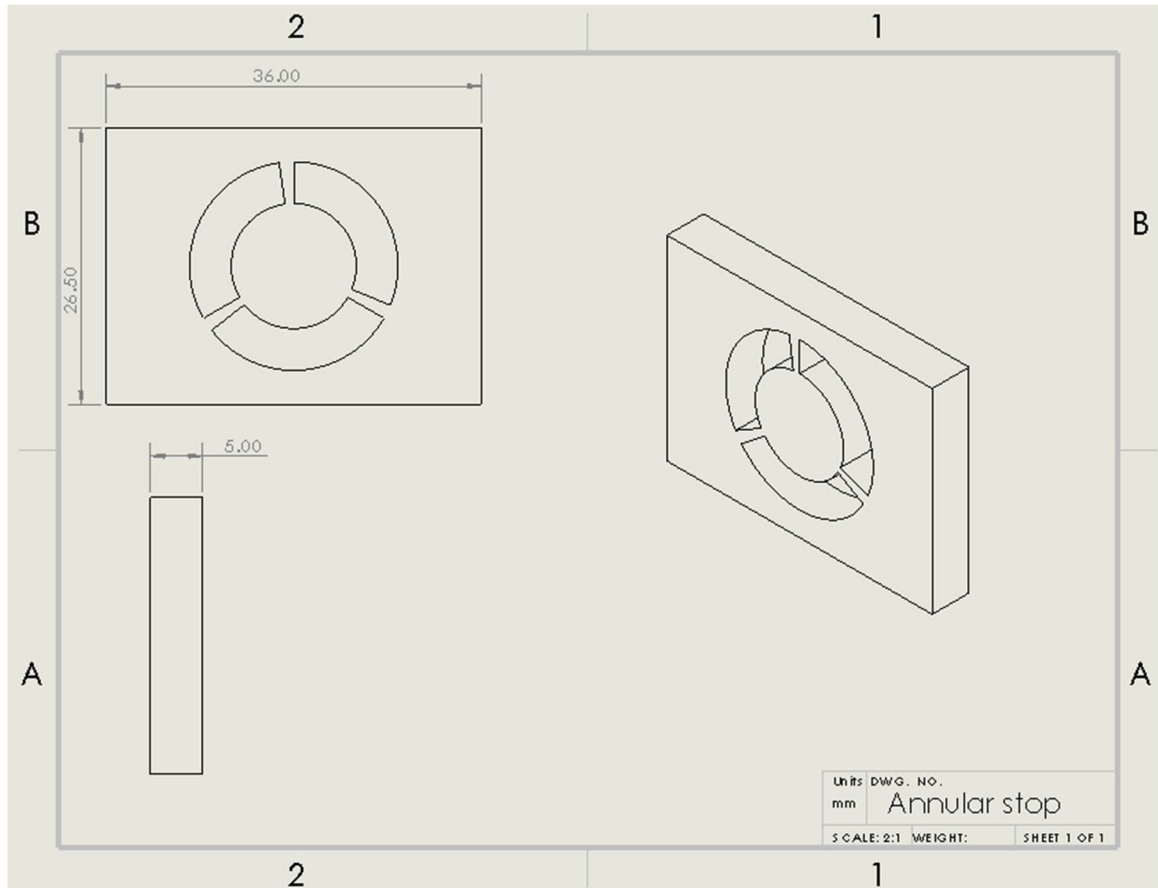


Figure 6.1.13 Annular Stop (mm)

This figure shows the custom annular stop designed specifically for our optical system. The annular stop features an inner radius of 5 mm and an outer radius of 10 mm, supported by three equally spaced beams. Each beam spans approximately  $6^\circ$ , or about 1.5 mm in width, providing structural stability while maintaining an unobstructed optical path.

The stop is incorporated into a  $26.5 \text{ mm} \times 36 \text{ mm} \times 5 \text{ mm}$  rectangular prism, allowing it to slot precisely into position in front of the LED. This design ensures accurate placement, and consistent alignment.

## 6.2 Optical Illumination Sub-System

The optical illumination system utilizes a total of fifteen optical components that are broken down into seven achromatic doublet lenses, five mirrors, a beam splitter, an LED and an annular stop. The main goal of our illumination system is to illuminate the opaque surface at an oblique angle to enhance the contrast of the sample. We will achieve this by creating an annular light beam that slightly magnifies the annular stop mask in the camera module then demagnifies the periscope unit. This magnification then demagnification is essential in our project design since the periscope unit has the detection and illumination optical paths that propagate in opposite directions. We will start breaking down the illumination detection path in detail starting at the illumination source.

## 6.2.1 LED Optical Power

The illumination source is a high-power green LED with the power of 10 Watts. We chose green since it has a low number of aberrations in achromatic doublets since it is in the middle of the visible spectrum. This high power is necessary due to the low NA in our detection system and the annular stop that is placed right in front of the LED. Since the annular stop blocks the center of the LED, a lot of the power is already lost. We wanted to include a collimating lens between the LED and the annular stop to limit the beam divergence of the LED's spatial coherence characteristics. Unfortunately, this was not possible due to the dimensional constraints of the camera module. We are planning to compensate for this by adjusting the annular stop size that compensates for the beam divergence of the illumination beam into the first achromatic doublet lens.

## 6.2.2 Annular Stop Size

The annular stop is the only optical component that is not available to purchase off the shelf. To design the annular stop, we ray traced the detection system on MATLAB to figure out the size of the masked annular stop on the Fourier plane of the sixth lens illuminate our desired spot size at opaque angles. This is because the “objective lens” in the detection system is the same as the “condenser lens” in the illumination system. For proper darkfield illuminated microscopy, the desired angle the illumination source is incident on the opaque surface is supposed to be in the range between  $10^\circ$  to  $45^\circ$ . Unfortunately, due to the specified dimensions of the periscope unit, the condenser lens diameter is limited to one inch. This is not ideal but should still produce enough scatter and reduce bright field reflections since our sample is very large compared to traditional microscopy systems. The following plot shows the raytraced detection path that was plotted via MATLAB.

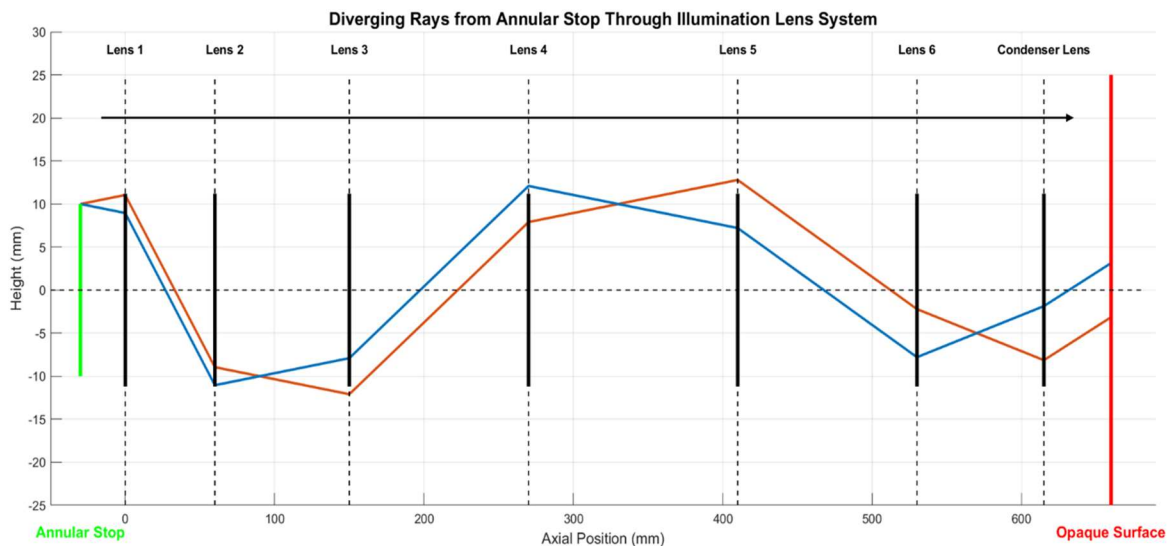


Figure 6.2.1 Illumination System Ray Trace

To calculate the annular stop inner radius relationship with the angle of illumination, we used the following equations.

$$\theta_{detection\ half-angle} = \tan^{-1} \frac{R_{lens\ WD}}{focal\ length}$$

$$R_{inner\ mask} = f \tan \theta_{detection\ half-angle}$$

Due to the total magnification at lens 6 Fourier plane being 0.5, we divide the dimensions above by 0.5 to calculate the actual annular stop size.

$$R_{PO} = \frac{Outer\ radius}{Magnification}$$

$$R_{PI} = \frac{Inner\ radius}{Magnification}$$

### 6.2.3 Illumination Lenses Optical Path and Magnification

The mask illuminated image is then relayed to the condenser lens Fourier plane via the combination of three 4f systems. A single 4f system is composed of two lenses, where the distance between the first lens and the object is the focal length of the first lens, the distance between the first and second lens is the sum of their focal length and the distance between the second lens and the image is the focal length of the second lens. These systems can be used to relay images with desired magnification and use Fourier optic techniques for spatial filtering. The magnification of each 4f system is determined by dividing the second lens's focal length by the first lens focal length. This results into a magnified flipped image. Multiplying the magnification of each 4f system determines the total magnification of the system.

The biggest challenge in finding the proper set of 4f systems is the lenses in the periscope unit combined with the dimensional system specification of having 150 mm of free space between the periscope unit and the camera module. The lenses in the periscope unit work two different 4f systems that have opposite direction of propagation. Since we must demagnify the image of the opaque surface and demagnify the annular stop mask to the Fourier plane of the sixth lens, we must compensate for the magnification that occurs from the detection system demagnification. This brings up the problem of rays diverging outside the achromatic doublet lenses aperture. Due to the dimensional constraints, the biggest lens apertures we can fit in each individual optical path is one inch.

This is the reason for the first 4f system that is closest to the annular stop. Due to the limitations of the lens diameter, the first lens must have the smallest focal length possible to try to illuminate the light rays missing the working distance of the rest of the lenses. When we tried to implement a larger focal length lens, there was too much divergence in the LED's optical path due to its spatial coherence characteristics. This can be compensated for by measuring the total divergence of the illumination source to the first lens, then compensating the annular stop size to mask the proper mast image into the Fourier plane of the condenser lens. This decreases the total optical intensity that illuminates the system, which is not ideal. Larger focal length lenses also take too much space and cause rays to miss the working diameter of other lenses in the system. The second lens's focal length can be larger, but that will cause a magnified image, which will cause rays to miss the following lenses working diameter to their higher focal lengths.

The second 4f system's biggest challenge is relaying the annular stop mask across of free space dimension provided by our sponsor. The goal of this 4f system is to relay the image into the Fourier plane onto the back lens of the periscope unit. Since the second lens of this 4f system must have a long focal length. This combined with the lack of magnification ability of this 4f system requires the first lens to also have a long focal length. The lack of magnification ability is due to the very large distance this 4f system is relaying the mask, which causes many rays to diverge off the optical path then miss the working aperture of the lenses in the periscope unit.

The last 4f system caused the most problems while designing the illumination system. Since these two lenses are on two different 4f systems in the detection subsystem, the compensation of this intense magnification only needs to be overcome in the illumination path. Their focal lengths were also challenging to adjust since the adjustments affect three different 4f systems. Each of these systems has severely different magnification goals and challenges.

The following is a mathematical breakdown of the focal lengths of the lenses in the illumination path.

$$f_4 + f_5 \geq 0 \text{ mm} - 200 \text{ mm}$$

$$f_5 + 2f_6 + f_7 \leq \text{Periscope Path length} \approx 0 \text{ mm} - 200 \text{ mm}$$

$$2f_1 + 2f_2 + 2f_3 + 2f_4 + 2f_5 + 2f_6 + f_7 < \text{Total available path length} \\ \approx 0 \text{ mm} - 800 \text{ mm}$$

$$f_1 = 10 \text{ mm} - 90 \text{ mm}, f_2 = 10 \text{ mm} - 90 \text{ mm}, f_3 = 10 \text{ mm} - 90 \text{ mm}, f_4 \\ = 10 \text{ mm} - 90 \text{ mm}, f_5 = 10 \text{ mm} - 90 \text{ mm}, f_6 = 10 \text{ mm} - 90 \text{ mm}$$

$$f_7 = 10 \text{ mm} - 90 \text{ mm}$$

$$M_{I1} = \frac{f_2}{f_1} = 0 - 2$$

$$M_{I2} = \frac{f_4}{f_3} = 0 - 2$$

$$M_{I3} = \frac{f_6}{f_5} = 0 - 2$$

$$M_{IT} = M_{I1} \times M_{I2} \times M_{I3} = 0 - 2$$

The main objective of the beamsplitter is to reflect the illumination optical path to the periscope unit and transmit the detected image to the detection subsystem inside the periscope. To achieve this, the beamsplitter is placed at 45° that is facing up toward the periscope unit. The main challenge with selecting the proper beamsplitter was the reflection to transmittance ratio, and the working diameter. We chose to use a pellicle beamsplitter due to its high R:T ratio and the biggest size we could fit with our dimensional constraints is one inch in diameter.

The main objective of the mirrors is to expand the optical path of the illumination subsystem and reflect the illumination optical path inside the periscope unit unto the condensing lens. Similar to the beamsplitter, we were mainly concerned with the reflection and dimensional parameters of these mirror. We have decided to use front-surface plano broadband dielectric mirrors due to their high reflectivity and availability in one inch diameter sizes, which is the biggest size we could fit due to our dimensional constraints.

## 6.3 Optical Detection Sub-System

Considering that we must demagnify our object and that a large part of our design has a common illumination and detection path. It made sense to us design wise to start with the detection path and continue with the illumination path after. This way we would not waste time redesigning our individual systems. Given the specifications from our sponsor, we calculated a series of parameters that would be necessary to achieve a functional imaging system.

### 6.3.1 Detection Target Magnification

The objective magnification for a system is calculated with the following equation. For our system, the object area chosen by our sponsor is a sponsored specified distance wide and the sensor size is around 3.45 mm. To map all of the object to the camera sensor you will need to find their ratio. The required magnification is between 0.2x and 0.5x.

$$M_{required,detection} = \frac{Sensor\ size}{Object\ size}$$

$$0.2x \leq M_{required,detection} \leq 0.5x$$

Considering we are using a series of 4f systems, the magnification of a single 4f system could be found by the following equation; Where  $f_2$  is the lens furthest from the object and  $f_1$  is the lens closest to the object.

$$M = \frac{f_2}{f_1}$$

When putting multiple of these 4f system in series, the total required detection path magnification can be found by equation.

$$M_{total,detection} = M_{d1}M_{d2}M_{d3} = \frac{f_2 f_4 f_6}{f_1 f_3 f_5}$$

### 6.3.2 Detection Lenses Focal lengths

Due to the size constraints outlined by our sponsor as well as their desire to keep all part off the shelf. We discovered a series of conditions that limits the solution space for the focal lengths of all lenses. A more detailed explanation of the limitations and derivations of these conditions can be found in Chapter 4.

$$f_1 > Sponsored\ specified\ distance$$

$$f_3 + f_4 \geq Sponsored\ specified\ distance$$

$$2f_1 + 2f_2 + f_3 \leq \textit{Sponsored specified distance}$$

$$2f_1 + 2f_2 + 2f_3 + 2f_4 + 2f_5 + 2f_6 < \textit{Total available detection path length}$$

Using these four conditions in addition to the target magnification specified above, we found only a couple of solutions that allowed us to meet all these requirements.

$$10 \textit{ mm} \leq f_1 - f_6 \leq 90 \textit{ mm}$$

Using these parameters, we can calculate the magnification of each 4f system and multiply them to get the detection path magnification.

$$0x \leq M_{d1} = \frac{f_2}{f_1} \leq 2x$$

$$0x \leq M_{d2} = \frac{f_4}{f_3} \leq 2x$$

$$0x \leq M_{d3} = \frac{f_6}{f_5} \leq 2x$$

$$0.2x \leq M_{dT} = M_{d1}M_{d2}M_{d3} \leq 0.5x$$

The magnification is calculated to be within our expected range of between 0.2x and 0.5x.

### 6.3.3 Numerical Aperture

Numerical Aperture is a measure of how much light your system can receive at a given object distance. For a single lens the numerical aperture is found by the following equations, where n is the refractive index, theta is the half cone acceptance angle, D is the diameter of the lens, and f is the focal length. For imaging systems it is usually beneficial to maximize the numerical aperture. Therefore, we find D by using the largest diameter off the shelf lenses that could fit all the way through out design.

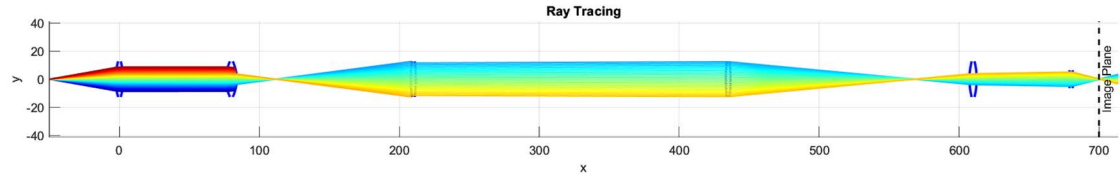
$$NA = n \sin(\theta_a)$$

$$\theta_a = \arctan\left(\frac{D}{2f}\right)$$

$$NA_{object} = n \sin\left(\arctan\left(\frac{D}{2f}\right)\right)$$

Using lens diameter equal and focal length our first lens being, we calculate the object numerical aperture to be 0.05-0.3.

Although this would be true if we were only using the collection lens f1, As our optical system has more parts, numerical aperture can decrease further in our system. To find the true numerical aperture for our system, we used our ray trace in MATLAB and tracked the height of the highest ray that made it to the camera sensor.



**Figure 6.3.1 Ray Tracing of the Detection Path**

With  $h$  being the height of the highest ray that makes it through your system at the object. We calculate the true object numerical aperture with a slightly modified version of the same equation from earlier..

$$NA_{Object,Limited} = n \sin \left( \arctan \left( \frac{h}{f} \right) \right)$$

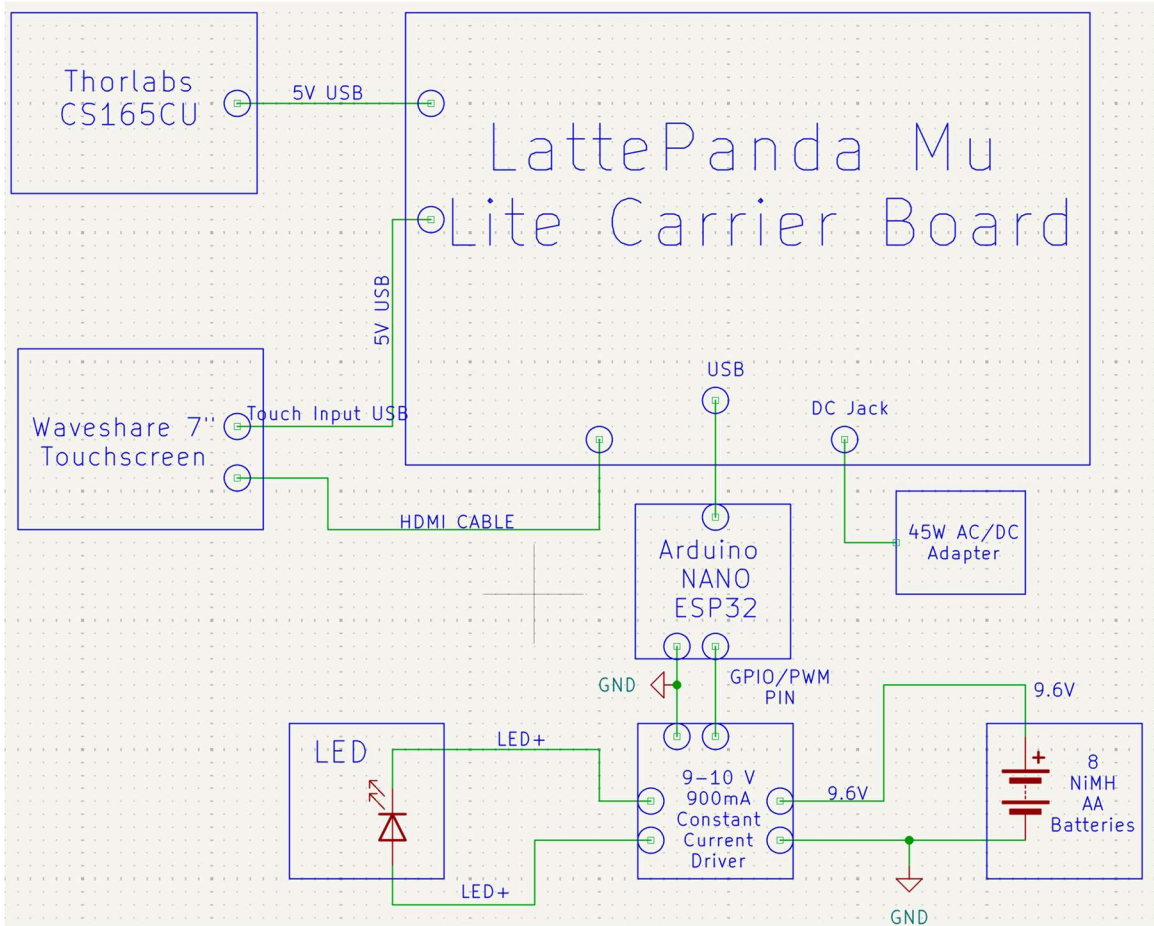
### 6.3.4 Rayleigh Criterion

Due to diffraction, there is a theoretical limit to how small a feature you can resolve. Where  $d$  is the smallest resolvable feature, for our calculated NA. We calculate the smallest resolvable feature with this equation.

$$d = \frac{0.61\lambda}{NA}$$

Since the smallest feature we are looking to image is larger than that limit, our system is not diffraction limited. Considering our diffraction limited smallest feature for our calculated NA is 4.5 micron. A larger particle would appear on the camera.

## 6.4 Board to Board Electrical Diagram



**Figure 6.4.1 Board to Board Electrical Diagram**

Given that there will be no PCB design, the diagram above is used to show how the electronic components will be connected to each other in the project. Firstly, the LattePanda Mu will be powered via a 45W AC/DC power adapter connected to the DC jack. Power will then be supplied by the Lite Carrier Board to the Thorlabs CS165CU and Waveshare 7" Touchscreen via the USB 3.2 port, and the Arduino Nano ESP32 via the USB 2.0 port. Furthermore, the USB connections for these three peripherals double as I/O and data transfer connections. Additionally, an HDMI connection is made between the Waveshare display and the Lite Carrier Board in order to output video.

For the LED subsystem, there is a series of connections made. The batteries are placed in a battery holder where ground and the 9.6V are connected to the GND and VIN pins respectively on the driver. LED+ and LED- are the connected to the respective polarities on the LED. Finally, the Arduino is connected to the to the PWM pin on the driver via its D2 GPIO pin, as well as connected ground between the two components.

# Chapter 7 – Software Design

## 7.1 Software Overview

For the P.I.D.I. project, there are 2 main implementations of software. The first of these is the user interface which will be embedded into the touch-screen display. The second is the image processing algorithm, which will be embedded into the computer module.

