

Low-Cost Fiber Extrusion Device for Educational Purposes: Redesign, Manufacture, and Computer Vision Integration

By

Gary Sefah

Bachelor of Science in Mechanical Engineering

University of Oklahoma, 2022

Submitted to the Department of Mechanical Engineering

in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN ADVANCED MANUFACTURING AND DESIGN

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

© 2023 Gary Sefah. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Author: Gary Sefah

Department of Mechanical Engineering

August 11, 2023

Certified by: Brian W. Anthony

Principal Research Scientist, Thesis Supervisor

Accepted by: Nicolas Hadjiconstantinou

Professor of Mechanical Engineering

Chairman, Department Committee on Graduate Theses

This page intentionally left blank.

Low-Cost Fiber Extrusion Device for Educational Purposes: Redesign, Manufacture, and Computer Vision Integration

by

Gary Sefah

Submitted to the Department of Mechanical Engineering on August 11, 2022, in partial

fulfillment of the requirements for the degree of Master of Engineering in Advanced

Manufacturing and Design

Abstract

The Fiber Extrusion Device (FrED) serves as a hands-on learning tool and laboratory experience, simulating the continuous fiber draw process to provide insights into data acquisition, control systems, and smart manufacturing. This system enables learners to conduct experiments, manipulate manufacturing parameters and control systems, gather data, and conduct analyses. While successful classroom activities have been conducted using FrED, the preceding model's cost precludes widespread distribution for remote learning, a growing trend in education.

This thesis encompasses a series of enhancements to FrED, aimed at refining its stability, cooling mechanisms, modularity, noise reduction, size, and overall functionality. Pulley variations were introduced to enhance fiber stability. Cooling strategies and pulley system's flexibility were optimized for the stability of the fiber, and noise reduction measures focused on the gear system. The camera system underwent significant redesigning, enabling more precise fiber diameter measurement. In addition to that, a shift from Teensy to Raspberry Pi improved system integration. Code for extrusion and gear motors, heater, and thermistor was rewritten, alongside redesigns of the extrusion system, PCB, and camera module.

The final FrED design accomplished a 42% cost reduction (\$159) and a weight reduction of 25% (1.7 kg) with optimal fiber cooling and stability, seamless integration of computer vision for diameter measurement and data collection was achieved, enabling its application in PID control and enhancing the teaching of machine learning principles.

Acknowledgement

I would firstly like to thank God for his unwavering grace and for a successful completion of my master's program. I extend my profound appreciation to my family, whose endless love, unwavering encouragement, and prayers have been my constant source of strength. My father, Ebenezer Sefah Snr., brother and wife, Ebenezer Sefah Jnr and Abigail Sefah, Sister, Joycelyn Sefah, and the Anderson family, to whom I am forever indebted. My achievements thus far are attributed to your steadfast support.

I would also like to thank my thesis advisor, Professor Brian Anthony, for the invaluable guidance and constructive feedback during weekly meetings and design reviews. Professor Anthony's feedback and ideas lead to several big and revolutionary enhancements to the design, imparting me with invaluable lessons and perspectives that will undoubtedly be applied to my future mechanical design career.

I am also grateful to Dr. David E. Hardt and Mr. Jose J. Pacheco for entrusting me with the opportunity to embark on this project. I would also like to thank Samantha Young for her consistent support throughout the project. Implementing this project without her timely support would not have been possible.

I would like to say the biggest thank you to the FrED project's summer team of 2023, Wenhao Xu and Somesh Jaiswal, who have been awesome teammates and a source of inspiration and joy throughout the Spring and Summer of 2022. I learned a lot from collaborating and enjoyed working with them. Our collaborative efforts in 35-017 will forever be remembered as a time of laughter, learning and fun.

I extend my heartfelt gratitude also to my friends from the MEng 2023 cohort for being great companions throughout the one-year period at MIT- Wenhao Xu, Vineeth Gowra, Gazi Sakib, Somesh Jaiswal, Keeghan Patrick, Pierre Lonni, Yutong Zhang, Hadeel Abdo, Cheng Chang, John Waterworth, Jonathan Sampson, Sophia Chueng and James Liggett. Thank you to MEng 2023 cohort for giving me a family I could rely on throughout the master's program. I would also like to thank Russel Bradley for being a great friend and a guide and support for my cohort and I, and for making our transition and masters journey seamless.