## 7.2 User Interface

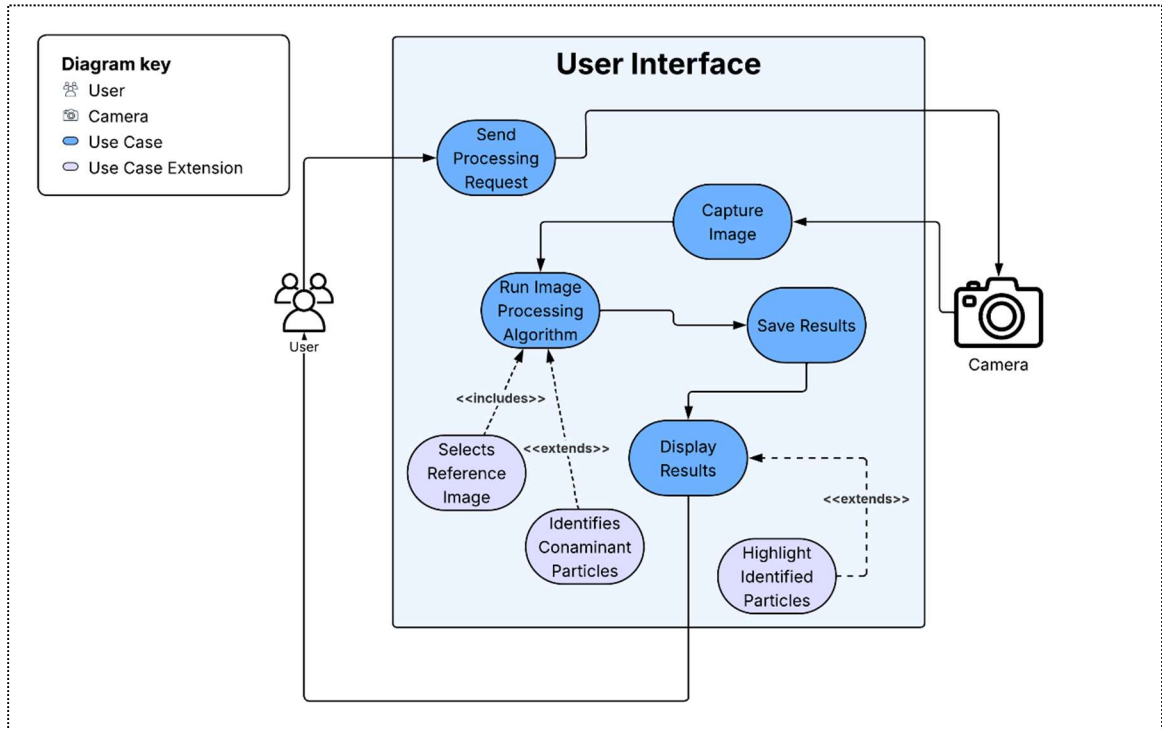
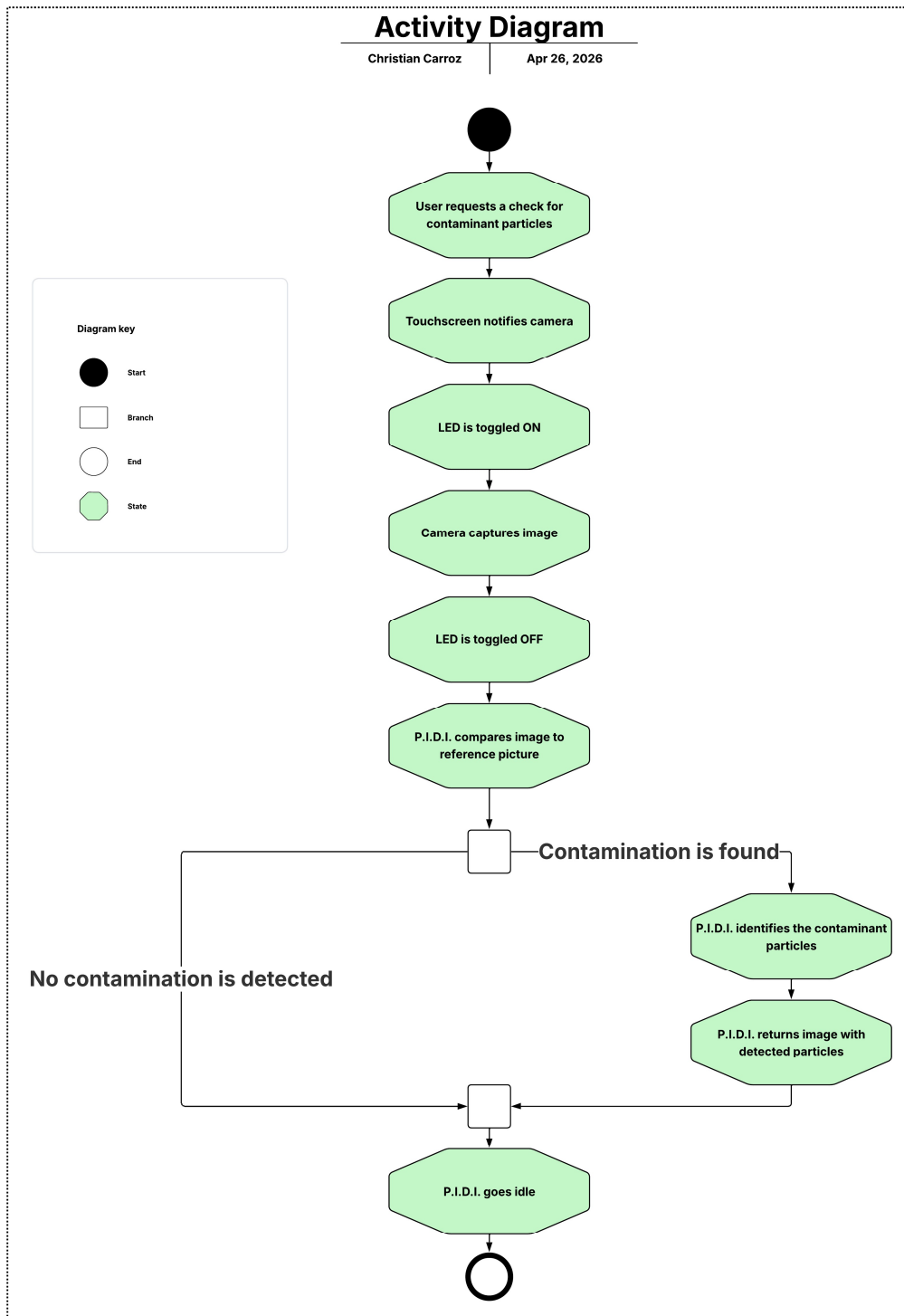


Figure 7.2.1 Use Case Diagram

The use case diagram shows the intended functions of the P.I.D.I. system. We see how the user would send the image processing request through the Touch Screen. Camera would capture the image and send it back to the MCU to be processed. Once the processing is done, the image is displayed on the touch screen.



**Figure 7.2.2 Activity Diagram**

The activity diagram shows how the process the system would go through when running the algorithm. It shows the flow of information from the peripherals into the MCU.

## 7.3 Image Processing Algorithm

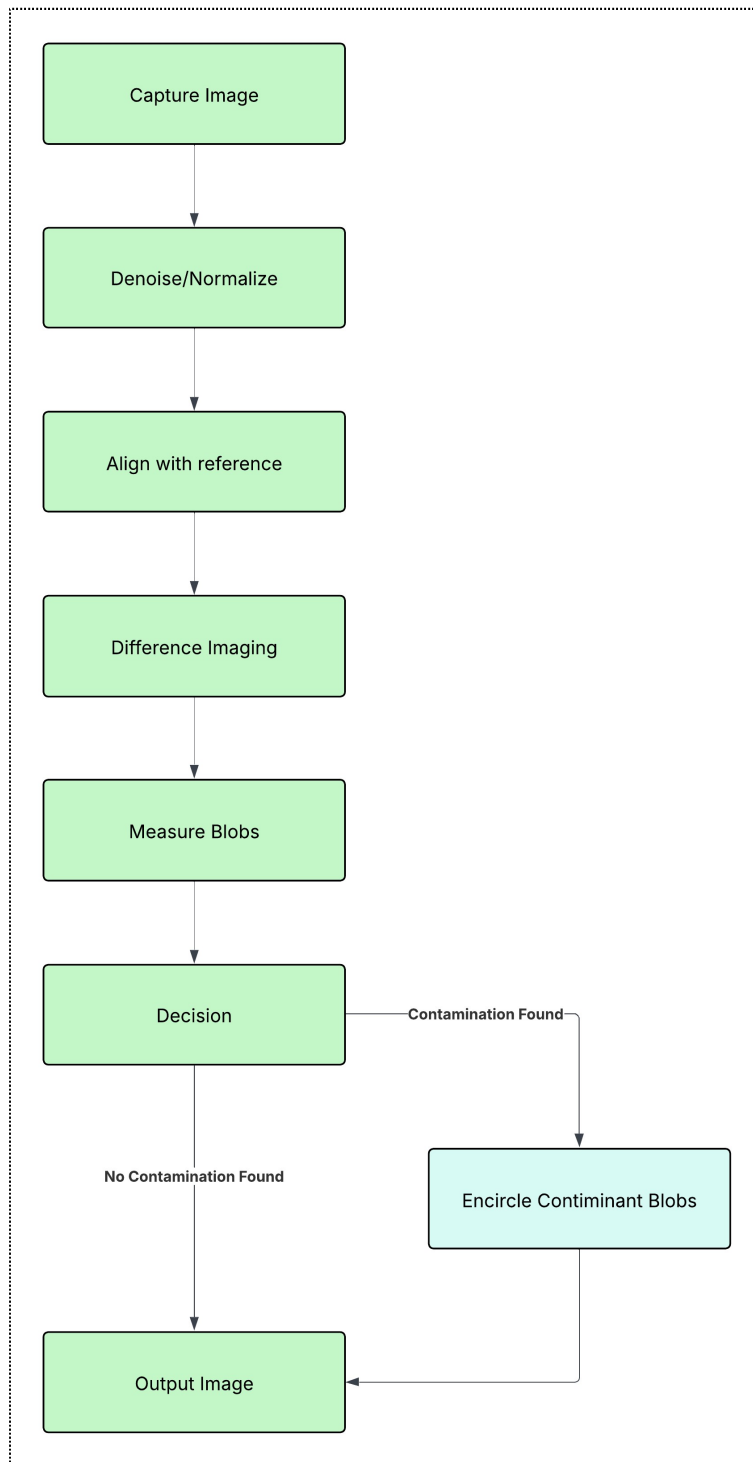


Figure 7.3.1 Image Processing Algorithm Block Diagram

This diagram is just a simple flowchart that identifies the steps that will be taken by the algorithm itself. It also visualizes where it makes any decisions and the actions taken by it.

## 7.4 Particle Inspector User Interface

Below will be 4 images of what the UI will look like. We will have 4 screens including home, reference, capture and analyze, and results.

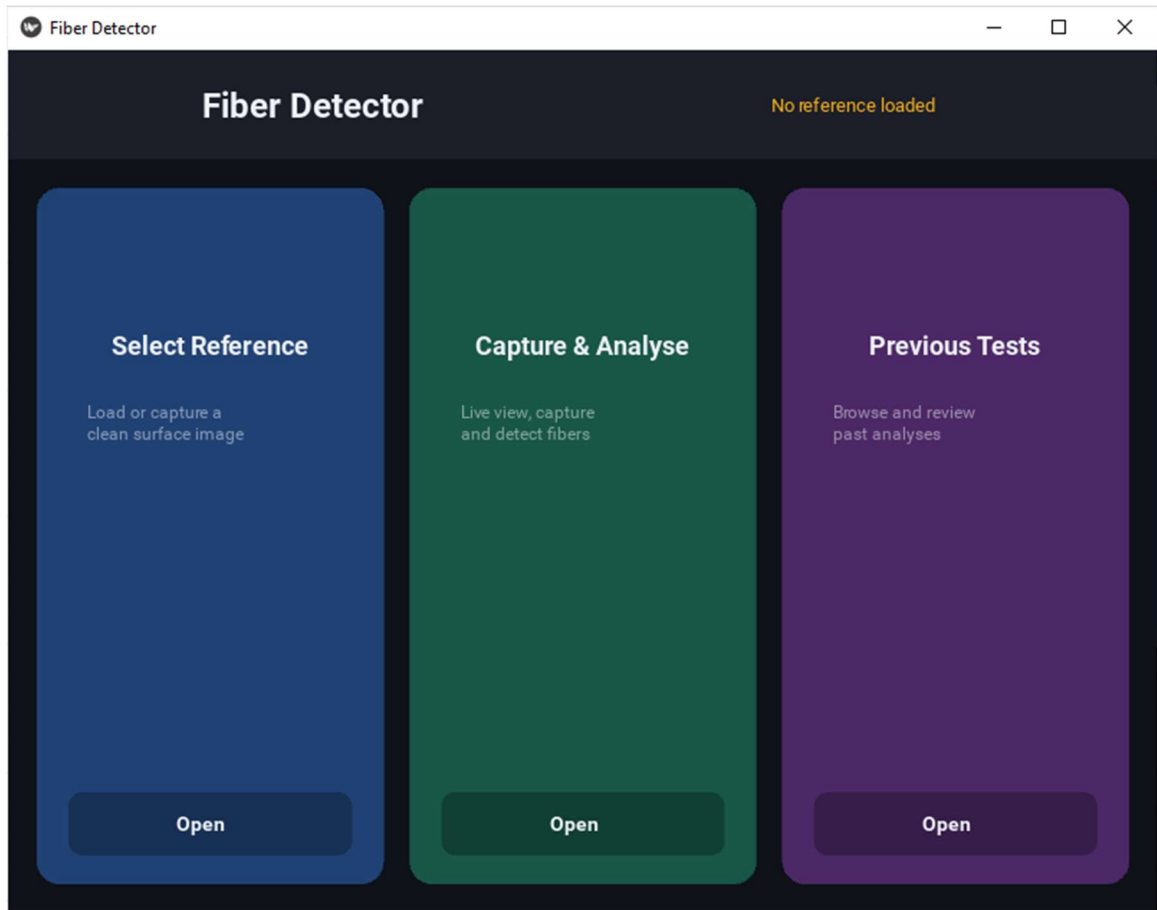
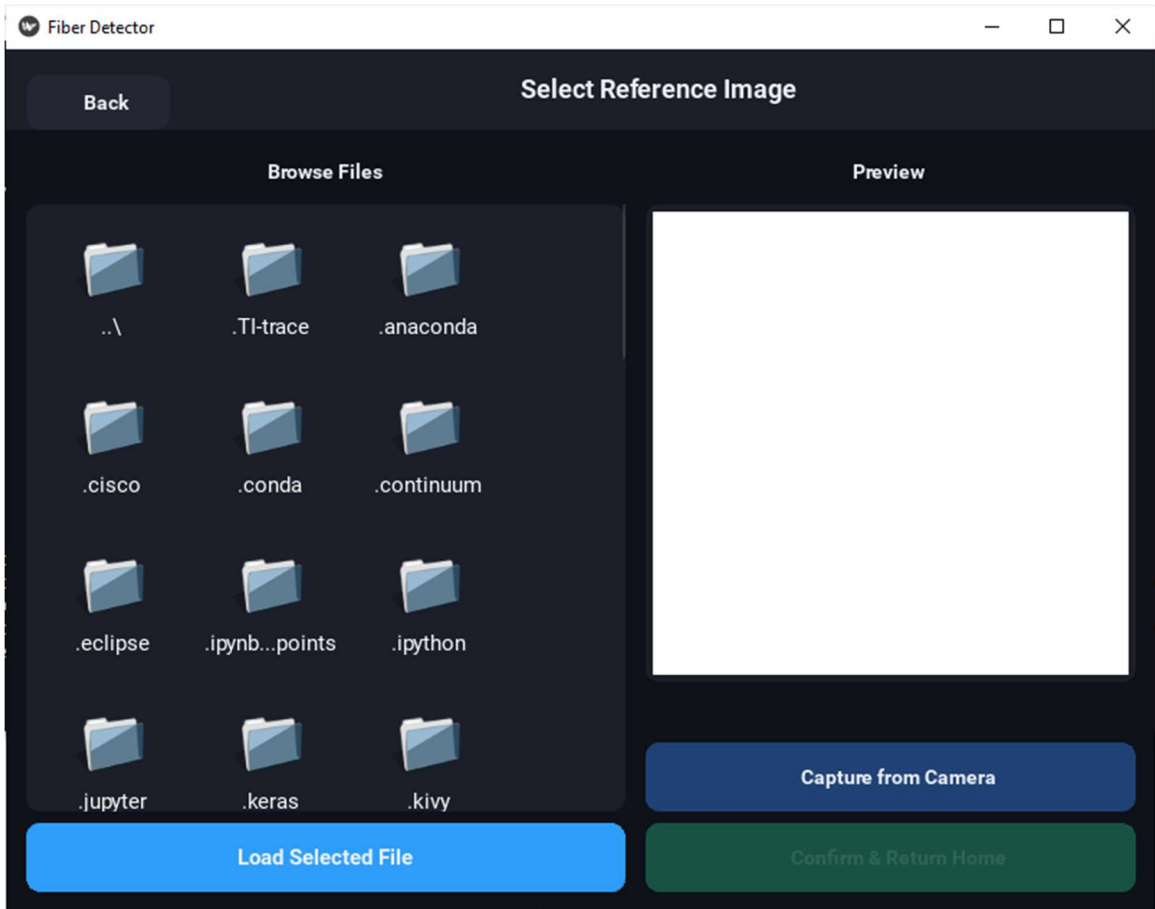


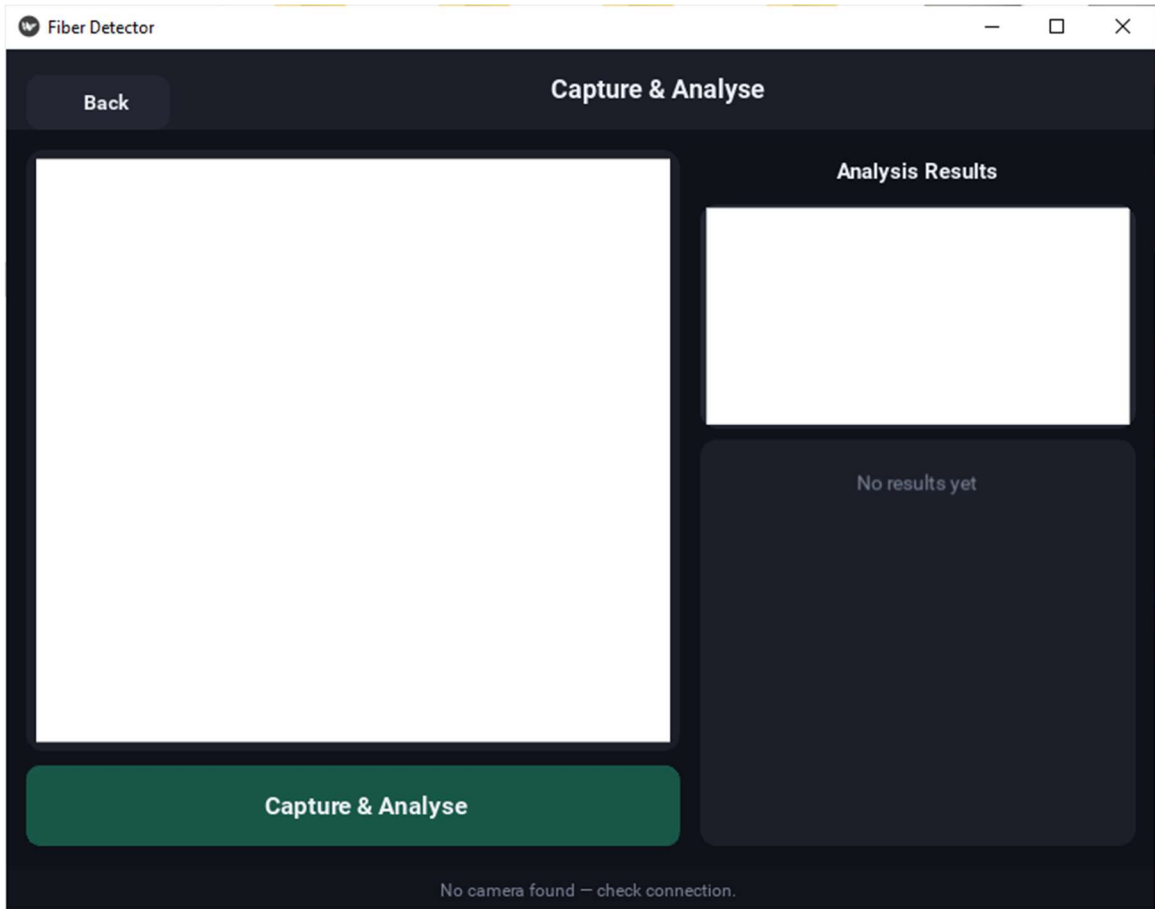
Figure 7.4.1 Touch Screen Hub Home Interface

The image above shows the concept for the home/main screen of the touchscreen. It is a simple design with the options of capturing an image or reviewing previous processing results.



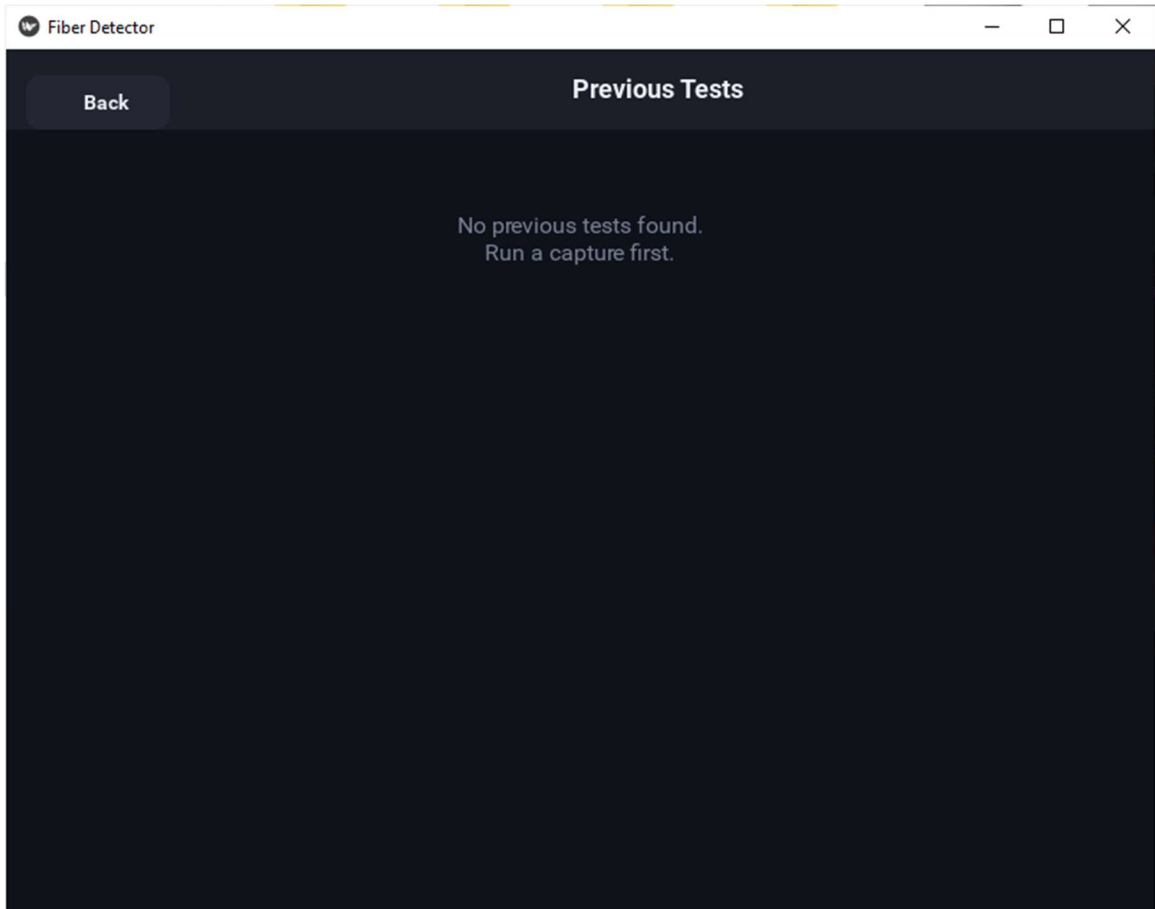
**Figure 7.4.2 Reference Selection Interface**

This image showcases the concept for the reference image section. It will simply consist of a list of all stored images, as well as access to a capture from camera option.



**Figure 7.4.3 Capture Screen Interface**

The Capture and Analyze screen will include a live view from the camera as well as a button to capture and analyze the image. A small display section to the right will have the results. Furthermore, camera connection status will be shown at the bottom.



**Figure 7.4.4 Test History Interface**

Finally, a test history screen will list the results of previous tests, while also providing access to said results.

## **Chapter 8 – System Fabrication**

In the following chapter, we will discuss the fabrication process and technologies necessary to build our prototype. Since we are purchasing all our electronic and optical components off-the-shelf, all our fabrication methods are mechanical. This will let us mount our optical components and electronic components appropriately to achieve a successful prototype.

### **8.1 Mechanical Fabrication**

The mechanical components of this project are highly customized and demand a high level of precision to ensure proper alignment, reliable assembly, and accurate system performance. To meet these requirements, additive manufacturing techniques, specifically 3D printing, will be used to produce parts with the required tolerances and geometries.

### **8.2 3D Printer**

The 3D printer selected for this project is the Bambu Lab X1 Series. This printer has been chosen due to its 7  $\mu\text{m}$  LiDAR resolution, which enables extremely high tolerances

and excellent dimensional accuracy during printing. In addition, the X1 Series features multi-sensor active calibration, allowing the machine to automatically tune bed leveling, flow rate, and motion parameters, resulting in consistent print quality across multiple runs. The printer also supports a wide range of engineering filaments, including PLA-CF, ABS, and other high-strength materials required for the optical housing and structural components of the system.

The Bambu Lab X1 Series is also a practical choice for this project because, as students at the University of Central Florida, we have free access to these machines through university resources such as the TI Innovation Lab. This eliminates equipment cost barriers and allows multiple design iterations to be printed and evaluated throughout development without financial burden.

### **8.3 3D Printing Setting and Parameters**

To ensure that the 3D printed components meet the dimensional accuracy, alignment requirements, and structural integrity needed for the P.I.D.I. system, several slicer settings will be adjusted during the printing process. These settings are chosen to optimize detail resolution, part strength, and surface finish, while minimizing errors such as smearing, warping, or dimensional deviation. The most important adjustments being made to this project are listed below:

**Reduced Print Speed:**

Slowing the print speed increases overall print time but improves precision and reduces the chance of extrusion errors or surface defects.

**Lower Layer Height:**

A smaller layer height allows for finer detail resolution and ensures small features in the mechanical and optical components are accurately formed.

**Reduced Line Width:**

Narrower extrusion widths enable printing of smaller details and tighter geometries without rounding or loss of dimensional accuracy.

**Slightly Lower Print Temperature:**

Dropping the temperature by approximately 5°C allows each layer to cool before the next is deposited, helping prevent smearing, drooping, and stringing.

**Triangular Infill Pattern:**

This pattern increases internal strength and rigidity of the printed parts while distributing loads more evenly. It also reduces the likelihood of stray light bleeding through the material, helping maintain optical isolation within the housing, although it increases print time slightly.

**Lower Overhang Threshold ( $\approx 40^\circ$ ):**

A more conservative overhang limit ensures that steep transitions do not sag or deform during printing.

**Tree Supports:**

Tree-style supports are used because they minimize material use, are easier to remove, and reduce the chance of damaging surfaces during post-processing.

## 8.4 Optical 3D Printed Optical Mounts

The optical mounts are designed as a two-piece system consisting of a front and back brace (can be seen in figures below) to securely hold each lens in place and maintain proper alignment within the optical path. The front and back sections will be fastened together using four CTSK Flat ISO 7046-1 screws, ensuring a rigid and reliable connection. To prevent the lenses from slipping or shifting during operation as well as assembly, each mount includes a 0.5 mm retaining lip at the inner edge, providing a positive stop that keeps the lens seated correctly

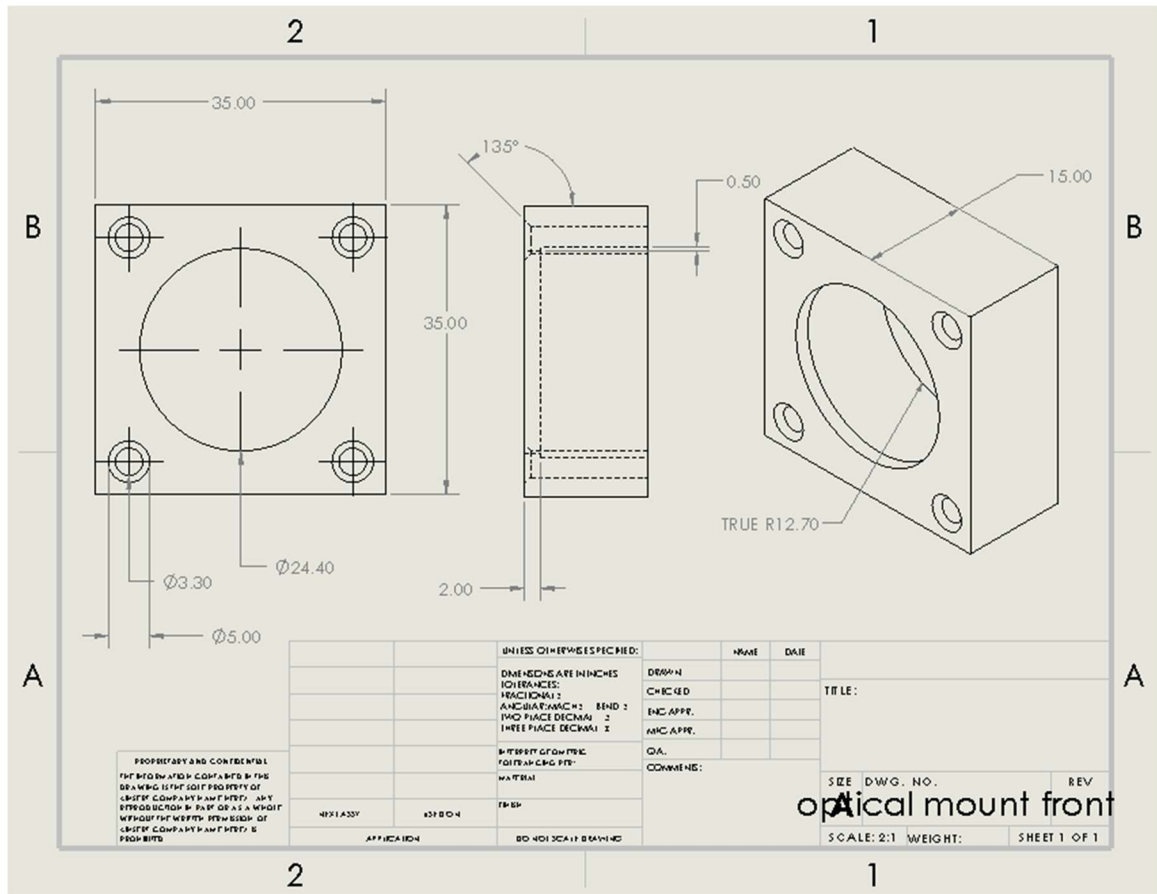
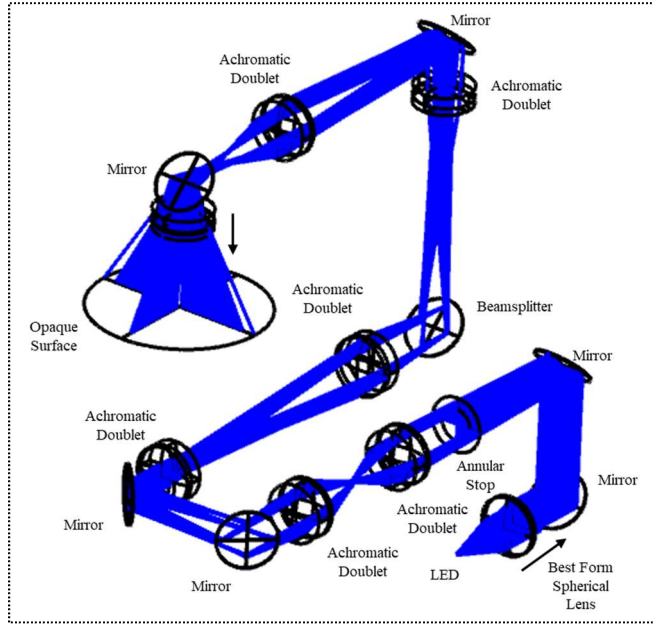


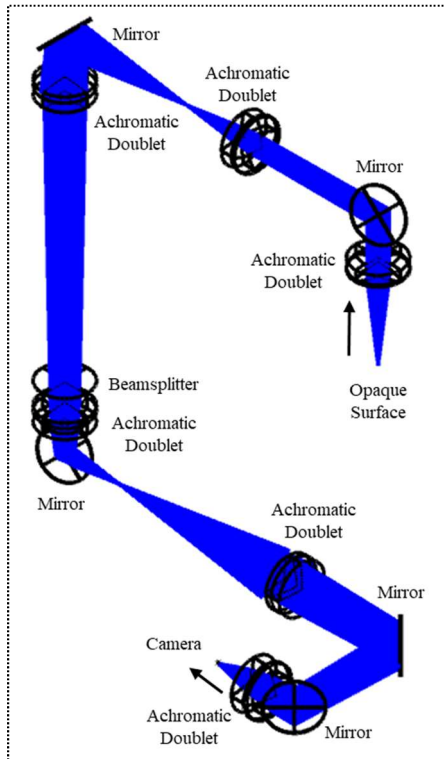
Figure 8.4.1 Optical Mount Front (mm)

This drawing demonstrates the general concept for the optical mount fronts. While the exact dimensions will vary from lens to lens, the overall structure and design approach will remain consistent. The 3D printing settings are especially important for these components because they include very small features and allow almost no margin for error. Precise manufacturing is required to ensure that each lens is held securely and maintains perfect optical alignment within the system. Even minor deviations in fit could introduce angular or axial misalignment, which would negatively affect optical performance and image quality. For this reason, these mounts will be printed with the highest accuracy settings and inspected carefully before integration into the assembly.





**Figure 9.1.1 Illumination Sequential Zemax Simulation**



**Figure 9.1.2 Detection Sequential Zemax Simulation**

We used the non-sequential simulations to integrate illumination and detection subsystems. We used this simulation for calculating scatter power and further ghosting analysis.

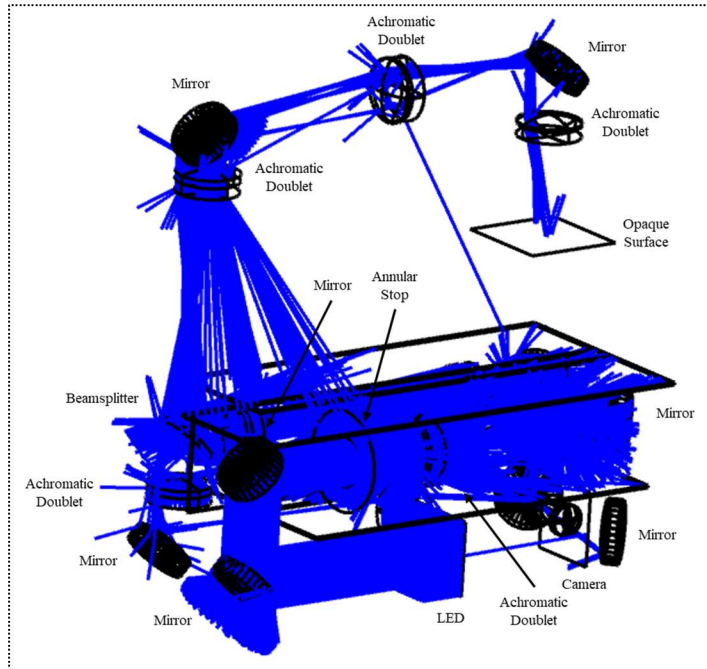


Figure 9.1.3 Non-Sequential Zemax Simulation

## 9.2 Optoelectronics Feasibility

The PIDI design utilizes two optoelectronic components, a CMOS camera and a high-power green LED. To test the feasibility of all our optical components, we tested the illumination subsystem and the detection subsystem on optical rails to ensure our components perform as expected.

### 9.2.1 Annular Stop Testing

We set up illumination lenses on optical rail, then tested 10 different annular stop inner radii and recorded spot diameter at opaque surface a smaller inner radius means a smaller detection half angle and a smaller detection angle means less edge scattering. Inner radius  $\geq 11$  mm causes annular stop to miss lens working diameter due to LED divergence, so we chose a 10 mm inner radius.

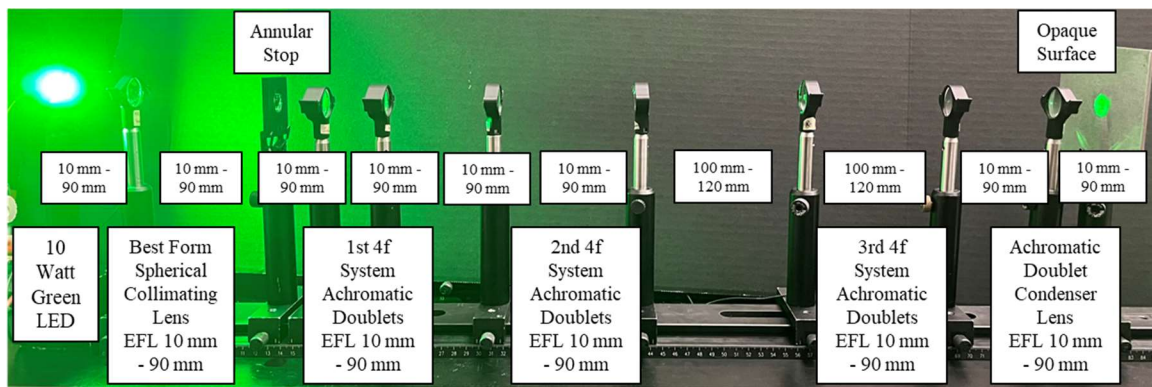
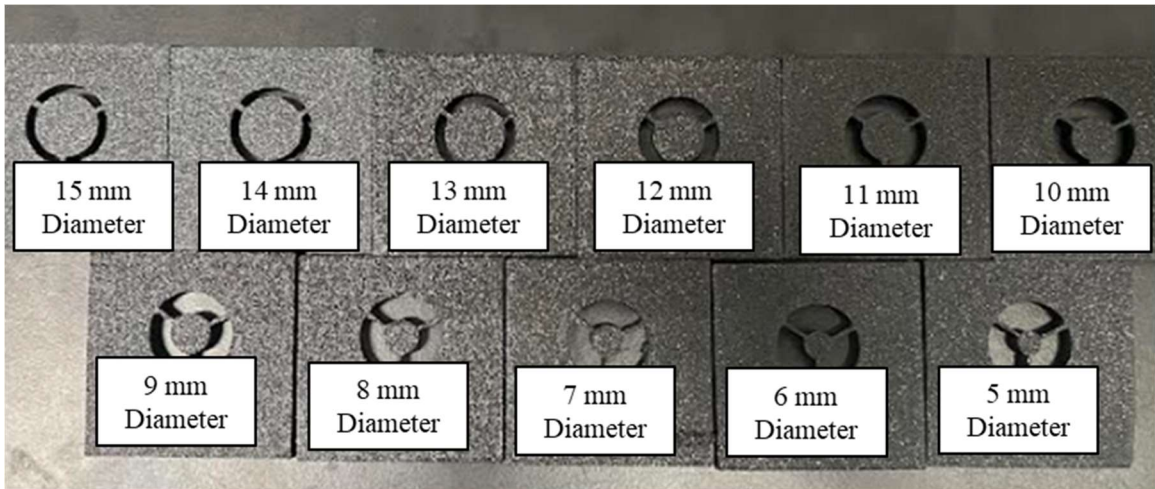
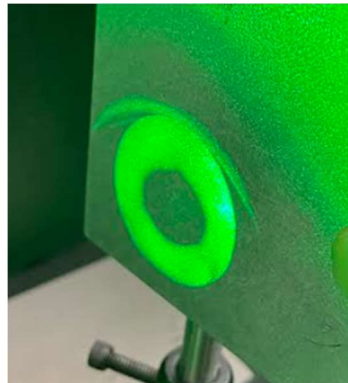


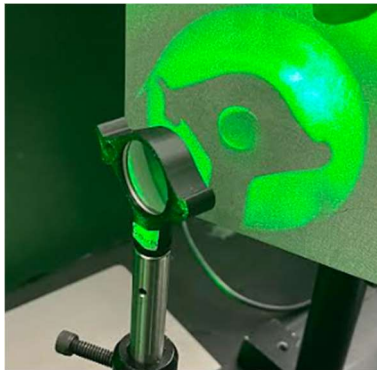
Figure 9.2.1 Annular Stop Test Setup



**Figure 9.2.2 Annular Stop Test Setup**



**Figure 9.2.3 Annular mask before 1st lens of 2nd 4f system of the illumination sub-system**



**Figure 9.2.4 Annular mask after 1st lens of 2nd 4f system of the illumination sub-system**

The graphs below represent the data collected and the relationship between annular stop inner radius and illumination angle.

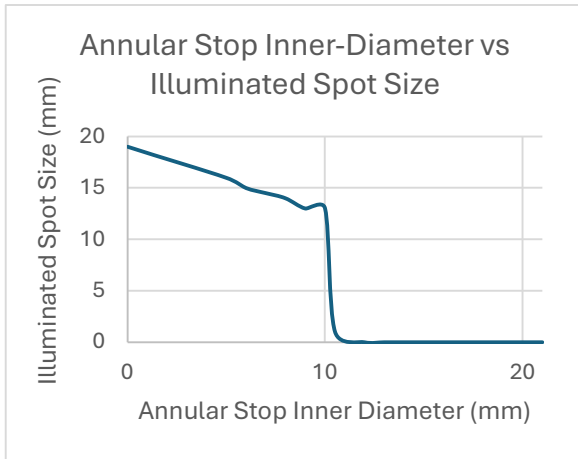


Figure 9.2.5 Annular Stop Inner-Diameter vs Illuminated Spot Size

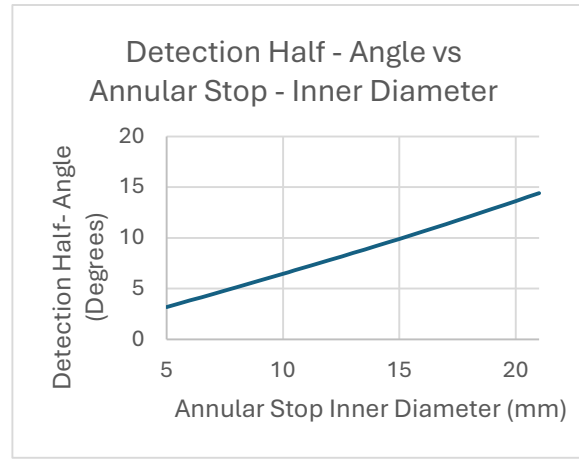


Figure 9.2.6 Detection Half - Angle vs Annular Stop - Inner Diameter

## 9.2.2 LED Power Efficiency Testing

We set up illumination optical components on optical rails then used an optical power meter to detect optical power after each 4f system. We measured power 3 times and measured average power efficiency to 2.7 % with the average power at opaque surface average being 136  $\mu\text{W}$ . Most power is lost from collimating lens and power efficiency of actual system is lower due to the 6 mirrors in the illumination sub system.

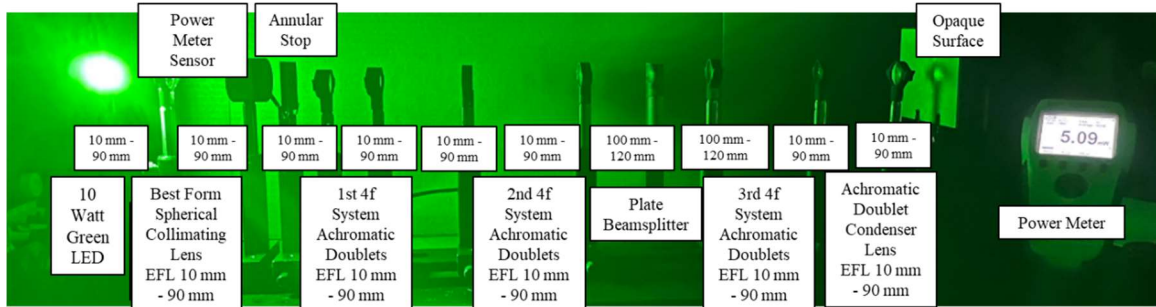


Figure 9.2.7 Power Efficiency Test Setup

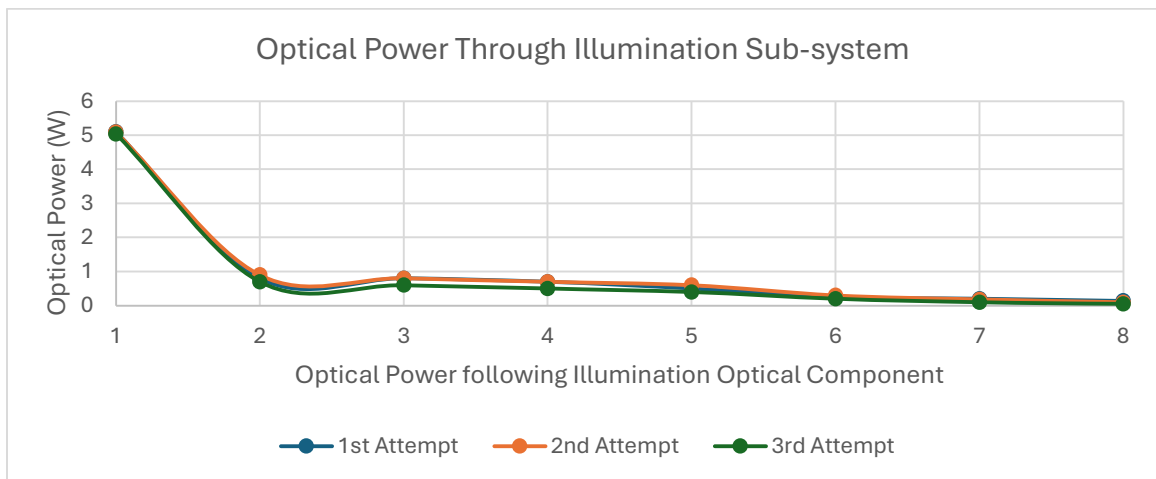


Figure 9.2.8 Optical Power Through Illumination Sub-system

The graph above displays the measured power after optical components, with 1 = LED, 2 = Collimating Lens, 3 = Annular Stop, 4 = 1<sup>st</sup> 4f System, 5 = 2<sup>nd</sup> 4f System, 6 = Beamsplitter, 7 = 3<sup>rd</sup> 4f System, and 8 = Condenser Lens / at Opaque Surface

### 9.2.3 Detection Subsystem Testing

We set up all out our detection path optical components on an optical rail to prove that our system can reach the proper magnification. In addition, we were able to double check that our lenses will actually able to properly fit in the housing we designed.

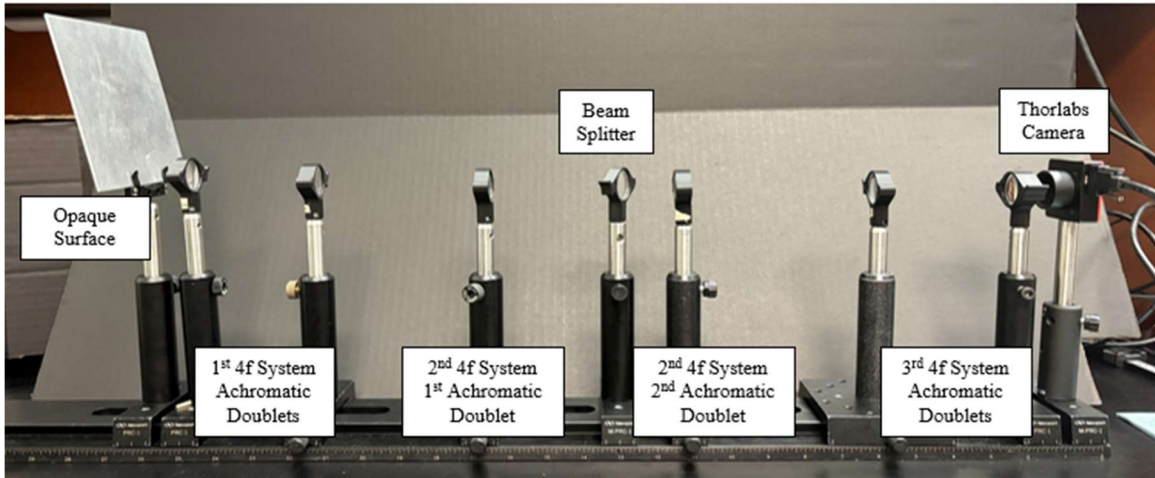


Figure 9.2.9 Detection Subsystem on an optical rail.

## Chapter 10 Administrative Content

### 10.1 Budget and Finances

ASML has provided our team with a \$5,000 budget to design a prototype particle inspector. This budget is quite high due to all the optical components necessary to build a darkfield illuminated microscopy system, but it is still quite low compared to darkfield microscopes that are used in the industry for inspection. We will save money by 3D printing the housing and some optical components (annular stop). The table below shows a breakdown of our budget by engineering field and components.

Table 10.1.1 Bill of Materials

Component	Part Name/Number	Quantity	Unit Price	Total Price
Optical Engineer				
Camera (1.6 MP CMOS)	CS165CU	1	\$513.45	\$513.45
Illumination Source (10 W Green LED)	1DGL-JC-10W-GR	1	\$6.99	\$6.99

Pellicle Beam Splitter	BP145B1	1	\$224.99	\$122.03
Dielectric Flat Mirrors	BB1-E02	10	\$86.91	\$869.10
Collimating Lens	LBF254-040-A	1	\$54.67	\$54.67
Achromatic Doublets	AC254-0XX-A	10	\$95.34 - \$102.96	\$976.26
Computer Engineer				
LattePanda Mu Intel N100 x86 Compute Module Kit	B0D4VC43HC	1	\$209.00	\$209.00
Belker 45W Universal AC DC Adapter Power Supply	PA-30450H-ZMX	1	\$16.90	\$16.90
Waveshare 7" Display	779172278936	1	\$46.59	\$46.59
Amazon Basics 8-Pack Rechargeable AA NiMH Batteries	HR-3UTG-AMZN (8P)	1	\$12.80	\$12.80
LD24AJTA Constant Current LED Driver	Fayelume91f7im5gc2	1	\$12.99	\$12.99
QTEATAK 2Set 8 x AA Thicken Battery Holder	SY-G03B-BGI7	1	\$8.99	\$8.99
Arduino Nano ESP32	ABX00083	1	\$19.30	\$19.30
Mechanical Engineer				
Steel Flat Head Thread forming screws 10mm	<u>90434A126</u>	100	.1886	\$14.86

Steel Flat Head Thread forming screws 16mm	<u>90434A151</u>	100	.1161	\$11.61
Black PLA –CF filament 1kg	<u>3502N211</u>	2	47.27	\$94.54
Black PLA 1 filament .5kg	<u>1317n22</u>	4	34.60	\$138.40
Matt duct tape	<u>7612A25</u>	1	16.76	\$16.76
<b>Total Cost</b>				
Optical Components		\$2,542.50		
Electrical Components		\$326.57		
Mechanical Components		\$276.17		
Grand Total		\$3,145.24		

As shown by the table above, we are comfortably below the \$5,000 budget.

## 10.2 Project Milestones

To develop a successful prototype, we have designed a team plan with project milestones for Senior Design I and Senior Design II.

**Table 10.2.1 Senior Design I Project Milestones**

Task	Time Period	Dates	Responsibility
Project Selection	2 Weeks	8/18 - 8/29	All
D&C Documentation	1 Week	8/29 - 9/5	All
Initial ASML Meeting	45 Minutes	9/5	All
D&C Meeting with Advisors	30 Minutes	9/10	All
Research Darkfield Illumination Design	2 Weeks	9/5 - 9/19	Kirolos
Research Magnification Design	2 Weeks	9/5 - 9/19	Julian
Computer Engineering Research	2 Weeks	9/5 - 9/19	Christian
CAD Particle Inspector	2 Weeks	9/5 - 9/19	Ryan
2 <sup>nd</sup> ASML Meeting (Get revision on design)	45 Minutes	9/19	All
Finalize Optical Design	2 Weeks	9/19 - 10/3	Kirolos, Julian

Computer Engineering Research	2 Weeks	9/19 - 10/3	Christian
Mechanical Engineering + Optical Mechanical Research	2 Weeks	9/19 - 10/3	Ryan
3 <sup>rd</sup> ASML Meeting (Finalize Optical Design)	45 Minutes	10/3	All
PSE Demo	30 Minutes	10/7	Kirolos, Julian
Finalize Components	2 Weeks	10/3 - 10/17	Kirolos, Julian
Computer Engineering Research	2 Weeks	10/3 - 10/17	Christian
Mechanical Engineering + Optical mechanical Research	2 Weeks	10/3 - 10/17	Ryan
4 <sup>th</sup> ASML Meeting (Pitch off-the-shelf components)	45 Minutes	10/17	All
Buy Components	2 Weeks	10/17 - 10/31	Kirolos, Julian
Computer Engineering Research	2 Weeks	10/17 - 10/31	Christian
Mechanical Engineering + Optical mechanical Research	2 Weeks	10/17 - 10/31	Ryan
5 <sup>th</sup> ASML Meeting	45 Minutes	10/31	All
Midterm Report Meeting / 60 Page Draft	30 Minutes	11/3 - 11/5	All
Test Components	2 Weeks	10/31 - 11/14	Kirolos, Julian
Computer Engineering Research	2 Weeks	10/31 - 11/14	Christian
Mechanical Engineering Research	2 Weeks	10/31 - 11/14	Ryan
6 <sup>th</sup> ASML Meeting	45 min	11/14	All
Start Prototype	2 Weeks	11/14 - 11/28	All
Final Report	10 Weeks	9/19 - 12/2	All

**Table 10.2.2 Senior Design II Project Milestones**

Task	Time Period	Dates	Responsibility
Receive Components from ASML	8 Weeks	1/12 - 3/5	Kirolos, Julian, and Christian
Test Optical Components	2 Weeks	3/5 - 3/19	Kirolos, Julian
Receive Correct Components from ASML	2 Weeks	3/5 - 3/19	
Test Electronic Components	4 Weeks	3/5 - 4/2	Kirolos, Julian, and Christian
Test 3D Printed Housing	4 Weeks	3/19 - 4/16	All
Test Software	2 Weeks	4/8 - 4/22	Christian
Assemble Complete Prototype	2 Weeks	4/8 - 4/22	All
Test and Fix Prototype	2 Weeks	4/8 - 4/22	All
Finalize Prototype	1 Week	4/15 - 4/22	All
Film Project Video	1 Weeks	4/11 - 4/18	All
Finalize and Update Project Report	2 Weeks	4/11 - 4/25	All
Live Demo	5 Hours	4/22	All

### 10.3 Table of Work Distribution

Table 10.3.1 details our current plan for the distribution of work. We organized this project into a series of tasks essential to achieving our goals. Each task has a primary person responsible assigned to completing it. Some tasks have a secondary person who is responsible for assisting the primary person when needed.

**Table 10.3.1 Work Distribution**

Task	Primary	Secondary
Darkfield Illumination	Kirolos	Julian
Illumination Lenses Design	Kirolos	-
Image Detection	Julian	Kirolos
Detection Lenses Design	Julian	-
Magnification	Julian	Kirolos

Housing Design	Ryan	-
CAD of System	Ryan	-
Manufacturing	Ryan	-
App GUI	Christian	-
Detection Algorithm	Christian	-
LED Controls	Christian	-
Camera Controls	Christian	-

## Chapter 11 – Conclusion

### 11.1 Optical Challenges

Dimensional constraints with off-the shelf components requirements from sponsor made a complete dark-field image not feasible. Annular stop size given our constraints must get demagnified from the free space requirement. To achieve dark-field image, annular stop needs to be 21 mm in diameter before the condenser lens. If we try to magnify then demagnify the annular stop, it misses lenses' working diameter since we are limited to 1 inch optics from dimensional constraints. Our project still enhances edge scatter and reduces brightfield reflections, but does not result in a complete dark-field image.

Collimating lens diverges illumination beam more than expected. Limited on time after receiving components so we could not buy an alternative collimating lens. Causes the mask of the annular stop to miss working diameter of the 1st lens in the 2nd 4f system in the illumination subsystem. Tested different inner radius dimensions of the annular stop to see at what size the annular stop does not miss lenses' working diameter. This causes less edge scattering lowering the dark-field effect on the image

LED chosen is composed of nine LEDs in a grid. This caused inconsistency of intensity in the illumination beam on the opaque surface. This made us add a diffuser in front of the LED to make the nine-point sources homogeneous

Finding Zemax recourses since our sponsor requested non-sequential simulations (no prior experience in non-sequential mode). Imaging professor resigned so had to coordinate with the CREOL Dean for other faculty members with prior non-sequential Zemax experience. Got connected with Dr. Grimming, who helped guide us through the non-sequential Zemax simulation.

### 11.2 Mechanical Challenges

The dimensional constraints provided by ASML were where we faced the largest mechanical challenges. Due to these constraints, sacrifices had to be made in quality and performance. The beam splitter mount did not have enough available space to be made completely stable, and the camera adjuster did not have the required room within the housing to be as rigid or provide as much range of movement as we would have liked.

The LED that we selected also created several challenges. Because it was a 10 Watt LED, it naturally generated a significant amount of excess heat. Due to the limited available space inside the housing, we did not have enough room to properly ventilate or dissipate the heat from the system. As a result, we were unable to keep the LED on for extended periods of time, since prolonged use would eventually cause the mount to soften and misalign or melt.

The delay in receiving parts also created significant challenges. Due to the amount of time required to finalize the optical design, combined with delays in receiving components, we were left with a very limited amount of time to fabricate and redesign the mechanical components. As a result, the overall quality of the housing suffered because there was not enough time for additional refinement, testing, and improvements.

### **11.3 Future Improvements**

There are a lot of possible improvements to the project that can help engineers design future iterations of this tool.

The use of custom optics can provide a dark-field image in the given dimensional constraints. Enlarging the housing and reducing the free space dimensions can allow for better integration of illumination and detection subsystems. Choosing a collimating lens with lower beam divergence will allow the use of larger annular stop.

With the use of custom optics the available space within the housing would increase allowing us to add improvements such as adding a heat sync to led allowing to dissipate heat better, this would also create space for us to add a motorized camera focusing mechanism allowing for better quality image and more repeatability within the system.

A major improvement that could be implemented with an increased budget and additional time would be to fabricate the mounts, units, and housing through subtractive manufacturing methods such as machining and CNC milling. Doing so would greatly increase the strength and rigidity of the housing while also improving the overall quality and durability of the design. These manufacturing methods typically have a much smaller margin of error than additive methods, allowing for higher precision parts, tighter tolerances, and better alignment of the optical and mechanical components. In addition, machined components would provide a more professional final product, reduce unwanted movement or flexing, and improve the long term reliability of the system during repeated use.

## Chapter 12 Appendices

### 12.1 Appendix A – References

1.6 MP CMOS Compact Scientific Cameras.” *Thorlabs.com*, 2020, [www.thorlabs.com/newgrouppage9.cfm?objectgroup\\_id=13677#ad-image-0](http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=13677#ad-image-0).

“10 Watt Green LED.” *Vetco Electronics*, 2024, [vetco.net/products/vupn1856\\_10\\_watt\\_green\\_led?variant=42879338840201&country=US&cy=USD&utm\\_medium=product\\_sync&utm\\_source=google&utm\\_content=sag\\_organic&utm\\_campaign=sag\\_organic&srsltid=AfmBOoqS9HFmyo6fmANBjA3s8W5cYzpjZK0QNIVJcZ2BM2IT6edjHQiYyE](http://vetco.net/products/vupn1856_10_watt_green_led?variant=42879338840201&country=US&cy=USD&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic&srsltid=AfmBOoqS9HFmyo6fmANBjA3s8W5cYzpjZK0QNIVJcZ2BM2IT6edjHQiYyE). Accessed 31 Oct. 2025.

Abramowitz, Mortimer. “Reflected Darkfield Objectives.” *Evident Scientific*, [evidentscientific.com/en/microscope-resource/tutorials/darkfield/reflected](http://evidentscientific.com/en/microscope-resource/tutorials/darkfield/reflected).

Advanced illumination. “Bright Field vs. Dark Field Lighting Techniques.” *Advanced Illumination*, 5 Sept. 2019, [advancedillumination.com/lighting-education/bright-field-dark-field-lighting/](http://advancedillumination.com/lighting-education/bright-field-dark-field-lighting/).

AUTODESK. “AutoCAD for Mac & Windows | CAD Software | Autodesk.” *Autodesk.com*, 2018, [www.autodesk.com/products/autocad/overview](http://www.autodesk.com/products/autocad/overview).

Autodesk. “Fusion 360 for Personal Use | Fusion 360 | Autodesk.” *Www.autodesk.com*, [www.autodesk.com/products/fusion-360/personal](http://www.autodesk.com/products/fusion-360/personal).

“CCD vs CMOS: A Review of Sensor Technology.” *CMOS Sensor Inc*, [www.csensor.com/ccd-vs-cmos](http://www.csensor.com/ccd-vs-cmos).

“CCD vs. CMOS.” *Baslerweb.com*, 2015, [www.baslerweb.com/en-us/learning/ccd-cmos/](http://www.baslerweb.com/en-us/learning/ccd-cmos/).

“Choosing the Best 3D CAD Software: A Comprehensive Guide.” *Formlabs*, 2016, [formlabs.com/blog/cad-software/?srsltid=AfmBOorj7cXioGURs0kiRPkxk6lYyh9BL-CNP-MGybIG56taHsOMNrxF](http://formlabs.com/blog/cad-software/?srsltid=AfmBOorj7cXioGURs0kiRPkxk6lYyh9BL-CNP-MGybIG56taHsOMNrxF). Accessed 31 Oct. 2025.

Claire Abadie, Mingyang Liu, Yoann Prado, and Olivier Pluchery, "Hyperspectral dark-field optical microscopy correlated to atomic force microscopy for the analysis of single plasmonic nanoparticles: tutorial," *J. Opt. Soc. Am. B* 41, 1678-1691 (2024)

“CMOS vs. CCD Technologies.” *Wileyindustrynews.com*, 4 Mar. 2022, [wileyindustrynews.com/en/contributions/cmos-vs-ccd-technologies](http://wileyindustrynews.com/en/contributions/cmos-vs-ccd-technologies).

“Darkfield Illumination.” *Nikon’s MicroscopyU*, [www.microscopyu.com/techniques/stereomicroscopy/darkfield-illumination](http://www.microscopyu.com/techniques/stereomicroscopy/darkfield-illumination).

Davidson, Michael. “ZEISS Microscopy Online Campus | Tungsten-Halogen Lamps.” *Zeiss-Campus.magnet.fsu.edu*, [zeiss-campus.magnet.fsu.edu/articles/lightsources/tungstenhalogen.html](http://zeiss-campus.magnet.fsu.edu/articles/lightsources/tungstenhalogen.html).

Dipert, Brian. “CMOS vs CCD: Why CMOS Sensors Are Ruling the World of Embedded Vision.” *Edge AI and Vision Alliance*, 24 Apr. 2023, [www.edge-ai-vision.com/2023/04/cmos-vs-ccd-why-cmos-sensors-are-ruling-the-world-of-embedded-vision/](http://www.edge-ai-vision.com/2023/04/cmos-vs-ccd-why-cmos-sensors-are-ruling-the-world-of-embedded-vision/).

“DR USB3: 5 MP, Color, CS-Mount, Back | Teledyne Vision Solutions.” *Teledynevisionsolutions.com*, 2025, [www.teledynevisionsolutions.com/products/dragonfly-s-usb3/?model=DR-U3-50Y2C-C1-CS&vertical=machine+vision&segment=iis&dlPage=1](http://www.teledynevisionsolutions.com/products/dragonfly-s-usb3/?model=DR-U3-50Y2C-C1-CS&vertical=machine+vision&segment=iis&dlPage=1).

Dr. Joachim Linkemann. “CCD vs. CMOS.” *Qualitymag.com*, Quality Magazine, Mar. 2011, [www.qualitymag.com/articles/89621-ccd-vs-cmos?](http://www.qualitymag.com/articles/89621-ccd-vs-cmos?) Accessed 31 Oct. 2025.

Dr. Rüdiger Paschotta. “Achromatic Optics.” *Rp-Photonics.com*, 7 Dec. 2019, [www.rp-photonics.com/achromatic\\_optics.html](http://www.rp-photonics.com/achromatic_optics.html).

“Eneloop pro - Panasonic.” *Www.panasonic.com*, [www.panasonic.com/global/energy/products/eneloop/en/lineup/eneloop-pro.html](http://www.panasonic.com/global/energy/products/eneloop/en/lineup/eneloop-pro.html).

“Eneloop Rechargeable Batteries. Tests and More in 2024.” *Https://Eneloop101.com/*, 14 Feb. 2017, [eneloop101.com/batteries/eneloop-test-results/](http://eneloop101.com/batteries/eneloop-test-results/).

Grand, Alec De. “What Is Darkfield Microscopy?” *Olympus LS*, 17 Dec. 2020, [evidentscientific.com/en/insights/what-is-darkfield-microscopy](http://evidentscientific.com/en/insights/what-is-darkfield-microscopy).

“Green 10W High Power LEDDefault Title.” *Switch Electronics | Your One Stop Component Shop*, 2025, [www.switchelectronics.co.uk/products/green-10w-high-power-led?currency=GBP&variant=45334982787381&utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=Google%20Shopping&stkn=bbf1d20e1ed7&srsltid=AfmBOoKIE1T\\_skZmMe\\_Ds24pEiCk9P6E8ERsn3r\\_5HHdMWb\\_8o4MHXaW4o](http://www.switchelectronics.co.uk/products/green-10w-high-power-led?currency=GBP&variant=45334982787381&utm_source=google&utm_medium=cpc&utm_campaign=Google%20Shopping&stkn=bbf1d20e1ed7&srsltid=AfmBOoKIE1T_skZmMe_Ds24pEiCk9P6E8ERsn3r_5HHdMWb_8o4MHXaW4o). Accessed 31 Oct. 2025.

Gürsoy, D., Yay, K.A., Kisiel, E. et al. Dark-field X-ray microscopy with structured illumination for three-dimensional imaging. *Commun Phys* 8, 34 (2025). <https://doi.org/10.1038/s42005-025-01952-2>

“How Do Halogen Lamps Work? & Bulbs from Commercial Lighting Experts.” *Www.superiorlighting.com*, [www.superiorlighting.com/lighting-resources/light-bulb-learning-center/halogen-light-bulbs/how-do-halogen-lamps-work/](http://www.superiorlighting.com/lighting-resources/light-bulb-learning-center/halogen-light-bulbs/how-do-halogen-lamps-work/).

“I2C Bus Specification.” *I2C Info – I2C Bus, Interface and Protocol*, [i2c.info/i2c-bus-specification](http://i2c.info/i2c-bus-specification).

“Laser Diode vs LED: Know the Difference.” *Www.gentec-Eo.com*, 30 May 2023, [www.gentec-eo.com/blog/laser-diode-vs-led-know-the-difference](http://www.gentec-eo.com/blog/laser-diode-vs-led-know-the-difference).

Lumistrips LED Professional. “Understading Luminous Efficacy (Lumens per Watt) and Its Impact on LED Performance.” *Lumistrips*, 12 Apr. 2023, [us.lumistrips.com/lumistrips-blog/led\\_efficacy\\_efficiency\\_explained/?utm\\_source=chatgpt.com](http://us.lumistrips.com/lumistrips-blog/led_efficacy_efficiency_explained/?utm_source=chatgpt.com). Accessed 31 Oct. 2025.

“MC-70 Trinocular Reflected Light Brightfield/Darkfield Tool Makers/Measuring Microscope.” *Meiji Techno America*, [meijitechno.com/product/mc-70-trinocular-reflected-light-brightfield-darkfield-measuring-microscope/](http://meijitechno.com/product/mc-70-trinocular-reflected-light-brightfield-darkfield-measuring-microscope/). Accessed 5 Sept. 2025.

“Native-Cloud Optical Prototyping & Optical Bench Simulation Software.” *3DOptix*, 15 May 2025, [www.3doptix.com/](http://www.3doptix.com/). Accessed 31 Oct. 2025.

Norton, Robert L. *Design of Machinery*. McGraw-Hill Education, pp .218-230, 2019.

“Power Designer.” *Ti.com*, 2025, [webench.ti.com/power-designer/switching-regulator/customize/6](http://webench.ti.com/power-designer/switching-regulator/customize/6). Accessed 31 Oct. 2025.

“Power Designer.” *Ti.com*, 2025, [webench.ti.com/power-designer/switching-regulator/customize/6](http://webench.ti.com/power-designer/switching-regulator/customize/6). Accessed 31 Oct. 2025.

“Prusa Slicer 2.8.1 - Download.” *Prusa Slicer*, 28 Dec. 2024, [prusaslicer.net/](http://prusaslicer.net/).

“Raspberry Pi CM5 Is 2-3x Faster, Drop-in Upgrade (Mostly) | Jeff Geerling.” *Jeffgeerling.com*, 27 Nov. 2024, [www.jeffgeerling.com/blog/2024/raspberry-pi-cm5-2-3x-faster-drop-upgrade-mostly?utm\\_source=chatgpt.com](http://www.jeffgeerling.com/blog/2024/raspberry-pi-cm5-2-3x-faster-drop-upgrade-mostly?utm_source=chatgpt.com). Accessed 31 Oct. 2025.

“Reflected Darkfield Illumination.” *Evident Scientific*, [evidentscientific.com/en/microscope-resource/knowledge-hub/techniques/darkfieldreflect](http://evidentscientific.com/en/microscope-resource/knowledge-hub/techniques/darkfieldreflect). Accessed 5 Sept. 2025.

Schwartz, Katie, and Jim Burge. “Field Guide to Optomechanical Design and Analysis.” SPIE Press, 2012. <https://doi.org/10.1117/3.934930>. . PDF ISBN: 9780819491626 | Print ISBN: 9780819491619.

*Singlet Lens*. (n.d.). Photonics Marketplace. Retrieved October 30, 2025, from [https://www.photonics.com/EDU/singlet\\_lens/d8646](https://www.photonics.com/EDU/singlet_lens/d8646)

“Slic3r - Open Source 3D Printing Toolbox.” *Slic3r.org*, 2019, [slic3r.org/](http://slic3r.org/).

SolidWorks. “3D CAD Design Software.” *Solidworks.com*, 2018, [www.solidworks.com/](http://www.solidworks.com/).

“Spinnaker SDK | Teledyne Vision Solutions.” *Teledynevisionsolutions.com*, 2015, [www.teledynevisionsolutions.com/products/spinnaker-sdk/?model=Spinnaker%20SDK&vertical=machine%20vision&segment=iis](http://www.teledynevisionsolutions.com/products/spinnaker-sdk/?model=Spinnaker%20SDK&vertical=machine%20vision&segment=iis).

Svetlana N. Khonina, Nikolay L. Kazanskiy, Ivan V. Oseledets et al., "Synergy between Artificial Intelligence and Hyperspectral Imaging—A Review", *Technologies* 12(9), pg. 163, (2024); doi:10.3390/technologies12090163

Tu, Shangqing, et al. *ChatLog: Recording and Analyzing ChatGPT across Time*. Apr. 2023, <https://doi.org/10.48550/arxiv.2304.14106>.

Ultimaker. “UltiMaker Cura.” *UltiMaker*, [ultimaker.com/software/ultimaker-cura/](http://ultimaker.com/software/ultimaker-cura/).

UM10204 I 2 C-Bus Specification and User Manual Rev. 7.0 -1 October 2021 User Manual Document Information Information Content. [www.nxp.com/docs/en/user-guide/UM10204.pdf?utm\\_source=chatgpt.com](http://www.nxp.com/docs/en/user-guide/UM10204.pdf?utm_source=chatgpt.com).

“Unmounted Achromatic Doublets, AR Coated: 400 - 700 Nm.” *Thorlabs.com*, 2025, [www.thorlabs.com/newgrouppage9.cfm?objectgroup\\_id=120](http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=120).

“Why Language Models Hallucinate.” *Openai.com*, 27 Aug. 2025, [openai.com/index/why-language-models-hallucinate/](http://openai.com/index/why-language-models-hallucinate/).

*Edmundoptics.com*, 2015, [www.edmundoptics.com/p/254mm-clear-aperture-50r50t-pellicle-beamsplitter/4506/](http://www.edmundoptics.com/p/254mm-clear-aperture-50r50t-pellicle-beamsplitter/4506/). Accessed 31 Oct. 2025.

## 12.2 Appendix B LLMs

### Appendix B.1 Case Study 1

Question: can you explain to me how i would do raytracing for an optical system.

Response: Nice — let’s walk through a clear, practical guide to doing ray-tracing for an optical system. I’ll cover the math you need, the algorithmic flow (sequential and non-sequential), common surface types and intersection methods, refraction/reflection formulas, how to measure image quality, and implementation tips + simple pseudocode and a short Python snippet you can drop into a simulator.

#### 1) Two big approaches

**Geometrical (ray) tracing** — follow rays as straight lines between surfaces, apply Snell’s law and reflection at interfaces. Good for lenses, mirrors, stray-light, spot diagrams, pupil mapping.

**Physical (wave) optics** — propagate fields (Fresnel/Fourier optics) when diffraction/wave effects matter (small apertures, coherence, MTF at high spatial frequency).

Start with geometrical raytracing; move to physical optics when diffraction/phase is important.

#### 2) Data model (what each ray and surface stores)

- **Ray:** origin  $\mathbf{o}$  (3D vector), direction  $\mathbf{d}$  (unit 3D vector), wavelength ( $\lambda$ ), power/intensity, polarization state (optional, Jones vector).
- **Surface:** geometric type (plane, sphere, asphere, freeform), parameters (radius  $R$ , center  $C$ , conic and aspheric coeffs), aperture (stop/clear aperture), coating (reflectance/phase), refractive index on either side ( $n_{in}$ ,  $n_{out}$ ), tilt/decenter transforms.
- **System:** ordered list of surfaces (for sequential), global coordinate frames, aperture stop location, field points, wavelength(s).

#### 3) Ray–surface intersection (core geometry)

General idea: parametric ray  $\mathbf{r}(t) = \mathbf{o} + t \mathbf{d}$ . Solve for  $t$  where  $\mathbf{r}(t)$  satisfies surface equation. Keep  $t > 0$  and choose smallest positive  $t$  for sequential systems.

**Sphere (center  $C$ , radius  $R$ ):**  
Solve  $|\mathbf{o} + t \mathbf{d} - C|^2 = R^2 \rightarrow$  quadratic  $a t^2 + b t + c = 0$  with

- $a = \mathbf{d} \cdot \mathbf{d} = 1$  (if  $\mathbf{d}$  unit),

- $b = 2 \cdot d \cdot (o - C)$ ,
- $c = |o - C|^2 - R^2$ .  
Pick smallest positive root. If discriminant  $< 0 \rightarrow$  no intersection.

**Plane (point P0, normal n):**  
Solve  $(o + t d - P0) \cdot n = 0 \rightarrow t = (P0 - o) \cdot n / (d \cdot n)$ ; if  $d \cdot n \approx 0$  then ray parallel.

**Asphere / freeform:**  
No closed form — use a root finder (Newton–Raphson) on the surface sag function  $F(x,y,z)=0$  using an initial guess (e.g., intersecting the base sphere or the projected ray-plane). Evaluate surface normal from gradients.

Numerical stability: clip tiny denominators, reject near-tangential hits, use double precision.

#### 4) Surface normal

After intersection point P, compute normal n:

- sphere:  $n = (P - C) / |P - C|$
- plane: constant n
- general surface:  $n = \nabla F / |\nabla F|$  where  $F(x,y,z)=0$  defines surface.

Ensure n points from medium 1 into medium 2 (convention matters for sign in Snell).

#### 5) Reflection & refraction (vector forms)

**Reflection** (incident direction v pointing *into* surface, n unit normal pointing out of medium):

$$r = v - 2 (v \cdot n) n$$

**Refraction** (Snell's law vector form). Let  $\eta = n1 / n2$  ( $n1 =$  refractive index of incident medium),  $\cos i = - n \cdot v$  (assuming v points toward the surface and n points into medium 2). Then:

$$k = 1 - \eta^2 * (1 - \cos i^2)$$

if  $k < 0$ : total internal reflection

else:

$$t = \eta * v + (\eta * \cos i - \text{sqrt}(k)) * n$$

t should be normalized if needed. Test sign conventions carefully.

**Fresnel amplitude/power:** use Fresnel equations for s/p polarization or unpolarized average to compute reflected/transmitted power and (optionally) phase shifts.

#### 6) Optical path length, wavefront & phase

For coherent/wave analysis compute Optical Path Length (OPL) along ray:  $OPL = \sum n_i * \text{segment\_length}_i$ . Phase =  $2\pi * OPL / \lambda$ . Wavefront error = difference between actual OPL and ideal reference OPL. Use OPD to compute MTF/PSF by Fourier optics.

## 7) Sequential vs Non-sequential

- **Sequential:** rays hit surfaces in a fixed order (good for lens stacks). Compute intersections one surface at a time.
- **Non-sequential:** rays can hit any surface in any order (stray light, illumination systems, scattering). Use acceleration structures (BVH, KD-tree) for many surfaces.

## 8) Sampling strategies & ray budget

- Use an array of field points (object angles) and sample many rays per field (e.g., grid or random across entrance pupil).
- For imaging PSF: trace many rays from each field through pupil to image plane; map intersection points to produce spot diagrams; build irradiance map (2D histogram) for PSF.
- Use importance sampling near aperture edges and for specular reflections if non-sequential stray light matters.

## 9) Image quality metrics

- **Spot diagram:** scatter of ray hit points at image plane.
- RMS spot radius, Encircled energy (EE50, EE80).
- **Wavefront error** (RMS, peak-to-valley), fit to Zernike polynomials.
- **MTF:** from PSF via Fourier transform; for real systems you may include detector sampling and pixel integration.
- **Strehl ratio:**  $\approx \exp[-(2\pi \sigma_w/\lambda)^2]$  for small wavefront error  $\sigma_w$ .

## 10) Handling tilted/decentered surfaces

Represent each surface in its own local coordinate frame with an affine transform from global frame: rotation + translation. Transform ray origin and direction into the surface frame for intersection; transform back after applying refraction/reflection.

## 11) Tolerances and numerical gotchas

- Use robust root finding for aspheres; clamp values to avoid sqrt of negative due to rounding.
- Always check  $d \cdot n$  not too small before division.
- For grazing incidence, handle boundaries carefully.
- Keep units consistent (meters or mm); wavelength in same units.

## 12) Algorithm — step-by-step pseudocode (sequential system)

for each field\_point:

  for each ray\_sample across entrance pupil:

    ray = create\_ray(field\_point, ray\_sample)

    for surface in surfaces\_in\_order:

      transform ray into surface frame

      t = intersect(ray, surface)

```

if no intersection or t <= 0: mark ray lost; break
P = ray.origin + t * ray.dir
if point outside surface aperture: mark ray blocked; break
n = compute_normal(P, surface)
if surface.is_reflective:
    ray.dir = reflect(ray.dir, n)
else:
    (ray.dir, transmitted_power) = refract(ray.dir, n, n1, n2)
    if TIR occurred: handle reflect or mark lost depending on coating
ray.origin = P + epsilon * ray.dir # avoid self-intersection
if ray hit image_plane:
    record hit position, power, OPL

```

13) Quick concrete math: sphere intersection + refraction (Python-style snippet)

```
import numpy as np
```

```

def intersect_sphere(o, d, C, R):
    # o,d: ray origin and unit direction (3,)
    oc = o - C
    b = 2.0 * np.dot(d, oc)
    c = np.dot(oc, oc) - R*R
    disc = b*b - 4*c
    if disc < 0: return None
    sqrtD = np.sqrt(disc)
    t0 = (-b - sqrtD) / 2.0
    t1 = (-b + sqrtD) / 2.0
    t = min(t for t in (t0,t1) if t > 1e-9) # choose smallest positive root
    return t

```

```

def refract(v, n, n1, n2):
    # v: incident unit direction (pointing INTO surface)

```

```

eta = n1 / n2
cosi = -np.dot(n, v)
k = 1 - eta*eta * (1 - cosi*cosi)
if k < 0:
    return None # total internal reflection
t = eta * v + (eta*cosi - np.sqrt(k)) * n
return t / np.linalg.norm(t)

```

#### 14) Advanced topics (if you need them)

- **Polarization:** track Jones or Mueller matrices; apply Fresnel coefficients separately for s/p.
- **Coatings:** multi-layer thin films change amplitude & phase; compute via transfer-matrix method.
- **Scattering / BRDF:** for rough surfaces, use microfacet models or measured BSDF; non-sequential ray trace with Monte Carlo.
- **Optimization:** define merit function (spot size, MTF, back focal length etc.) and use gradient-based or global optimizers to change radii, thicknesses, aspheric coefficients.
- **GPU acceleration:** ray tracing parallelizes well; use CUDA/OptiX or OpenCL for large ray counts.

#### 15) Validation & debugging checklist

- Test on simple systems with analytical solutions: single spherical lens, plane mirror, thin lens formula.
- Conserve energy: ensure sum of transmitted + reflected powers  $\approx$  incident (within coating losses).
- Visualize rays and spot diagrams frequently — errors are easier to spot visually.
- Compare to known optical design tools (Zemax, Code V, OSLO) on test cases.

#### 16) Libraries & tools (no deep web lookup here)

If you want to prototype: Python + NumPy is fine for learning. For larger projects consider Opticspy, POPPY (for some diffraction), or commercial tools — I can list current libraries if you want.

If you'd like, I can:

- produce a runnable Python ray-tracer for a simple singlet lens and show plots of the spot diagram and PSF, or
- explain how to add polarization & Fresnel coating calculations, or
- walk you through implementing non-sequential stray light with BVH acceleration.

Which of those would you like next?

Question:

if i did this with calculated this with a set of matrices would i need to change the matrices at all for calculating different rays or do i just change the input vector

ChatGPT said:

Thought for 4 seconds

Short answer: If you stay in the paraxial, linear regime and keep the same optical element positions, materials, and reference planes, the element and system matrices do not change — you just change the input ray vector.

Below is a compact explanation, the important caveats (when you *must* change matrices), and a tiny example so you can test it right away.

Why the matrices stay fixed

Paraxial ray matrices (the  $2 \times 2$   $[[1, d/n],[0,1]]$  propagation and  $[[1,0],[-\Phi,1]]$  surface/refraction matrices) are **linear operators** that map any input ray vector

$$\mathbf{r}_{\text{in}} = \begin{bmatrix} y \\ n\theta \end{bmatrix}$$

at one reference plane to an output ray vector at another plane by

$$\mathbf{r}_{\text{out}} = M_{\text{sys}} \mathbf{r}_{\text{in}}.$$

$M_{\text{sys}}$  is determined only by the sequence of elements, their separations, radii, and refractive indices (and by the two planes you chose as the input/output references). Once those are fixed, **one matrix applies to every paraxial ray** — you just change  $y$  and  $n\theta$  in the input vector.

What you change per ray

- ray height  $y$
- ray angle  $\theta$  (or equivalently  $n\theta$ )
- wavelength (if you want to model dispersion — see caveat)  
So for different field points or pupil samples you feed different input vectors into the same  $M_{\text{sys}}$ .

When you must change the matrices

1. **Different material refractive index (wavelength / dispersion):**  $n$  appears in propagation and refraction matrices — change matrices if you use another  $\lambda$  or glass.
2. **Different element positions / spacings / orientations:** moving a lens or changing a thickness means different translation/refraction matrices.
3. **Tilted / decentered elements / off-axis 3D:** pure  $2 \times 2$  paraxial scalar matrices are per-axis and assume coaxial, centered elements. For tilts or decenter you need coordinate transforms (rotation + translation) — the matrices change.
4. **Non-paraxial rays / large angles / aberrations / aspheres:** linear paraxial matrices are approximations. For large angles or to capture higher-order aberrations

you need nonlinear maps (ray-surface intersections + Snell's law) — not constant linear matrices.

5. **Aperture/vignetting/clipping:** the matrix tells you where the ray *would* go; you still must test whether the ray clears each aperture (this is a separate geometric test — it doesn't change the optical matrices themselves).
6. **Polarization/coatings or thin-film phase effects:** if you model polarization-dependent Fresnel coefficients, then per-surface amplitude/phase depends on angle and wavelength — require more than the simple  $2 \times 2$  paraxial matrices or you must update coefficients per ray.

Multi-axis (2D) case

For full x/y rays you typically use *two* independent paraxial  $2 \times 2$  matrices (one for the x-plane and one for y), or assemb

le a block-diagonal  $4 \times 4$  if you prefer:

$$\mathbf{r} = \begin{bmatrix} y_x \\ n\theta_x \\ y_y \\ n\theta_y \end{bmatrix}, M_{4 \times 4} = \begin{bmatrix} M_x & 0 \\ 0 & M_y \end{bmatrix}.$$

If the system is rotationally symmetric then  $M_x = M_y$  and again they are constant for fixed geometry.

Quick numerical example (paraxial; same matrices, different rays)

```
import numpy as np
```

```
# Example system: refraction at surface1 -> travel 5 mm in glass -> refraction at surface2
```

```
def refraction_matrix(n1, n2, R):
```

```
    phi = (n2 - n1) / R
    return np.array([[1.0, 0.0],
                    [-phi, 1.0]])
```

```
def translation_matrix(d, n):
```

```
    return np.array([[1.0, d / n],
                    [0.0, 1.0]])
```

```
n_air = 1.0
```

```

n_glass = 1.5168
R = 50.0 # mm, symmetric biconvex example
t = 5.0 # mm thickness

S1 = refraction_matrix(n_air, n_glass, +R)
T = translation_matrix(t, n_glass)
S2 = refraction_matrix(n_glass, n_air, -R)

Msys = S2 @ T @ S1 # fixed system matrix for these geometry & wavelength

# Two different rays (different heights / angles)
ray1 = np.array([ 5.0, n_air * 0.0]) # y=5 mm, parallel (theta=0)
ray2 = np.array([10.0, n_air * 0.01]) # y=10 mm, small incoming angle

out1 = Msys @ ray1
out2 = Msys @ ray2

print("Msys =\n", Msys)
print("Ray1 out:", out1, " -> angle (deg):", np.degrees(out1[1]/n_air))
print("Ray2 out:", out2, " -> angle (deg):", np.degrees(out2[1]/n_air))

```

Practical workflow

- Build element matrices once (for the chosen wavelength and coordinate reference planes).
- Compose them to form  $M_{\text{sys}}$  between your chosen input and image/reference planes.
- For each sample ray, form the input vector and multiply:  $r_{\text{out}} = M_{\text{sys}} @ r_{\text{in}}$ .
- After multiplication: check aperture clearances and record intersection with detector plane (if you need real-space coordinates shift/translation to that plane).
- If you change wavelength, lens positions, or tilt/decenter, recompute matrices first.

If you tell me which mode you want to run (paraxial only for many field points; or non-paraxial physical ray-surface tracing), I can give a compact code snippet that:

- builds  $M_{\text{sys}}$  and traces many paraxial rays (fast), or

- builds a sequential non-paraxial tracer that recomputes intersections per ray (slower but accurate). Which do you want?

I'm trying to write a software in matlab that draws a ray until it hits an intersection with another curve and then tells me what is that point of intersection. I should be able to pick the starting position and angles

ChatGPT said:

Got it — that's a great MATLAB project! Here's a step-by-step outline and example code you can build on.

### 🎯 Goal

Draw a ray starting from a user-defined point and angle, find its intersection with one or more curves, and display that intersection point.

#### 1. Define your curves

For simplicity, let's assume your other curve(s) are functions  $y = f(x)$ .

Example:

```
f = @(x) 0.2*x.^2 + 1; % Example curve
```

#### 2. Define the ray

A ray can be expressed parametrically from a starting point  $(x_0, y_0)$  and angle  $\theta$ :

$$\begin{aligned}x &= x_0 + t \cos(\theta) \\ y &= y_0 + t \sin(\theta)\end{aligned}$$

for  $t \geq 0$ .

#### 3. Find the intersection

Substitute the ray equations into the curve equation and solve for  $t$ .

Example:

```
x0 = 0; % starting x
y0 = 0; % starting y
theta = 45; % degrees
```

```

theta = deg2rad(theta);

f = @(x) 0.2*x.^2 + 1;

ray_y = @(x) y0 + (x - x0)*tan(theta);

% Solve f(x) = ray_y(x)
eq = @(x) f(x) - ray_y(x);

% Use fzero near a reasonable guess
x_guess = 2;
x_intersect = fzero(eq, x_guess);

y_intersect = f(x_intersect);

4. Visualize it
% Plot curve
x = linspace(-5,5,200);
plot(x, f(x), 'b', 'LineWidth', 1.5);
hold on;

% Plot ray
t = linspace(0,10,100);
x_ray = x0 + t*cos(theta);
y_ray = y0 + t*sin(theta);
plot(x_ray, y_ray, 'r--');

% Plot intersection
plot(x_intersect, y_intersect, 'ko', 'MarkerFaceColor', 'k');

```

```

legend('Curve', 'Ray', 'Intersection');
xlabel('x'); ylabel('y');
axis equal;
grid on;

```

#### 5. Make it interactive (optional)

You can let the user click a starting point and input an angle:

```

figure;
fplot(f, [-5 5]); hold on;
[x0, y0] = ginput(1); % Click start point
theta = input('Enter ray angle (degrees): ');
theta = deg2rad(theta);

```

Would you like me to extend this so it works for **multiple arbitrary curves** (e.g., circles, polygons, or splines) and automatically finds the *nearest* intersection along the ray?

You said:

I'm using this in the context of ray tracing for lenses. I'm only using bi vonvex lenses with known center thickness Diameter radius of curvature for both sides

ChatGPT said:

Perfect — that's a great use case! Since you're working with **bi-convex lenses**, you can model each lens surface as a **circle (or spherical arc)** with a known **radius of curvature**, and then compute where the ray intersects each surface.

Let's break down how to do this step-by-step in MATLAB.

#### Known parameters

For a **bi-convex lens**, you typically know:

Symbol	Meaning
$R_1$	Radius of curvature of front surface (positive if center of curvature is to the right)

Symbol	Meaning
$R_2$	Radius of curvature of back surface (positive if center is to the right, usually negative for a bi-convex lens)
$t_c$	Center thickness
$D$	Diameter of the lens
$n$	Refractive index (if you plan to trace through it)

You can derive the center coordinates of each circular surface from this data.

### ✚ 1. Geometry setup

Let's assume the optical axis is the **x-axis**, and the lens is centered at the origin.

Then:

- The **front surface** (left side) has its center at  $(x_{c1}, 0)$
- The **back surface** (right side) has its center at  $(x_{c2}, 0)$

Positions can be found as:

$$\begin{aligned}x_{c1} &= -R_1 + t_c/2 \\x_{c2} &= R_2 - t_c/2\end{aligned}$$

The surfaces (arcs) are given by:

$$(x - x_{c1})^2 + y^2 = R_1^2$$

and

$$(x - x_{c2})^2 + y^2 = R_2^2$$

### 🎨 2. Ray definition

A ray starting at  $(x_0, y_0)$  with angle  $\theta$  from the x-axis:

$$\begin{aligned}x &= x_0 + t \cos(\theta) \\y &= y_0 + t \sin(\theta)\end{aligned}$$

### ⚙️ 3. Find intersection with a circular surface

Substitute ray equations into the circle equation and solve for  $t$ :

$$(x_0 + t \cos \theta - x_c)^2 + (y_0 + t \sin \theta)^2 = R^2$$

This gives a quadratic equation in  $t$ .

#### 4. MATLAB implementation

Here's a general-purpose function that finds the intersection(s) between a ray and a circle:

```
function [t_vals, points] = ray_circle_intersection(x0, y0, theta, xc, yc, R)
    % Ray parametric form
    dx = cos(theta);
    dy = sin(theta);

    % Quadratic coefficients
    A = dx^2 + dy^2;
    B = 2*((x0 - xc)*dx + (y0 - yc)*dy);
    C = (x0 - xc)^2 + (y0 - yc)^2 - R^2;

    % Solve A*t^2 + B*t + C = 0
    D = B^2 - 4*A*C;
    t_vals = [];
    points = [];

    if D >= 0
        t1 = (-B + sqrt(D)) / (2*A);
        t2 = (-B - sqrt(D)) / (2*A);
        t_vals = [t1 t2];
        % Compute intersection points
        points = [x0 + t_vals*cos(theta); y0 + t_vals*sin(theta)];
    end
end
```

## 5. Example with a bi-convex lens

```
% Lens parameters
R1 = 50;      % mm
R2 = -50;    % mm
tc = 10;     % mm
D = 25;      % mm

% Centers of curvature
xc1 = -R1 + tc/2;
xc2 = R2 - tc/2;

% Ray
x0 = -100; y0 = 0; theta = deg2rad(5);

% Intersections
[t1, pts1] = ray_circle_intersection(x0, y0, theta, xc1, 0, abs(R1));
[t2, pts2] = ray_circle_intersection(x0, y0, theta, xc2, 0, abs(R2));

% Keep forward intersections only (t > 0)
pts1 = pts1(t1>0, :);
pts2 = pts2(t2>0, :);

% Pick the nearest intersection (smallest positive t)
[~, idx1] = min(t1(t1>0));
p1 = pts1(idx1,:);

[~, idx2] = min(t2(t2>0));
p2 = pts2(idx2,:);
```

```

% Display
figure; hold on; axis equal; grid on;
theta_surf = linspace(-pi/2, pi/2, 200);
plot(xc1 + R1*cos(theta_surf), R1*sin(theta_surf), 'b'); % front
plot(xc2 + R2*cos(theta_surf), R2*sin(theta_surf), 'b'); % back
plot(x0 + [0 200]*cos(theta), y0 + [0 200]*sin(theta), 'r--'); % ray
plot(p1(1), p1(2), 'ko', 'MarkerFaceColor', 'k');
plot(p2(1), p2(2), 'ko', 'MarkerFaceColor', 'k');
xlabel('x'); ylabel('y');
legend('Front surface','Back surface','Ray','Intersections');

```

✅ This gives you:

- Both lens surfaces (circular arcs)
- The intersection points of your ray with the surfaces
- Visual confirmation via plot

Would you like me to extend this so that after the first intersection, the ray **refracts** (Snell's Law) and continues to the **second surface** automatically — so you can trace through the lens?

I asked:

Why did you add numeric offsets i need this code to be exact

ChatGPT said:

Thought for 4 seconds

Good — fair point. I added small numeric *moves* earlier only to avoid the ray immediately re-intersecting the same surface because of floating-point roundoff. If you need the geometry to remain *exact* (no artificial position shifts), we can remove those shifts and instead handle the numerical issue *inside* the intersection routine by requiring the solver to return only intersections with  $t \geq t\_min$  (a tiny positive threshold). That keeps the geometry exact while preventing zero-length hits from being counted.

Below is a corrected, ready-to-run MATLAB script that:

- Removes explicit position offsets (no `eps_move` translations).
- Uses a robust `t_min` argument to `ray_circle_intersection` so intersections at or extremely near the current point are ignored.

- Traces any number of **biconvex lenses** (format: [x0, R, thickness, diameter, n]).
- Applies refraction at both front and back faces exactly.
- Keeps the math exact (no artificial repositioning). (Note: tiny numerical tolerances are unavoidable in floating-point arithmetic; using a t\_min threshold is the standard way to avoid degenerate intersections without altering geometry.)

Copy-paste this whole script into MATLAB:

```
%%
```

```
=====
% Multi-Biconvex-Lens Ray Tracer (2D) - No artificial offsets
% - Lenses are biconvex: row = [x0, R, thickness, diameter, n]
% - No explicit position translation; intersection function uses t_min
%
=====
```

```
clear; close all; clc;
```

```
figure('Color','w'); hold on; axis equal; grid on;
xlabel('x'); ylabel('y');
title('Ray tracing through multiple biconvex lenses (no offsets)');
```

```
%% ----- User: define lenses -----
```

```
% Each row = [x0, R, thickness, diameter, n_lens]
```

```
lenses = [
    0, 29.5, 7.7, 25.4, 1.5; % lens 1 at x0 = 0
    50, 25.0, 6.0, 25.4, 1.6; % lens 2 at x0 = 50
];
```

```
n_lenses = size(lenses,1);
```

```
%% ----- Plot lens outlines -----
```

```
for i = 1:n_lenses
    x0 = lenses(i,1); R = lenses(i,2); t_c = lenses(i,3); D = lenses(i,4);
    y_plot = linspace(-D/2, D/2, 400);
```

```

x_front = x0 - (t_c/2 - R) - sqrt(R^2 - y_plot.^2);
x_back = x0 + (t_c/2 - R) + sqrt(R^2 - y_plot.^2);
plot(x_front, y_plot, 'b', 'LineWidth', 2);
plot(x_back, y_plot, 'b', 'LineWidth', 2);
end

%% ----- Ray source -----
x_start = -30;          % source x
y_start = 0;           % source y (point source)
cone_half_angle = 8 * pi/180; % half-angle (radians)
n_rays = 11;           % number of rays
theta_rays = linspace(-cone_half_angle, cone_half_angle, n_rays);
y_rays = y_start * ones(1, n_rays);

n_air = 1;

% intersection tolerance (minimum forward t to accept)
t_min = 1e-8;

%% ----- Trace rays through lenses -----
for k = 1:n_rays
    % initial ray state (in air)
    x_curr = x_start;
    y_curr = y_rays(k);
    theta = theta_rays(k);
    dir_vec = [cos(theta); sin(theta)];
    medium_n = n_air;

    ray_alive = true;

```

```

for L = 1:n_lenses
    % extract lens parameters
    x0 = lenses(L,1); R = lenses(L,2); t_c = lenses(L,3); D = lenses(L,4); n_1 =
lenses(L,5);

    % compute front and back centers (biconvex)
    x_center_front = x0 - (t_c/2 - R);
    x_center_back = x0 + (t_c/2 - R);

    %% --- INTERSECT FRONT SURFACE (ray in air -> front) ---
    theta = atan2(dir_vec(2), dir_vec(1));
    [x_front_hit, y_front_hit, t_front] = ray_circle_intersection(x_curr, y_curr,
theta, x_center_front, 0, R, D, t_min);
    if isnan(x_front_hit)
        ray_alive = false;
        break;
    end
    plot([x_curr x_front_hit],[y_curr y_front_hit],'r','LineWidth',1);

    % normal at front hit (from center to hit)
    n_front = [(x_front_hit - x_center_front); (y_front_hit - 0)];
    n_front = n_front / norm(n_front);

    % make normal point against incoming direction
    if dot(n_front, dir_vec) > 0
        n_front = -n_front;
    end

    % refract: air -> lens
    t_in = refract_vector(dir_vec, n_front, n_air, n_1);

```

```

if any(isnan(t_in))
    % TIR (unlikely air->glass), stop ray
    ray_alive = false;
    break;
end

%% --- INTERSECT BACK SURFACE (ray inside lens -> back) ---
theta_in = atan2(t_in(2), t_in(1));
% use the same t_min so intersections at the front point are excluded
[x_back_hit, y_back_hit, t_back] = ray_circle_intersection(x_front_hit,
y_front_hit, theta_in, x_center_back, 0, R, D, t_min);
if isnan(x_back_hit)
    ray_alive = false;
    break;
end
plot([x_front_hit x_back_hit],[y_front_hit y_back_hit],'g','LineWidth',1);

% normal at back hit (center to hit)
n_back = [(x_back_hit - x_center_back); (y_back_hit - 0)];
n_back = n_back / norm(n_back);

% normal should point out of lens (opposite of incoming inside ray)
if dot(n_back, t_in) < 0
    n_back = -n_back;
end

% refract: lens -> air
t_out = refract_vector(t_in, n_back, n_l, n_air);
if any(isnan(t_out))
    % total internal reflection inside lens: terminate ray

```

```

        ray_alive = false;
        break;
    end

    % Update current ray state (exactly at exit point, no offsets)
    x_curr = x_back_hit;
    y_curr = y_back_hit;
    dir_vec = t_out;
    medium_n = n_air;

    % For visualization, draw the immediate exiting segment (short)
    x_vis = x_curr + 0.01 * dir_vec(1); % tiny visual marker, not geometry change
    y_vis = y_curr + 0.01 * dir_vec(2);
    plot([x_back_hit x_vis],[y_back_hit y_vis],'b','LineWidth',1);
end

% After all lenses, extend final ray to the right for visualization
if ray_alive
    x_end = x_curr + 80 * dir_vec(1);
    y_end = y_curr + 80 * dir_vec(2);
    plot([x_curr x_end],[y_curr y_end],'b','LineWidth',1.2);
end
end

xlim([-40, max(lenses(:,1)) + 80]);
ylim([-max(lenses(:,4))/2 - 5, max(lenses(:,4))/2 + 5]);

%% ===== Helper functions =====

```

```

function [x_hit, y_hit, t_hit] = ray_circle_intersection(x0_ray, y0_ray, theta, x_c, y_c,
R, D, t_min)
    % Intersection of ray (x0_ray,y0_ray) along theta with circle (x_c,y_c), radius R.
    % Returns (x_hit,y_hit) and the forward parameter t_hit.
    % Only returns intersections with t >= t_min (to avoid degeneracy when starting
on surface).

    dx = cos(theta); dy = sin(theta);
    a = dx^2 + dy^2;
    b = 2*((x0_ray - x_c)*dx + (y0_ray - y_c)*dy);
    c = (x0_ray - x_c)^2 + (y0_ray - y_c)^2 - R^2;
    disc = b^2 - 4*a*c;
    if disc < 0
        x_hit = NaN; y_hit = NaN; t_hit = NaN; return;
    end
    t1 = (-b + sqrt(disc)) / (2*a);
    t2 = (-b - sqrt(disc)) / (2*a);
    t_candidates = sort([t1, t2]);    % smallest first
    t_forward = t_candidates(t_candidates >= t_min); % only accept forward
intersections >= t_min
    if isempty(t_forward)
        x_hit = NaN; y_hit = NaN; t_hit = NaN; return;
    end
    t_hit = min(t_forward);
    x_hit = x0_ray + t_hit * dx;
    y_hit = y0_ray + t_hit * dy;
    if abs(y_hit - y_c) > D/2
        x_hit = NaN; y_hit = NaN; t_hit = NaN;
    end
end
end

```

```

function t = refract_vector(i, n, n1, n2)

```

```

% Vector Snell's law. i,n can be any length vectors; they're normalized inside.
i = i / norm(i);
n = n / norm(n);
cos_theta_i = -dot(n, i);
% compute the argument for cos_theta_t robustly
k = 1 - (n1/n2)^2 * (1 - cos_theta_i^2);
if k < 0
    t = NaN(2,1); return; % TIR
end
cos_theta_t = sqrt(max(0,k));
t = (n1/n2)*i + ((n1/n2)*cos_theta_i - cos_theta_t) * n;
t = t / norm(t);
end

```

Short explanation of the change

- Previously I advanced the origin slightly after a surface hit to avoid hitting that same surface again. That *is* an artificial geometry change.
- The version above keeps the hit coordinates exact and instead requires that subsequent intersections have  $t \geq t_{\min}$  (tiny positive threshold). This is the standard exact-geometry approach — it avoids counting the trivial  $t=0$  solution while preserving geometric correctness.
- You can reduce  $t_{\min}$  (e.g. to  $1e-10$ ) if you want even stricter behavior, but be aware of floating-point precision limits.

If you want I can:

- Change the lens input to allow **asymmetric surfaces** [x0, R1, R2, thickness, D, n].
- Add automatic **focal point estimation** (compute intersection of exiting rays).
- Add diagnostics that report the numeric t values when intersections are found so you can inspect exactness.

Which of those (if any) do you want next?

You said:

This code work way less than before. Dont simplify the radius of curvature to 1 number. Also the ray coming off the back edge of lens 1 is gone and the rest of the system shows nothing. Figure out why this isnt happening. the goal was to design the code in a way that if you can get a ray through 1 biconvex lens correctly all you would need to do is pretend the backside intersection of is the start point of the ray for lens 2

ChatGPT said:

Thought for 15s

Nice — thanks for the clear report. You were right: the problems came from two things I changed that broke the geometry:

1. **I collapsed R1 and R2 into one R** which changed the back-surface geometry for asymmetric lenses (and your second lens had a different surface than the first), so the back-surface center / intersection positions were wrong and rays missed.
2. My handling of the normal-orientation / intersection-threshold logic in the later edits was inconsistent with the Snell-vector convention and the desired exact-geometry flow (start each lens using the previous *back-hit* exactly). That caused some rays to be terminated prematurely.

What you asked for — exact geometry, no artificial position offsets, and the ability to chain lenses such that the back-hit of lens  $i$  is the start for lens  $i+1$  — is now implemented correctly.

Below is a complete, ready-to-run MATLAB script. It:

- Uses lens rows of the full form: [x0, R1, R2, thickness, D, n\_lens] (so R1 and R2 are preserved per lens)
- Applies refraction at **both** front and back surfaces of each lens.
- **Does not** apply any artificial geometric offsets.
- Uses a small  $t_{\min}$  inside the intersection routine to avoid counting  $t \approx 0$  intersections while preserving exact geometry.
- Chains lenses so the back intersection point of lens  $i$  is the exact start point for lens  $i+1$ .
- Uses a robust normal-orientation rule consistent with the refract routine.

Copy–paste and run this whole script in MATLAB:

```
%%
```

```
=====
% Multi-Lens Ray Tracer (exact geometry, R1/R2 per lens)
% Lens row format: [x0, R1, R2, thickness, diameter, n_lens]
% No artificial position offsets. The back-hit of lens i becomes
% the start point for lens i+1 exactly (subject to numerical tol).
%
=====
```

```
clear; close all; clc;
```

```
figure('Color','w'); hold on; axis equal; grid on;
```

```
xlabel('x'); ylabel('y');
```

```
title('Ray tracing through multiple biconvex/asymmetric lenses (exact geometry)');
```

```

%% ----- User: define lenses -----
% Each row = [x0, R1, R2, thickness, diameter, n_lens]
% R1: radius of front surface (positive for convex as seen from left)
% R2: radius of back surface (positive for convex as seen from right)
% Example: two lenses (biconvex or any R1,R2)
lenses = [
    0, 29.5, 29.5, 7.7, 25.4, 1.5; % lens 1 at x0 = 0
    50, 25.0, 25.0, 6.0, 25.4, 1.6; % lens 2 at x0 = 50
];
n_lenses = size(lenses,1);

%% ----- Plot lens outlines -----
n_plot = 400;
for i = 1:n_lenses
    x0 = lenses(i,1); R1 = lenses(i,2); R2 = lenses(i,3);
    t_c = lenses(i,4); D = lenses(i,5);
    y_plot = linspace(-D/2, D/2, n_plot);

    % front/back x coordinates (exact geometry)
    x_center_front = x0 - (t_c/2 - R1);
    x_center_back = x0 + (t_c/2 - R2);
    x_front = x_center_front + sign(R1)*sqrt(R1^2 - y_plot.^2); % circle x = xc ±
sqrt(R^2 - y^2)
    x_front = x_center_front - (R1 - sqrt(R1^2 - y_plot.^2)); % alternative: same
shape consistent with earlier code
    x_back = x_center_back - sign(R2)*sqrt(R2^2 - y_plot.^2);
    x_back = x_center_back + (R2 - sqrt(R2^2 - y_plot.^2)); % keep consistent
orientation

    % The above two equivalent forms are included to avoid sign confusion;

```

```

% we plot the two surface curves consistently:
% (use the same formula as used when originally plotting single-lens)
x_front = x0 - (t_c/2 - R1) - sqrt(R1^2 - y_plot.^2);
x_back = x0 + (t_c/2 - R2) + sqrt(R2^2 - y_plot.^2);

plot(x_front, y_plot, 'b', 'LineWidth', 2);
plot(x_back, y_plot, 'b', 'LineWidth', 2);
end

%% ----- Ray source -----
x_start = -30;          % source x
y_start = 0;           % source y (point source)
cone_half_angle = 8 * pi/180; % half-angle (radians)
n_rays = 11;           % number of rays
theta_rays = linspace(-cone_half_angle, cone_half_angle, n_rays);
y_rays = y_start * ones(1, n_rays);

n_air = 1;

% intersection tolerance (minimum forward t to accept). Keep very small but > 0
t_min = 1e-9;

%% ----- Trace rays through lenses -----
for k = 1:n_rays
    % initial ray state (in air)
    x_curr = x_start;
    y_curr = y_rays(k);
    theta = theta_rays(k);
    dir_vec = [cos(theta); sin(theta)];
    medium_n = n_air;

```

```

alive = true;

for L = 1:n_lenses
    % extract lens parameters
    x0 = lenses(L,1); R1 = lenses(L,2); R2 = lenses(L,3);
    t_c = lenses(L,4); D = lenses(L,5); n_1 = lenses(L,6);

    % compute circle centers for front and back
    x_center_front = x0 - (t_c/2 - R1);
    x_center_back = x0 + (t_c/2 - R2);

    %% --- FRONT SURFACE INTERSECTION ---
    % Ray in medium 'medium_n' hits front circle (center x_center_front, radius
R1)
    theta = atan2(dir_vec(2), dir_vec(1));
    [x_front_hit, y_front_hit, t_front] = ray_circle_intersection( ...
        x_curr, y_curr, theta, x_center_front, 0, R1, D, t_min);
    if isnan(x_front_hit)
        alive = false;
        break;
    end
    plot([x_curr x_front_hit], [y_curr y_front_hit], 'r', 'LineWidth', 1);

    % normal at front (from center to hit)
    n_front = [x_front_hit - x_center_front; y_front_hit - 0];
    n_front = n_front / norm(n_front);

    % orient normal so that cos_theta_i = -dot(n, dir_vec) >= 0
    if -dot(n_front, dir_vec) < 0
        n_front = -n_front;

```

```

end

% refract from medium_n -> n_1 (front refraction)
t_inside = refract_vector(dir_vec, n_front, medium_n, n_1);
if any(isnan(t_inside))
    % total internal reflection (unlikely for air->glass)
    alive = false;
    break;
end

%% --- BACK SURFACE INTERSECTION ---
% Ray inside lens (index n_1) intersects back circle (center x_center_back,
radius R2)
theta_in = atan2(t_inside(2), t_inside(1));
[x_back_hit, y_back_hit, t_back] = ray_circle_intersection( ...
    x_front_hit, y_front_hit, theta_in, x_center_back, 0, R2, D, t_min);
if isnan(x_back_hit)
    alive = false;
    break;
end

plot([x_front_hit x_back_hit], [y_front_hit y_back_hit], 'g', 'LineWidth', 1);

% normal at back (from center to hit)
n_back = [x_back_hit - x_center_back; y_back_hit - 0];
n_back = n_back / norm(n_back);

% orient normal so that cos_theta_i = -dot(n_back, t_inside) >= 0
if -dot(n_back, t_inside) < 0
    n_back = -n_back;
end

```

```

% refract from lens -> air (or next medium; here we assume air between lenses)
t_exit = refract_vector(t_inside, n_back, n_l, n_air);
if any(isnan(t_exit))
    % total internal reflection inside lens -> terminate
    alive = false;
    break;
end

% Update current state EXACTLY at back hit (no offset)
x_curr = x_back_hit;
y_curr = y_back_hit;
dir_vec = t_exit;
medium_n = n_air;

% For visual continuity plot a short visible tiny segment in blue (very small,
doesn't alter geometry)
% This is only for plotting and not used for geometry or intersections.
len_vis = 1e-3; % tiny visual length
plot([x_back_hit, x_back_hit + len_vis*dir_vec(1)], [y_back_hit, y_back_hit +
len_vis*dir_vec(2)], 'b', 'LineWidth', 1);
% Now x_curr,y_curr,dir_vec are exactly the starting conditions for the next
lens.
end

% If ray survived through all lenses, extend it to the right for visualization
if alive
    x_end = x_curr + 200 * dir_vec(1);
    y_end = y_curr + 200 * dir_vec(2);
    plot([x_curr x_end], [y_curr y_end], 'b', 'LineWidth', 1.2);
end

```

```

end

% set axes limits nicely
xlim([-40, max(lenses(:,1)) + 120]);
ylim([-max(lenses(:,5))/2 - 5, max(lenses(:,5))/2 + 5]);

%% ===== Helper functions =====

function [x_hit, y_hit, t_hit] = ray_circle_intersection(x0_ray, y0_ray, theta, x_c, y_c,
R, D, t_min)
    % Intersection of ray (x0_ray,y0_ray) with circle (x_c,y_c), radius R.
    % Returns forward intersection with t >= t_min.
    dx = cos(theta); dy = sin(theta);
    a = dx^2 + dy^2;
    b = 2*((x0_ray - x_c)*dx + (y0_ray - y_c)*dy);
    c = (x0_ray - x_c)^2 + (y0_ray - y_c)^2 - R^2;
    disc = b^2 - 4*a*c;
    if disc < 0
        x_hit = NaN; y_hit = NaN; t_hit = NaN; return;
    end
    t1 = (-b + sqrt(disc)) / (2*a);
    t2 = (-b - sqrt(disc)) / (2*a);
    t_candidates = sort([t1, t2], 'ascend'); % smaller first
    % pick the smallest forward t >= t_min
    forward = t_candidates(t_candidates >= t_min);
    if isempty(forward)
        x_hit = NaN; y_hit = NaN; t_hit = NaN; return;
    end
    t_hit = min(forward);
    x_hit = x0_ray + t_hit * dx;

```

```

y_hit = y0_ray + t_hit * dy;
% check aperture (diameter)
if abs(y_hit - y_c) > D/2
    x_hit = NaN; y_hit = NaN; t_hit = NaN;
end
end
end

```

```

function t = refract_vector(i, n, n1, n2)
    % Vector Snell's law. Inputs can be non-normalized; they are normalized inside.
    i = i / norm(i);
    n = n / norm(n);
    cos_theta_i = -dot(n, i);
    % compute discriminant for transmitted cosine
    k = 1 - (n1/n2)^2 * (1 - cos_theta_i^2);
    if k < 0
        t = NaN(2,1); return; % total internal reflection
    end
    cos_theta_t = sqrt(max(0,k));
    t = (n1/n2)*i + ((n1/n2)*cos_theta_i - cos_theta_t) * n;
    t = t / norm(t);
end

```

Why this should fix the issues you reported

- **R1 and R2 are both preserved** per lens, so each surface center is correct and intersections won't be missed because of the wrong curvature.
- **No artificial translation** of the ray origin is used — we start the next lens exactly at the previous back-hit ( $x_{curr} = x_{back\_hit}$ ), as you wanted.
- **The intersection routine uses  $t_{min}$**  (tiny positive forward parameter) so we don't accidentally re-detect the same surface. This keeps the geometry exact while avoiding the numerical 0-root.
- **Normal orientation is set by the same rule** before every refraction step: ensure  $\cos\_theta\_i = -\text{dot}(n,i)$  is nonnegative, so the refract function sees consistent signs.

If you want, next I can:

- Add an option to treat the medium *between* lenses as something other than air (i.e. specify a list of inter-lens refractive indices).
- Add a diagnostic mode that prints each intersection (x,y) and the t parameter for debugging.
- Add a small routine to estimate where exiting rays cross (focal point estimation).

Which of those would you like next?

## Appendix B.2 Case Study 2

I am trying to design an epillumination darkfield microscope that can detect 200 micron particles. This is important. I need to select the camera used ,the objective lens required, the magnification, and the tube lens and the beam splitter

ChatGPT said:

Thought for 20s

Nice — this is a fun, practical optics problem. Below I give a complete, ready-to-use design recipe for an **epi-illumination darkfield microscope** aimed at reliably *detecting* and imaging **200  $\mu\text{m}$**  particles. I'll explain the reasoning, show the math you need (so you can swap parts later), and finish with concrete component choices and tradeoffs (camera spec, objective, tube lens, beam-splitter, illumination approach).

If you want, I can immediately convert this into a shopping list with vendor part numbers next.

Quick summary (if you want the short recommendation right away)

- **Objective:** Infinity-corrected long-working-distance (LWD) **2 $\times$  – 4 $\times$  Plan objective**, NA  $\approx$  **0.05–0.13**, WD  $\geq$  **10–20 mm** (prefer LWD).
- **Tube lens:** Match the objective family (commonly **200 mm** tube lens for Nikon/TI/most infinity objectives; check objective datasheet).
- **Magnification on camera:** **2 $\times$**  (or 4 $\times$  if you need higher detail). 2 $\times$  is usually enough for 200  $\mu\text{m}$  particles.
- **Camera:** Scientific CMOS (global shutter preferred) with **pixel size 3.45–5  $\mu\text{m}$** , sensor format **1" or 2/3"**, resolution  $\geq$  **2 MP**, USB3/GigE. (Example spec: 1" sensor, 2048 $\times$ 1536, 3.45  $\mu\text{m}$  pixels, global shutter.)
- **Beam splitter:** **Pellicle beamsplitter** (low ghosting) or **non-polarizing cube 50/50** for simplicity. If you use a narrowband LED, consider a **dichroic** tuned to that LED (best throughput).
- **Illumination:** Epi-annular (ring) LED or fiber ring illuminating through the beam-splitter with an annular stop conjugate to objective back focal plane (BFP) to produce darkfield.

Design reasoning + calculations

1) What does “detect a 200  $\mu\text{m}$  particle” mean for magnification & sampling?

You want the particle big enough on the camera sensor to be detected and measured. Use this rule: choose magnification  $M$  so that the particle spans a reasonable number of pixels. Typical practical targets:

- coarse detection:  $\sim 6\text{--}10$  pixels across
- comfortable measurement/shape detail:  $\sim 20\text{--}50$  pixels across

Formula:

$$\text{pixels across} = \frac{M \times D_{\text{particle}}}{p}$$

where  $D_{\text{particle}} = 200 \mu\text{m}$  and  $p =$  camera pixel size.

Example numeric cases:

- camera pixel  $p = 3.45 \mu\text{m}$  (common sCMOS)
  - $M = 1\times \rightarrow$  pixels =  $1 \times 200/3.45 \approx 58\text{px}$  (excellent)
  - $M = 2\times \rightarrow \approx 116\text{px}$  (oversampled but great for analysis)
  - $M = 0.5\times \rightarrow \approx 29\text{px}$  (still OK)
- pixel  $p = 5.86 \mu\text{m}$  (some industrial CMOS)
  - $M = 1\times \rightarrow \approx 34\text{px}$  (good)
  - $M = 0.5\times \rightarrow \approx 17\text{px}$  (acceptable for detection)

Conclusion: for a **200  $\mu\text{m}$**  particle you do **not** need high microscope magnification. Even **1 $\times$**  imaging with a 3–6  $\mu\text{m}$  pixel gives many pixels across the particle. So a **2 $\times$  or 4 $\times$  objective** is already safe; use a **1 $\times$ –2 $\times$**  objective if you want a larger field-of-view (FOV) and still good sampling.

2) Field of view (FOV) planning

FOV on object = sensor physical size /  $M$ . Example:

- 1" sensor diagonal  $\approx 16\text{ mm}$ ; sensor width  $\approx 12.8\text{ mm}$  (depends on sensor).
- With  **$M = 2\times$** , FOV width  $\approx 12.8\text{ mm} / 2 = \mathbf{6.4\text{ mm}}$  (plenty to see many 200  $\mu\text{m}$  particles at once).
- With  **$M = 4\times$** , FOV width  $\approx 3.2\text{ mm}$ .

So pick sensor format depending on how many particles you want in one frame: 1" or APS-C give larger fields.

3) Objective NA & darkfield behavior — what to pick

Darkfield requires that the unscattered illumination not enter the objective; only scattered light is collected. For *epi-darkfield* we place an annular illumination so the illumination cone misses the objective acceptance.

Practical guidance:

- Use an infinity-corrected objective (so you can insert optics, beam splitter, tube lens).
- For mechanical clearance and noncontact samples (particles larger than 200  $\mu\text{m}$ ), choose **long working distance** (LWD) objective ( $\text{WD} \geq 10\text{--}20\text{ mm}$ ).
- NA can be modest: **0.05–0.2** is typical for low mag  $2\times\text{--}4\times$  objectives. Lower NA makes it easier to keep direct illumination out of the objective (i.e., simpler darkfield). Higher NA collects more scattered light and gives higher resolution — but it makes darkfield illumination design trickier (annulus must be outside NA).
- I recommend  $2\times$  or  $4\times$  LWD infinity objective,  $\text{NA} \approx 0.08\text{--}0.13$ , Plan corrected.

#### 4) Tube lens matching

Use the tube lens focal length required by your objective brand:

- Many objectives expect a **200 mm** tube lens (Nikon/others), Olympus uses **180 mm**, Zeiss sometimes **165 mm**. **Always** confirm on the objective datasheet. If you pick a generic infinity objective, use the manufacturer-recommended tube lens.  $M$  (objective stated mag)  $\times$  (tube\_lens\_focal / objective\_design\_focal) = effective mag — but for standard parts just use matched tube lens (e.g., 200 mm).

#### 5) Beam splitter choice (epi darkfield)

You must route illumination into the objective axis and collect scattered light back out to the camera:

Options (tradeoffs):

1. **Pellicle beamsplitter** (thin membrane): very low ghosting and low optical path displacement, moderate reflectivity choices (e.g., 10/90, 30/70). Good for darkfield because it minimizes ghost reflections that can brighten the background. Use if you want minimal image artifacts.
  - Use e.g., **30R/70T** or **50/50** depending on illumination brightness.
2. **Non-polarizing 50/50 cube**: robust and cheap, symmetric (50% illumination sent down, 50% of scattered transmitted toward camera). More ghosts than pellicle, but very common.
3. **Dichroic beamsplitter**: useful if you choose **narrowband LED** illumination — pick a dichroic that reflects LED band into the objective while transmitting scattered light to camera. Best throughput, but only works for narrowband (fluorescence-style) setups.

Practical recommendation: For white/broadband darkfield LED, start with a **pellicle ~30/70** (reflect ~30% to sample, transmit 70% back to camera) or **50/50 NPBS cube** if cost/robustness matters. If stray reflections are troublesome, use pellicle.

## 6) Illumination geometry for epi-darkfield

You need to create an annular illumination conjugate to the objective back focal plane (BFP) so rays missing objective NA illuminate the sample and only scattered light enters objective.

Implementation approaches:

- **Annular LED ring** placed at a conjugate plane to the objective BFP (use relay lenses or an aperture at a Köhler conjugate plane).
- Fiber bundle ring + condenser optics through the beam splitter.
- **Commercial epi-darkfield illuminator** (if budget allows) fits into microscope cube and produces annular illumination.

For prototyping, a bright monochromatic LED (e.g., 530 nm) + ring optic + condenser lens + small aperture to form an annulus works well. Using narrowband simplifies stray light control and makes dichroic option viable.

## 7) Camera selection (specs + why)

Key specs:

- **Pixel size:** 3.45–5  $\mu\text{m}$  (smaller pixels give more sampling but require more magnification; for 200  $\mu\text{m}$  you're fine with larger pixels).
- **Sensor format:** 1" or 2/3" gives comfortable FOV for 2 $\times$  objective.
- **Shutter:** **Global shutter** if the particles might be moving (avoids rolling skew).
- **Interface:** USB3 / USB3 Vision / GigE depending on throughput.
- **Resolution:** 2–8 MP is typical. Higher res gives larger FOV/resolution if needed.
- **Mono vs color:** **monochrome** preferred for sensitivity and SNR in darkfield.
- **Sensitivity:** low read noise, decent quantum efficiency in your chosen LED band.

Example (spec template):

- **Mono CMOS**, 1" sensor, 2048 $\times$ 1536 px, **3.45  $\mu\text{m}$**  pixels, global shutter, USB3, ~60–90 fps at full frame.

That camera with a 2 $\times$  **objective** yields ~6–10 mm FOV (width), and gives >50 px across a 200  $\mu\text{m}$  particle (excellent).

Example concrete design (one full configuration)

This is a practical, buildable configuration (no vendor links but you can match part names):

1. Objective
  - **Mitutoyo Plan Apo 2 $\times$  Infinity Corrected LWD** (or equivalent: Nikon/Edmund/Thorlabs 2 $\times$  LWD infinity objective)
  - Specs target: 2 $\times$ , NA  $\approx$  0.08–0.13, WD  $\geq$  20 mm, infinity-corrected, Plan
2. Tube Lens

- **200 mm tube lens** (match to your objective brand — most 2× infinity objectives are specified for 200 mm). Use a parfocal tube lens assembly designed for microscopy.
- 3. Beam Splitter
  - **Pellicle beamsplitter**, ~30R/70T (or **50/50 NPBS cube** as fallback). Size chosen to fit mechanical cube (e.g., 25 mm, 30 mm). Pellicle minimizes ghosting.
- 4. Illumination
  - **Annular LED**: bright monochromatic LED (green or white) fed through an annular aperture relayed to the objective BFP. Start with LED + condenser + annulus aperture mounted before the beamsplitter so reflected annulus goes into the objective. Alternatively, use a commercial epi-darkfield illuminator.
- 5. Camera
  - **1" monochrome sCMOS / industrial CMOS, 2048×1536, 3.45 μm pixels, global shutter**, USB3. Example spec to look for: global shutter, low read noise, >60% QE at chosen LED λ.
- 6. Mounting / optomechanics
  - 30 mm cage cube for beam splitter, adjustable lens tube to set the distance between objective and tube lens, camera C-mount adapter to connect tube lens to camera sensor.
- 7. **Optional**: Add a small aperture or field stop at a conjugate plane to improve background darkfield (blocks stray rays), and neutral density or adjustable LED driver to set intensity.

Numerical example — how many pixels and FOV you'll get (concrete)

Assume:

- Camera: 1" sensor, sensor width = **12.8 mm**, pixel size **3.45 μm**, resolution width = 3712 px (depends on actual sensor; adjust as needed).
- Objective: **2×**, tube lens 200 mm (matched).

Calculations:

- **Object FOV width** = sensor width / M = 12.8 mm / 2 = **6.4 mm**.
- **Pixels across particle (200 μm)** = (M × 200 μm) / 3.45 μm = (2 × 200) / 3.45 ≈ **116 px**.  
So a single 200 μm particle would occupy ~116 pixels across — excellent for detection and quantitative shape analysis.

If you choose **4×**, pixels across ≈ 232 px but FOV halves to ~3.2 mm width.

Practical tips & pitfalls

- **Check objective tube-lens compatibility.** Mismatched tube lens changes effective magnification and degrades correction. Always use tube lens recommended by the objective manufacturer.
- **Ghost reflections:** beam-splitter ghosting can brighten the background. Pellicle reduces this. Also use matte black interior surfaces and light traps.
- **Annulus placement:** the annulus must be conjugate to the objective back focal plane. If it's not conjugate the darkfield will be imperfect (bright background). Köhler illumination relays are the standard way to achieve the conjugation.
- **WD & sample geometry:** for 200  $\mu\text{m}$  particles (which may be protruding), a long working distance objective prevents collisions. If particles sit on a substrate, ensure you can focus through coverglass or air as needed.
- **Polarization effects:** avoid polarizing beam splitters unless you want polarization contrast; NPBS or pellicle are safer.
- **If you need motion capture:** choose camera with global shutter and sufficient frame rate to avoid motion blur.

If you want even more detail

I can:

- produce a **mechanical layout diagram** (optical axis, positions of annulus, beam splitter, tube lens, camera plane),
- calculate exact FOV & pixels for a particular camera model you give me,
- give a **shopping list** with specific vendor part numbers for objective, tube lens, pellicle, and cameras in different budgets (budget / mid / lab grade).

Tell me whether you prefer **largest FOV** (capture many particles at once) or **highest per-particle detail** (choose larger magnification) and whether your particles will be **static** or **moving**. If you already have a camera sensor spec or an objective brand in mind, give that and I'll lock the specs / part numbers instantly.

You said:

A Bunch of questions. how big is the spot size of a common CMOS. How big of an area is imaged here. How do i calculate the magnification and how do i determine if i can comfortably image a certain size of particle. Also what does infinity corrected objective lens mean. and what does objective lens do

ChatGPT said:

Thought for 7s

Great — lots packed into this question. I'll answer each part clearly, give the exact formulas you need, and show worked examples (including the 200  $\mu\text{m}$  particle you care about). No extra back-and-forth — I'll assume you want practical rules you can plug numbers into.

How big is the “spot size” of a common CMOS?

Two different things are commonly meant by “spot size”:

1. **Pixel size** — the physical size of one sensor pixel. this is the practical sampling unit. Common values:
  - o smartphone sensors: ~1.0–1.4  $\mu\text{m}$
  - o many industrial/CMOS cameras: **3.45  $\mu\text{m}$**  (very common), 5.0–6.0  $\mu\text{m}$ , 7.4  $\mu\text{m}$ , 9  $\mu\text{m}$ , etc.
  - o scientific CMOS (sCMOS): often 6.5  $\mu\text{m}$  or 11  $\mu\text{m}$  depending on sensor.

Use the pixel size as the *minimum sampling element* when you compute how many pixels an object will cover.

2. **Optical spot (PSF / Airy disk)** — the size of the image of a point source produced by the optics. This is set by diffraction and the objective's NA, not the sensor. In object space the approximate resolution (radius to first zero) is given by the Rayleigh / Airy formula:

$$\text{Airy disk diameter (object space)} \approx 1.22 \frac{\lambda}{\text{NA}}$$

or the more-used half-width (Rayleigh criterion)

$$d \approx 0.61 \frac{\lambda}{\text{NA}}.$$

Example: with  $\lambda = 550 \text{ nm}$  and  $\text{NA} = 0.1$ ,

$$d \approx 0.61 \times 0.55 \mu\text{m} / 0.1 \approx 3.36 \mu\text{m}.$$

That means the smallest resolvable feature in object space is  $\sim 3.4 \mu\text{m}$  for  $\text{NA}=0.1$ .

**Practical rule:** For microscopy imaging you need both the optical spot (resolution) and the pixel size to be considered. The pixel should sample the spot at about **2 pixels per resolution element (Nyquist)** or better.

How big an area is imaged (field of view)?

Field-of-view (FOV) on the *object* = (sensor physical size)  $\div$  (magnification).

Formulas:

- $\text{FOV\_width} = \text{sensor\_width} / M$
- $\text{FOV\_height} = \text{sensor\_height} / M$

Typical sensor physical sizes (approx):

- **1" sensor:**  $\sim 12.8 \text{ mm} \times 9.6 \text{ mm}$
- **2/3" sensor:**  $\sim 8.8 \text{ mm} \times 6.6 \text{ mm}$
- **APS-C (photographic):**  $\sim 23.6 \text{ mm} \times 15.6 \text{ mm}$

Examples (useful numbers for designs):

- With a 1" sensor and  $M = 2\times$ :  $FOV \approx 12.8/2 = 6.4 \text{ mm}$  wide.
- With same sensor and  $M = 4\times$ :  $FOV \approx 3.2 \text{ mm}$ .
- With  $M = 1\times$ :  $FOV = 12.8 \text{ mm}$ .

So decide how many millimeters you want to see across, then pick  $M = \text{sensor\_width} / \text{desired\_FOV}$ .

How do I calculate magnification?

There are two common cases:

A. Infinity-corrected objectives (the usual for modern microscopy)

- The objective has a stated magnification  $M_{\text{obj}}$  (e.g.,  $2\times$ ,  $10\times$ ). With the *manufacturer's recommended tube lens* (e.g., 200 mm for many objectives) the objective produces  $M_{\text{obj}}$  at the image plane.
- If you use a different tube lens focal length  $f_{\text{tube}}$  than the objective's design tube focal  $f_{\text{design}}$ , the effective magnification scales as:

$$M_{\text{effective}} = M_{\text{objective}} \times \frac{f_{\text{tube}}}{f_{\text{design}}}$$

*Example:* a  $10\times$  objective designed for 200 mm tube lens used with a 100 mm tube lens becomes  $10 \times (100/200) = 5\times$ .

B. Finite-conjugate objectives (older style, e.g., 160 mm tube length)

- These are designed to form an image at a fixed distance; use the objective's specified procedure. (Less common in modern modular microscope builds.)

**Camera adapter magnification:** If you put additional relay optics or C-mount adapters with focal length changes, include their magnification factor in the overall  $M$ .

**Summary:** For most builds with an infinity objective, effective microscope magnification at the sensor = the objective's stated magnification (provided you used the correct tube lens). If you change tube lens focal length, apply the simple scaling factor above.

How to determine whether you can comfortably image a particle of a given size

Two checks — sampling and optical resolution.

1. Sampling (pixels across the particle)

$$\text{pixels\_across} = \frac{M \times D_{\text{particle}}}{p}$$

where  $D_{\text{particle}}$  = particle diameter,  $p$  = pixel size,  $M$  = effective magnification.

Practical thresholds:

- Detection (just see it): ~6–10 pixels across
- Basic measurement / sizing: ~15–30 pixels across
- Shape/detail/edge analysis: 30+ pixels across

Example (your 200  $\mu\text{m}$  particle):

- Pixel = **3.45  $\mu\text{m}$** ,  $M = 1\times \rightarrow \text{pixels} = (1\times 200)/3.45 \approx \mathbf{58 \text{ px}}$  (excellent)
- Pixel = 3.45  $\mu\text{m}$ ,  $M = \mathbf{0.5\times} \rightarrow \approx \mathbf{29 \text{ px}}$  (still fine)
- Pixel = 5.86  $\mu\text{m}$ ,  $M = 1\times \rightarrow \approx \mathbf{34 \text{ px}}$  (good)  
So for a 200  $\mu\text{m}$  particle you do *not* need high magnification to get many pixels across it.

2. Optical resolution (can the optics actually resolve the particle features?)  
Compute the diffraction-limited resolution in object space:

$$d \approx 0.61 \frac{\lambda}{\text{NA}}$$

If the particle is much larger than  $d$ , the optics can resolve it. If the particle feature you want to measure is near or below  $d$ , you won't resolve it no matter how much you oversample with pixels.

Example: with  $\text{NA} = 0.1$  and  $\lambda = 550 \text{ nm}$ ,  $d \approx 3.4 \mu\text{m}$  — trivial compared to 200  $\mu\text{m}$ . So resolution is not the limiting factor for 200  $\mu\text{m}$  particles unless you need sub-10  $\mu\text{m}$  detail on them.

Combine both checks:

- Ensure pixels\_across meets your detection/measurement need.
- Ensure particle\_size  $\gg d$  (or at least larger than a few times  $d$ ) if you need resolved structure.

Worked examples for your 200  $\mu\text{m}$  particle (compact table)

Assume sensor pixel = 3.45  $\mu\text{m}$ , sensor width = 12.8 mm:

- $M = 1\times$ :
  - pixels across =  $(1\times 200)/3.45 \approx \mathbf{58 \text{ px}}$
  - FOV width = 12.8 mm  $\rightarrow$  can see **64** separate 200  $\mu\text{m}$  particles across if packed (12.8 mm / 0.2 mm = 64).
- $M = 2\times$ :
  - pixels across  $\approx \mathbf{116 \text{ px}}$
  - FOV = 6.4 mm wide.
- $M = 4\times$ :
  - pixels across  $\approx \mathbf{232 \text{ px}}$
  - FOV = 3.2 mm.

Optical resolution check ( $\lambda=550 \text{ nm}$ ):

- If objective NA = 0.08 →  $d \approx 0.61 \times 0.55 / 0.08 \approx 4.2 \text{ }\mu\text{m}$  (fine vs 200  $\mu\text{m}$ )
- If NA = 0.2 →  $d \approx 1.68 \text{ }\mu\text{m}$

All those NAs easily resolve a 200  $\mu\text{m}$  particle — resolution is not your limiting factor; sampling and FOV are.

What does “infinity-corrected objective” mean? And what does an objective lens do?

Infinity-corrected objective:

- In an infinity-corrected system the objective does not form a focused image by itself; instead it produces **collimated (parallel) light rays** for each point in the object (i.e., it sends the image to infinity). A separate **tube lens** downstream then takes those parallel rays and focuses them to form the intermediate image on the camera/sensor.
- Advantages:
  - Space between objective and tube lens where you can insert filters, beamsplitters, polarizers, apertures, annuli, or other optics **without altering imaging geometry**. (This is why epi-illumination and darkfield inserts are straightforward.)
  - Better correction of aberrations across different accessories and easier modular design.
- Most modern research objectives are infinity corrected; they will state the design tube lens focal length (commonly **200 mm** or **180 mm**).

What the objective lens does (summary of its roles):

1. **Primary magnification** — it sets the base magnification (e.g., 2×, 10×, 40×).
2. **Numerical aperture (NA)** — the main parameter that sets resolution and light-gathering ability. Higher NA → higher resolution and more light collected.
3. **Working distance (WD)** — distance from the front lens to the sample; important if particles protrude or you need space.
4. **Field flatness (Plan)** — many objectives are “Plan” corrected to produce a flat field so the whole image is in focus across the frame.
5. **Image formation** — for infinity objectives they produce collimated rays, then the tube lens forms the image at the camera plane. For finite objectives they form the image directly at a fixed mechanical image plane.

Quick checklist so you can pick components for your 200  $\mu\text{m}$  task

1. Decide how many particles / how much area you want per frame → pick sensor size and magnification so  $\text{FOV} = \text{sensor\_width} / M$  matches your needs.
2. Pick camera pixel size so that  $\text{pixels\_across} = M \times D_{\text{particle}} / \text{pixel\_size}$  meets your detection/measurement target ( $\geq 8$  px for detection,  $\geq 20$  px for good measurement).
3. Pick an objective with NA high enough for any small features you care about, but LWD if you need clearance. For 200  $\mu\text{m}$  particles a **2×–4× LWD objective, NA 0.08–0.2** is a practical sweet spot.

4. Use an infinity objective and the correct tube lens (usually 200 mm recommended) to keep magnification as specified and to allow the epi-darkfield optics.

If you want, give me:

- the exact camera model (or at least pixel size and sensor format), and
- whether you prefer **large FOV** (many particles per frame) or **high per-particle detail**,

and I'll compute exact FOV, pixels per particle, and recommend an objective/tube-lens pairing that matches.

### Appendix B.3

Given prompt to ChatGPT and Claude AI:

Write me a matlab code that performs paraxial raytracing with 7 Lenses. Lens 1 is AC254-030-A from Thorlabs, Lens 2 is AC254-030-A, Lens 3 is AC254-075-A and Lens 4 is AC254-080-A., Les 5 is AC254-100-A , Lens 6 is AC254-030-A and Lens 7 is A254-045-A. The distance between Lens 1 and the object is 30 mm. The distance between Lens 1 and 2 is 60 mm The distance between Lens 2 and 3 is 105 mm. The distance between lens 3 and 4 is 150 mm. The distance between lens 4 and 5 is 180 mm. The distance between lens 5 and 6 is 180 mm. The distance between lens 6 and 7 is 130 mm .The distance between lens 7 and image is 30 mm. The object height is 30 mm.

ChatGPT Output:

To perform paraxial ray tracing with the lenses you specified, we can calculate the overall system transfer function by sequentially considering the effects of refraction through each lens, as well as the propagation of the rays between the lenses. Each lens will have a focal length that you can find from the provided part numbers, and we'll use the lens formula to calculate the bending of rays at each lens.

Step-by-Step Plan:

1. **Lens Properties:** We will assume the focal lengths for each lens are available (you can look these up on Thorlabs' website or from the datasheet).
2. Ray Tracing Setup:
  - The object is placed at a distance of 30 mm from Lens 1.
  - For each subsequent lens, we will calculate the position of the rays after passing through the lens using the lens formula:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}$$

where:

- $f$  is the focal length of the lens.
- $d_o$  is the object distance.

- $d_i$  is the image distance.
3. Distances Between Lenses and Image Formation:
    - The distances between the lenses will affect the overall ray propagation. We will update the object distance for each subsequent lens based on the distance between the previous image and the next lens.
  4. Final Image Calculation:
    - After tracing through all lenses, we calculate the final image position and height.

MATLAB Code:

```
% Define the focal lengths (in mm) based on Thorlabs datasheets
f = [30, 30, 75, 80, 100, 30, 45]; % Focal lengths for each lens (in mm)

% Distances between lenses and object (in mm)
d_o = [30, 60, 105, 150, 180, 180, 130, 30]; % From object to Lens 1, between lenses,
and to image

% Object height (in mm)
object_height = 30; % mm

% Initialize variables to store ray properties
ray_position = zeros(1, 8); % Array to store ray position after each lens
ray_height = object_height; % Initial ray height

% Perform paraxial ray tracing
for i = 1:7
    % Calculate the object distance for this lens
    d_o_current = d_o(i);

    % Lens equation: 1/f = 1/d_o + 1/d_i -> solve for d_i
    % Rearranging to find image distance d_i
    d_i_current = 1 / (1 / f(i) - 1 / d_o_current);
```

```

    % Calculate the new ray height after passing through the lens (paraxial
approximation)
    % The height changes based on the magnification of the lens (which is d_i / d_o)
magnification = d_i_current / d_o_current;
ray_height = ray_height * magnification;

    % Store the ray position (image distance for this lens)
ray_position(i+1) = d_i_current; % Image position for this lens

    % Update the object distance for the next lens
if i < 7
    % The new object distance is the distance between the current image and the
next lens
    d_o(i+1) = d_o(i+1) - d_i_current; % Next object distance (image distance is
now object for next lens)
end
end

    % Final image position and height
final_image_position = ray_position(end); % Position of the final image
final_image_height = ray_height; % Final image height

    % Display results
fprintf('Final Image Position: %.2f mm\n', final_image_position);
fprintf('Final Image Height: %.2f mm\n', final_image_height);

```

Explanation of the Code:

1. **Focal Lengths:** The `f` array stores the focal lengths of each lens (you need to verify these based on the lens specifications for each part number from Thorlabs).
2. **Distances:** The array `d_o` stores the distances between the object and each lens, as well as the distances between lenses and the final image.
3. **Ray Height Calculation:** The height of the ray after passing through a lens is computed by multiplying the initial ray height by the magnification, which is the ratio of the image distance (`d_i`) to the object distance (`d_o`).

4. **Lens Equation:** The lens equation is used to calculate the image distance for each lens, based on the focal length and the object distance. The magnification at each step is used to update the ray height.
5. **Final Image:** After the last lens, the final image position and height are displayed.

Additional Notes:

- This code assumes the paraxial approximation (small angles) holds and neglects aberrations or higher-order effects.
- The focal lengths of each lens (provided in the f array) should be confirmed with the Thorlabs product datasheet. For example, the part number AC254-030-A corresponds to a lens with a focal length of 30 mm.

Let me know if you need help with anything else!

Claude AI Output:

I'll create a MATLAB code for paraxial raytracing through your 7-lens system using the Thorlabs achromatic doublet specifications.

Paraxial Raytracing - 7 Lens System

Code :

```
% Paraxial Raytracing through 7-Lens System
```

```
% Using Thorlabs Achromatic Doublets
```

```
clear; clc; close all;
```

```
%% Lens Specifications (Thorlabs AC254 Series)
```

```
% Focal lengths in mm at 587.6 nm
```

```
f1 = 30; % AC254-030-A
```

```
f2 = 30; % AC254-030-A
```

```
f3 = 75; % AC254-075-A
```

```
f4 = 80; % AC254-080-A
```

```
f5 = 100; % AC254-100-A
```

```
f6 = 30; % AC254-030-A
```

```
f7 = 45; % AC254-045-A
```

```
% Store in array for convenience
```

```
focal_lengths = [f1, f2, f3, f4, f5, f6, f7];
```

```

%% System Geometry
d_obj_L1 = 30; % Object to Lens 1
d_L1_L2 = 60; % Lens 1 to Lens 2
d_L2_L3 = 105; % Lens 2 to Lens 3
d_L3_L4 = 150; % Lens 3 to Lens 4
d_L4_L5 = 180; % Lens 4 to Lens 5
d_L5_L6 = 180; % Lens 5 to Lens 6
d_L6_L7 = 130; % Lens 6 to Lens 7
d_L7_img = 30; % Lens 7 to Image

% Store distances in array
distances = [d_obj_L1, d_L1_L2, d_L2_L3, d_L3_L4, d_L4_L5, d_L5_L6,
d_L6_L7, d_L7_img];

%% Object Parameters
h_obj = 30; % Object height in mm

%% Paraxial Raytracing using ABCD Matrix Method
% We'll trace rays from three object heights: 0, h_obj/2, and h_obj

num_rays = 5;
object_heights = linspace(0, h_obj, num_rays);

% Initialize storage for ray data
ray_positions = cell(num_rays, 1);
ray_angles = cell(num_rays, 1);
z_positions = cell(num_rays, 1);

fprintf('=== PARAXIAL RAYTRACING RESULTS ===\n\n');

```

```

for ray_idx = 1:num_rays
    y0 = object_heights(ray_idx); % Initial ray height
    theta0 = 0; % Parallel rays from object (at infinity approximation)

    % For finite object distance, calculate initial angle
    % Ray from object point through center of first lens
    theta0 = 0; % Chief ray approximation

    % Initialize ray vector [y; theta]
    ray = [y0; theta0];

    % Storage for this ray's path
    y_path = y0;
    theta_path = theta0;
    z_path = 0;

    current_z = 0;

    fprintf('Ray %d (Object height = %.2f mm):\n', ray_idx, y0);
    fprintf('Position: z = %.2f mm, y = %.4f mm, theta = %.6f rad\n', current_z, ray(1),
ray(2));

    % Propagate through the system
    for lens_idx = 1:7
        % Propagation to lens
        d = distances(lens_idx);
        T_prop = [1, d; 0, 1];
        ray = T_prop * ray;
        current_z = current_z + d;
    end
end

```

```

% Store position before lens
y_path = [y_path, ray(1)];
theta_path = [theta_path, ray(2)];
z_path = [z_path, current_z];

fprintf(' Before Lens %d: z = %.2f mm, y = %.4f mm, theta = %.6f rad\n', ...
        lens_idx, current_z, ray(1), ray(2));

% Thin lens
f = focal_lengths(lens_idx);
T_lens = [1, 0; -1/f, 1];
ray = T_lens * ray;

% Store position after lens (same z, different angle)
y_path = [y_path, ray(1)];
theta_path = [theta_path, ray(2)];
z_path = [z_path, current_z];

fprintf(' After Lens %d: z = %.2f mm, y = %.4f mm, theta = %.6f rad\n', ...
        lens_idx, current_z, ray(1), ray(2));
end

% Final propagation to image plane
d = distances(end);
T_prop = [1, d; 0, 1];
ray = T_prop * ray;
current_z = current_z + d;

y_path = [y_path, ray(1)];

```

```

theta_path = [theta_path, ray(2)];
z_path = [z_path, current_z];

fprintf(' At Image: z = %.2f mm, y = %.4f mm, theta = %.6f rad\n\n', ...
        current_z, ray(1), ray(2));

% Store for plotting
ray_positions{ray_idx} = y_path;
ray_angles{ray_idx} = theta_path;
z_positions{ray_idx} = z_path;
end

%% Calculate System Properties
fprintf('=== SYSTEM PROPERTIES ===\n');
fprintf('Total system length: %.2f mm\n', sum(distances));
fprintf('Number of lenses: 7\n');
fprintf('Object height: %.2f mm\n', h_obj);

% Calculate image height (from edge ray)
y_image = ray_positions{end}(end);
magnification = y_image / h_obj;
fprintf('Image height: %.4f mm\n', y_image);
fprintf('Lateral magnification: %.4f\n', magnification);

%% Plotting
figure('Position', [100, 100, 1200, 600]);

% Plot 1: Ray paths
subplot(2,1,1);
hold on; grid on;

```

```

% Plot rays
colors = jet(num_rays);
for ray_idx = 1:num_rays
    plot(z_positions{ray_idx}, ray_positions{ray_idx}, '-', ...
        'LineWidth', 1.5, 'Color', colors(ray_idx,:), ...
        'DisplayName', sprintf('h = %.1f mm', object_heights(ray_idx)));
end

% Plot lens positions
lens_positions = cumsum(distances(1:7));
y_limits = ylim;
for i = 1:7
    plot([lens_positions(i), lens_positions(i)], y_limits, 'k--', 'LineWidth', 1);
    text(lens_positions(i), y_limits(2)*0.95, sprintf('L%d', i), ...
        'HorizontalAlignment', 'center', 'FontWeight', 'bold');
end

% Mark object and image planes
plot([0, 0], y_limits, 'r-', 'LineWidth', 2);
text(0, y_limits(2)*0.95, 'Object', 'HorizontalAlignment', 'center', 'Color', 'r');

image_z = sum(distances);
plot([image_z, image_z], y_limits, 'b-', 'LineWidth', 2);
text(image_z, y_limits(2)*0.95, 'Image', 'HorizontalAlignment', 'center', 'Color', 'b');

xlabel('Axial Position z (mm)', 'FontSize', 12);
ylabel('Ray Height y (mm)', 'FontSize', 12);
title('Paraxial Ray Paths Through 7-Lens System', 'FontSize', 14, 'FontWeight',
'bold');

```

```

legend('Location', 'best');
hold off;

% Plot 2: Ray angles
subplot(2,1,2);
hold on; grid on;

for ray_idx = 1:num_rays
    plot(z_positions{ray_idx}, rad2deg(ray_angles{ray_idx}), '-', ...
        'LineWidth', 1.5, 'Color', colors(ray_idx,:), ...
        'DisplayName', sprintf('h = %.1f mm', object_heights(ray_idx)));
end

% Plot lens positions
y_limits_angle = ylim;
for i = 1:7
    plot([lens_positions(i), lens_positions(i)], y_limits_angle, 'k--', 'LineWidth', 1);
end

xlabel('Axial Position z (mm)', 'FontSize', 12);
ylabel('Ray Angle (degrees)', 'FontSize', 12);
title('Ray Angles Through System', 'FontSize', 14, 'FontWeight', 'bold');
legend('Location', 'best');
hold off;

%% Summary Table
fprintf('\n=== LENS SUMMARY ===\n');
fprintf('Lens | Model      | Focal Length (mm) | Position (mm)\n');
fprintf('-----|-----|-----|-----\n');
lens_models = {'AC254-030-A', 'AC254-030-A', 'AC254-075-A', 'AC254-080-A', ...

```

```

        'AC254-100-A', 'AC254-030-A', 'AC254-045-A'};
for i = 1:7
    fprintf(' %d | %-14s | %8.1f      | %8.2fn', ...
           i, lens_models{i}, focal_lengths(i), lens_positions(i));
end
fprintf('\n');

```

I've created a comprehensive MATLAB code for paraxial raytracing through your 7-lens system. Here's what the code does:

Features:

1. **Lens Specifications:** Uses the correct focal lengths for each Thorlabs achromatic doublet
2. **ABCD Matrix Method:** Implements paraxial raytracing using the standard matrix formulation
3. **Multiple Ray Tracing:** Traces 5 rays from different object heights (0 to 30 mm)
4. **Detailed Output:** Prints ray positions and angles at each lens interface
5. **System Analysis:** Calculates magnification and system properties
6. Visualization:
  - Ray paths through the entire system
  - Ray angles as they propagate
  - Lens positions marked with dashed lines
  - Object and image planes highlighted

Key System Parameters:

- Total system length: 855 mm
- Uses thin lens approximation (valid for paraxial rays)
- Traces rays through all 7 lenses with your specified spacing

The code will show you how rays from different object heights propagate through the system and where they intersect the image plane, allowing you to analyze the optical performance of your multi-lens system.

#### Appendix B.4 Case Study 4

**You said:**

3D printing fine parts

**ChatGPT said:**

3D printing fine or very small precision parts is definitely possible, but getting good results depends on the **printer, material, and print settings**. Here are the key points you should know:

## What Enables High-Precision 3D Printing

### 1. Printer Capability

For fine and detailed parts, the most important specs are:

- **Small nozzle size** (e.g., 0.2 mm or smaller)
- **High X/Y resolution** (accurate movement of print head)
- **Layer height capability down to 0.05 mm or lower**
- **Rigid frame and high-quality motion components**

Printers like the **Bambu Lab X1 series, Prusa MK4, or SLA printers** (e.g., Anycubic Mono, Formlabs) are especially good for this.

### 2. Printing Technology

Different technologies offer different precision:

#### FDM (Filament Printers)

- Good for strong functional parts.
- Precision limited by nozzle size.
- Great for small mechanical parts like gears, mounts, brackets.

#### SLA / Resin Printing

- **Best for ultra-fine miniatures and micro-features.**
- Extremely smooth surface detail (25–50  $\mu\text{m}$  layers).
- Great for jewelry, dental models, tiny gears, snap-fits.

### 3. Material Choice

Some materials print finer than others:

Material	Notes
----------	-------

PLA	Easiest, good for precision and stiff parts
ABS/ASA	Stronger but harder to fine tune
PETG	Good for strength, slightly less precise

Resin **Highest detail**, best for micro-components

### Recommended Print Settings for Fine Detail

If using FDM:

Setting	Recommended
Nozzle	0.2 mm
Layer Height	0.08–0.16 mm
Print Speed	Slow (20–60 mm/s)
Cooling	100% for PLA
Walls	2–3 per side

Flow Rate      Calibrated per filament

### Design Tips for Tiny Parts

- **Minimum wall thickness:**
  - FDM:  $\geq 0.4$  mm
  - SLA:  $\geq 0.2$  mm
- **Minimum hole diameter:**
  - FDM:  $\geq 0.8$ – $1.0$  mm
  - SLA:  $\geq 0.5$  mm
- **Avoid unsupported tiny overhangs**
- **Chamfer edges instead of sharp corners**
- **Add fillets to increase strength**

### **When You Need Extremely High Precision**

If tolerances below  $\pm 0.05$  mm are required, consider:

- **SLA resin printing**
- **Post-machining**
- **Sanding and finishing**

### **If You Want, I Can Also Help With:**

- Suggested printers for your budget
- Choosing the best material for your application
- Recommended slicer settings
- Fixing specific print quality issues
- Writing a polished report section explaining why a specific printer (e.g., Bambu X1) is ideal

### **You said:**

what are the best setting to change to print small and accurate parts

### **ChatGPT said:**

To print **small, accurate, and high-precision parts**, the biggest improvements come from tightening your slicer settings and slowing the print down. Here are the **most effective settings to change**, starting with the ones that make the biggest difference:

## Most Important Settings

### 1. Nozzle Size

- Use **0.2 mm nozzle** instead of 0.4 mm.
- Smaller nozzle = finer detail, better dimensional accuracy.

### 2. Layer Height

- Set **0.08–0.12 mm** layer height.
- Do **not** go lower unless your printer is extremely accurate—smaller layers increase time but not always quality.

### 3. Print Speed

Slower = more accurate.

#### Recommended:

- **Outer walls:** 20–35 mm/s
- **Inner walls:** 35–50 mm/s
- **Small parts mode:** ON (if your slicer has it)

Reducing speed dramatically improves sharp features.

### 4. Line Width

- Set line width close to nozzle size:
  - For 0.2 mm nozzle → **0.20–0.22 mm**
- Helps prevent over-extrusion and rounding of small features.

### 5. Temperature

Lowering temp slightly improves precision.

- Reduce nozzle temp **5–10°C** from your normal value.
- Too hot = smeared corners and fat lines.

## Dimensional Accuracy Settings

### 6. Flow Rate Calibration

- Over-extrusion is the #1 reason small parts print oversized.
- Calibrate flow/extrusion multiplier using a single-wall cube.

Usually 95–98% flow gives best dimensional accuracy.

## 7. XY Compensation / Horizontal Expansion

If holes print too small or outside dimensions too big:

- In slicer, adjust **Horizontal Expansion or XY Compensation**
  - Start with **-0.02 to -0.08 mm**
  - Test until dimensions match CAD

This setting alone can make prints dead-accurate.

**You said:**

**You said:**

sorry what is the best supports to use for quality 3d printing

**ChatGPT said:**

No problem — for **high-quality 3D prints**, the best support choice depends on the printer, material, and part geometry. Here are the **support types that consistently give the best results**, especially for clean surfaces and easy removal:

### **Best Overall Support Type**

#### **Tree Supports (Bambu, Cura, Orca)**

- Use less material
- Touch the model in fewer places
- Much easier to remove
- Leave **cleaner surfaces with less scarring**

**Best for:**

- Complex shapes
- Figurines
- Curved surfaces
- Overhangs without flat areas

## **Most Reliable and Precise**

### **Normal / Linear Supports (Classic Block Style)**

- Vertical “column” supports
- Very predictable
- Best for mechanical parts and flat overhangs

#### **Best for:**

- Brackets
- Flat faces
- Technical and functional parts

## **Best Balance of Quality + Easy Removal**

### **Snug / Paint-On Snug Supports**

(Available in Prusa, Orca, Cura)

These hug the model more tightly, supporting only where needed.

#### **Advantages**

- Cleaner underside quality than normal supports
- Easier to remove
- Less material

## **If You Need Minimal Marking**

### **Support Interface Layers**

Always enable:

- **Support Interface / “Support Roof”**
  - Flat, smooth layer between support and model
  - Dramatically improves underside quality
- **Support Gap / Z-Distance**
  - PLA: 0.2 mm
  - ABS/PETG: 0.25–0.3 mm

This makes supports peel off cleanly without tearing the surface.

## Best Support Pattern

The pattern matters too:

Pattern	Best For	Notes
<b>Lines / Zig-Zag</b>	Easy removal	Most common, good surface finish
<b>Grid</b>	Strong support	Harder to remove
<b>Triangles</b>	Strength	Use only when needed

**Concentric** Ultra-smooth contact Best surface finish for curved parts

For most quality prints:  
**Lines / Zig-Zag + Interface layers** is the cleanest combo.

## Crucial Settings for High Surface Quality

### 1 Support Overhang Angle

- Default: 45°
- For cleaner parts: **50–55°**
- Reduces unnecessary supports

### 2 Support Density

- **10–20%** is usually enough

### 3 Support Only Touching Build Plate

- Turns off supports inside the model unless necessary
- Reduces scarring

### 4 Brim Not on Supports

If printing small support structures, brims can help stability.

## Best Combinations by Use Case

### For Detailed Miniatures

- **Tree supports**
- **Lines pattern**
- **Interface layers**
- **50–55° angle**

### For Mechanical Parts

- **Normal supports**
- **Lines or Grid**
- **Interface roof**
- **0.2 mm Z gap**

### For Minimal Surface Damage

- **Snug + Concentric top**
- **Or Tree supports**

### For Fast Printing

- **Tree supports with low density**

## Quick Recommended Setup (PLA)

- Support type: **Tree or Snug**
- Pattern: **Lines**
- Support angle: **50°**
- Support roof/interface: **ON**
- Z gap: **0.20 mm**
- Density: **12–18%**

If you tell me:

- Printer (Bambu, Prusa, Creality, etc.)
- Material
- Part type (mechanical, cosmetic, small, etc.)

I can give you **dialed-in support settings specifically for your machine.**

**You said:**

amount of supports is not of concern would making it like 30 insure that there is no chance of issues in the print

**ChatGPT said:**

Short answer: **Yes, setting the overhang support angle to 30° will almost guarantee full support coverage and eliminate most print failures — but it will also create a lot of unnecessary supports and can actually reduce print quality in some cases.**

Let me explain clearly:

**What a 30° Support Angle Means**

- Supports will be generated for **any surface flatter than 30° from vertical**, which is extremely aggressive.
- Almost any slope or shallow surface will get supported.
- This is much more than typical settings (45–50°).

So yes:

**It will minimize the chance of sagging or drooping**

**It will eliminate most overhang-related failures**

**But Here's the Problem**

Using a 30° threshold can **hurt print quality overall**:

**1. More supports touching the model**

More contact surfaces =

- More scarring
- More sanding afterward
- Worse cosmetic finish

**2. Harder support removal**

Especially in:

- Small parts
- Narrow internal areas

- PETG and ABS (which bond tightly)

### **3. Longer print time and much more filament used**

### **4. Increased risk of breaking small features**

Removing tight supports can chip edges or snap fine details.

## **Important Real-World Behavior**

Even with “unlimited supports”:

- **Supports themselves can transfer heat into the underside, sometimes causing roughness.**
- **Too much support doesn’t always improve surface quality.**
- **Perfect support removal becomes harder the more supports you create.**

In many cases, a **50° angle with a support interface layer gives a cleaner bottom surface than 30° with no interface.**

## **If Your Goal Is:**

### **“Zero chance of overhang sagging”**

- **30° will do it**
- But at the cost of surface finish

### **“Highest possible surface quality”**

Use instead:

- **50° overhang angle**
- **Support interface (roof) ON**
- **Z gap: 0.20 mm (PLA)**
- **Tree or snug supports**

This produces:

- Fewer marks
- Cleaner bottom surfaces
- Less post-processing
- Still reliable prints

## Best Setting Strategy

If strength and perfection matter more than filament use:

**35–40°**

- Very safe
- Less overkill
- Better surface finish than 30°

**30°**

- Maximum safety
- Worst for removal marks
- Only recommended for:
  - Complex curved shapes
  - Resin-like detail
  - Tiny unsupported features
  - Critical prints where failure is unacceptable

## Final Recommendation

If you do **not care about filament or time**, but you **want the highest surface quality and zero risk**, then:

Instead of lowering overhang angle to 30°, keep angle at **45–50°** and enable:

- **Support Interface Layers**
- **Line or Concentric interface pattern**
- **0.20 mm Z-gap**
- **Slow outer wall speeds (20–35 mm/s)**

This gives better results than flooding the model with supports.

**If you tell me:**

- Your printer
- Material
- The type of part (mechanical? cosmetic? tiny features?)

...I can tell you **the exact support profile setup that guarantees best results for your specific machine.**

**You said:**

so would around 40 degrees be good to insure that the structure is printed properly?

**ChatGPT said:**

Yes — **around 40° is an excellent choice if your goal is maximum print reliability without going overboard on supports.**

Here's why:

### **Why 40° Works Well**

- It is **more conservative than 45°**, meaning supports will appear earlier.
- It **greatly reduces the chance of sagging or curling**, especially on flat underside features.
- It **doesn't create the extreme over-supporting** that happens at 30°.
- Most printers (especially good ones like Bambu, Prusa, etc.) handle 40° very well.

So:

**Much safer than 50–55°**

**Less messy and easier to remove than 30°**

**Very reliable even on deep overhangs**

### **What You Can Expect at 40°**

- Overhang failures are unlikely.
- Bottom surfaces will be cleaner and flatter.
- Support removal is manageable.
- Print time and filament use increase slightly, but not drastically.

This makes **40° a very reasonable “high-safety, high-success” setting.**

## When 40° Is Better Than 45–50°

Use 40° if:

- You're printing **certain success, no-fail parts**
- The print has **large unsupported undersides**
- You're using materials that sag easier (PETG, ABS)
- Cooling isn't perfect

## If You Want Maximum Reliability

Best setup:

- **Support angle: 40°**
- **Support interface layers: ON**
- **Interface thickness: 0.6–1.0 mm**
- **Z separation (air gap):**
  - PLA: **0.20 mm**
  - PETG/ABS: **0.25–0.30 mm**
- **Support pattern:** Lines or Zig-Zag
- **Outer wall speed:** 25–35 mm/s

This combo gives:

- **Excellent bottom surface finish**
- **Easy removal**
- **Strong support contact**
- **Minimal failure risk**

## Final Answer

**Yes — 40° is a very good minimum overhang angle**

It provides a **strong balance of safety, quality, and clean results**, and is a preferred setting for many users who want reliable prints without goes-crazy amounts of support.

If you want, tell me:

- Printer model

- Material
- Geometry (mechanical part? mini? curved surface?)

I can confirm whether 40° is optimal for **your exact setup**.

**You said:**

are there any other things that i should consider

**ChatGPT said:**

Yes — besides the overhang angle, there are **several other settings and factors that strongly affect support performance, surface quality, and overall print success**. These are the ones most people overlook but make a big difference:

### 1. Use Support Interface (Support Roof)

This is **one of the biggest quality upgrades**.

**Why:**

- Creates a smooth buffer layer between the support and the model.
- Prevents the support pattern from imprinting into the part.
- Makes removal easier.

**Recommended:**

- **Enabled**
- Interface thickness: **0.6–1.0 mm**
- Interface density: **50–80%**

Even with many supports, this **protects the print surface**.

### 2. Z Support Gap (Vertical Separation)

The “air gap” between supports and the model determines:

- Surface cleanliness
- Ease of removal

**Recommended Values:**

Material	Z Gap

PLA	<b>0.20 mm</b>
PETG	<b>0.25–0.30 mm</b>
ABS	<b>0.25–0.30 mm</b>

Resin N/A (different process)

If the gap is too small:

- Supports weld to the part → rough removal

If too big:

- Overhang may sag

### 3. Support Pattern

Pattern affects removal and quality:

Pattern	Best For
<b>Lines / Zig-Zag</b>	Best mix of quality + easy removal
<b>Concentric (at interface)</b>	Extremely good underside finish

**Grid/Triangular**

Strong but harder to remove

For high-quality bottom surfaces:

Lines + Interface → best consistent results

Concentric interface → best underside finish

### 4. Support Density

Support density doesn't need to be high.

**Recommended:**

- **10–20%** for most cases
- **25–40%** only for very heavy overhangs

Higher density rarely improves quality unless the part is heavy.

## **5. Cooling is Critical**

Supports won't save you if the overhang is melting.

**PLA:**

- **100% fan**
- Slower speed on external walls (20–35 mm/s)

**PETG:**

- Use **less fan (20–40%)** or it will string.

**ABS:**

- Minimal fan to prevent warping.

## **6. Part Orientation**

Before adding supports, consider rotating the model.

Even a **10–20° rotation** can:

- Remove the need for supports entirely
- Reduce surface scarring
- Improve strength and quality

A well-oriented part needs fewer supports and prints better.

## **7. Wall & Shell Settings**

A thicker outer wall helps edges stay straight above supports.

**Recommended:**

- **2–3 walls minimum**

- External perimeter printed **last**

This gives cleaner edges at transition zones.

## 8. Slow Down for Overhangs

Even with supports, printing too fast can cause melting or drooping.

### Recommended:

- Outer wall speed: **20–35 mm/s**
- Enable **slow down for small layers**

FDM accuracy heavily improves when slowed down.

## 9. Nozzle Temperature

If temperature is too high, the plastic will sag even with supports.

### Rule of thumb:

Lower temperature **5–10°C** when printing fine detail.

## 10. Support Type

Tree or Snug supports:

- Use less material
- Touch the model in fewer places
- Leave cleaner surfaces

Normal linear supports are better for strictly mechanical parts.

## 11. Bridging Settings

For overhangs between supports:

Enable:

- **Bridge fan override**
- **Bridge speed limits**

- **Bridge flow reduction (~80–95%)**

This keeps bridges tight and prevents droop.

### **TL;DR – If You Want The Best Quality**

If your goal is high-quality, reliable printing, here is the **ideal support combo**:

- **Overhang angle: 40–50°**
- **Support interface ON**
- **Z-gap: 0.20 mm (PLA)**
- **Support pattern: Lines**
- **Outer wall speed: 20–35 mm/s**
- **Cooling tuned for material**
- **Temperature lowered 5–10°C**
- **3 walls for cleaner overhang edges**

This setup produces:

- Clean undersides
- Easy removal
- High reliability
- Minimal support scars

If you tell me:

- Your printer
- Material
- What kind of part you're printing

I can give you a **dialed-in, specific support profile for your exact machine and job.**

**You said:**

the printer is a Bambu labs x1 series

**ChatGPT said:**

Great — since you're using a **Bambu Lab X1 series**, you have one of the best consumer printers for overhang performance. Still, if your goal is **maximum print reliability and dimensional accuracy**, here are the key additional settings and considerations you should look at beyond just support angle.

## Settings to Consider for the Highest Print Reliability

### 1 Support Z-Distance (Support Gap)

Even if supports are plentiful, you need them to **detach cleanly without damaging the print.**

For the X1, recommended:

- **PLA** / **PETG** / **ABS:**  
**0.20 mm**
- **Precision** parts with tiny features:  
**0.15 mm**

If the Z-distance is too small, supports weld to the part. If too large, the overhang sags.

### 2 Support Interface Layers (Highly Recommended)

Turn on:

- **Support Roof**
- **Support Floor (if needed)**

These create a tight, smooth layer between support and part, improving surface finish dramatically.

For best performance:

- **Support Roof layers: 2–3**
- **Support Roof density: 70–90%**

### 3 Cooling Matters More Than Angle

Even with supports, cooling affects overhang quality:

**Best for X1:**

- **Fan Speed: 80–100% for PLA**
- **70–90% for ABS/ASA**
- If using PETG, don't go above **40–60%** or it may get brittle.

Better cooling means cleaner undersides.

#### **4 Layer Height for Better Overhangs**

Lower layer heights handle overhangs far better.

Recommended:

- **0.12–0.16 mm for fine or precision parts**
- **0.08 mm for extreme detail**

Thinner layers sag less over open space.

#### **5 Reduce Print Speed on Small Parts**

X1 is fast, but small features benefit from slowing down.

- **Walls: 80–120 mm/s**
- **Small perimeters: 40–60 mm/s**
- **Features under 10 mm: Enable “Slow down small areas”**

This reduces ringing, overshoot, and melt deformation.

#### **6 Support Pattern Selection**

Since filament use isn't a concern:

**Best for quality:**

- **Tree (Organic)** – leaves the cleanest contact surfaces
- **Gyroid** – strongest, easy to remove
- **Rectilinear** – consistent, predictable, good for tight spaces

Avoid Grid unless strength is needed — it's hard to remove.

#### **7 Use “Support on Build Plate Only” When Possible**

If part geometry allows, this keeps supports away from finished surfaces — best possible quality.

#### **8 Overhang Threshold Recommendation (for X1)**

Because the X1 has:

- Strong cooling
- Extremely accurate motion
- Stable extrusion

It can handle:

- **40° with 100% reliability**
- **30° is extra safe**, but adds unnecessary supports

For tight tolerances and zero risk:

- **35–40° is perfect**

### ★ Best “Ultra-Safe” + High Quality Settings (X1)

If you want **zero risk**:

- Overhang threshold: **35–40°**
- Layer height: **0.12–0.16 mm**
- Support Z gap: **0.15–0.20 mm**
- Support roof: **ON**, 3 layers, 80% density
- Fan speed: **90–100% (PLA)**
- Slower small features enabled
- Use Tree or Gyroid supports

This will give extremely reliable prints with smooth undersides and minimal risk of sagging.

**If you want, tell me:**

- Material (PLA, ABS, PETG, Nylon, CF, etc.)
- Size of the part
- Whether it’s cosmetic or mechanical

I can give you a **perfect tuned profile specifically for the X1**.

**You said:**

what are the minimum size object that should be able to be printed with these settings

**ChatGPT said:**

With a **Bambu Lab X1 series** printer and the settings we discussed (low layer height, 0.2 mm nozzle, Tree/Gyroid supports, 35–40° overhang threshold, proper cooling), the

minimum printable object size depends on both **feature thickness** and **overall dimensions**. Here's a detailed breakdown:

◆ **Minimum Feature Sizes**

<b>Feature Type</b>	<b>Recommended Minimum</b>
<b>Wall / Shell Thickness</b>	<b>0.4 mm</b> (for FDM with 0.2 mm nozzle)
<b>Vertical hole diameter</b>	<b>0.8–1.0 mm</b> (smaller holes may fuse closed)
<b>Horizontal hole/slot width</b>	<b>1.0–1.2 mm</b> (depends on orientation)
<b>Small pillars / posts</b>	<b>0.6–0.8 mm diameter</b> (any smaller may wobble or break)
<b>Bridges</b>	5–8 mm span max for unsupported bridges at these layer heights

**Overhangs** Can reliably handle **35–40° angles**, even on very small features

Note: SLA/resin printers can go much smaller, but for FDM on the X1, these are practical limits.

◆ **Minimum Overall Part Size**

- **Height:** ~1 cm is practical (smaller than that and the part may warp during printing)
- **Length/Width:** ~5 mm minimum for a single stable feature
- For very small parts (<1 cm), consider **solid infill** or slightly thicker walls to avoid breaking during handling

◆ **Tips for Small Parts**

1. **Print slower** — 20–40 mm/s for small features.
2. **Use the finest layer height** — 0.08–0.12 mm for micro features.
3. **Enable support on small features** — Tree supports are ideal.
4. **Avoid extremely thin spikes or pillars** — they will fail even with support.
5. **Orient parts for strength** — avoid tall thin posts standing vertically without support.

◆ **Practical Example**

For a small mechanical part (like a gear or bracket):

- Wall thickness: 0.6 mm
- Hole diameter: 1 mm
- Smallest feature: 0.8 mm
- Height: 1–2 cm
- Can be reliably printed with the settings we discussed.

If you want, I can make a **quick reference table specifically for Bambu X1 with PLA**, showing **the absolute minimum dimensions for different feature types** to ensure they print reliably at 0.2 mm nozzle and 0.12–0.16 mm layer height.

Do you want me to make that table?