I would like to also thank all my friends from Ghana, UWC Changshu China, and University of Oklahoma who have been a steadfast source of support and encouragement during my master's program.

Lastly, I dedicate this whole masters journey and my success to my late mother, Margaret Sefah, and my late brother Samuel Sefah. I made it to the dream school and made it through successfully. It is my fondest wish to share these triumphs with you both, with boundless happiness.

In closing, this journey has been one of personal and academic growth. To all those mentioned and to the countless others whose contributions helped shape my path, I offer my deepest gratitude. May the future be as promising and fulfilling as this chapter has been.

Table of Contents

List of Figures	8
List of Tables	15
1. Introduction	16
1.1 Background	16
1.2 Product Overview.....	17
1.3 Project Overview.....	21
1.4 Methodology overview	22
1.5 Stakeholders	23
1.6 Thesis Outline	24
2. Enhancing the Cooling System:	25
2.1 Previous Design.....	25
2.2 Issues	27
2.3 Design Changes.....	28
2.3.1 Axial facing front at a 75-degree angle:	28
2.3.2 Axial facing front at an angle and hood:.....	29
2.3.3 Using the Peltier and heat sink system:	31
2.3.4 Using centrifugal fans:.....	33
2.3.5 Made a modular frame for the centrifugal fan:.....	34
2.3.6 Made it with a hood:.....	38
2.3.7 Used the 3D printed square washer to identify the right fan frame length:.....	40
3. Improving Fiber Stability and Uniformity:	43
3.1 Previous Design:	43
3.2 Issues:	45
3.3 Design Changes:.....	46
3.3.1 Change it to pulley support that can be moved up and down:.....	46
3.3.2 Tried different PLA printed pulleys with different groves:	48
3.3.3 Plastic pulleys:.....	49
3.3.4 Metal pulleys:	50
3.3.5 Support with the ball design:	52
3.3.6 Alternating the fiber direction (one behind-two in front), shortening one support, and making it one-sided:	54
3.3.7 Fully modulated pulley system to go up and down:.....	56

4. Refining the Gear System:	58
4.1 Previous Design:	58
4.2 Issues:	59
4.3 Design Changes:.....	60
4.3.1 Different materials used to reduce noise and vibration:	60
4.3.2. Moving the gear box inwards to have less tension on the rubber:	61
4.3.3 Changing hanging motor to DC motor:	61
5. Integrating Computer Vision Technology:	63
5.1 Laser Micrometer:	63
5.2 Low cost FrED diameter fiber reading set up:	67
5.3 Image processing for pictures:	74
5.4 Image processing for recorded video:	81
5.5 Image processing for live video:	86
5.6 Image processing on Raspberry Pi Camera model 3:	88
6. Implementing Hardware Improvements:	98
6.1 Extrusion System:	98
6.1.1 Previous Design:	98
6.1.2 Design Changes:.....	101
6.2 The Frame:	108
6.2.1 Previous Design:.....	108
6.2.2 Design changes:	110
7. Upgrading the Micro processing System:	113
7.1 Teensy:.....	113
7.2 Raspberry Pi:	114
7.3 Raspberry Pi Installation:	116
7.4 Raspberry Pi and PCB integration	119
7.5 Translation of code to Raspberry Pi:	122
7.6 PCB remodeling:	122
8. Cost Analysis:	125
8.1 Old Design:	125
8.2 New FrED design:.....	126
8.2.1 Cost analysis of the fiber extrusion subcomponents:	126

8.2.2 Cost analysis of the cooling subcomponents:.....	127
8.2.3 Cost analysis of the diameter measurement subcomponents:	128
8.2.4 Cost analysis of the fiber collection subcomponents:	129
8.2.5 Cost analysis of the frame subcomponents:	130
8.2.6 Cost analysis of the electronics subcomponents:	131
8.2.7 Cost analysis of the overall FrED system changes:.....	132
9. Future Works	133
10. Reference	136
11. Appendix	139

List of Figures

Figure 1. This is the initial researched-based version of FrED developed by D. D. Kim, et al. ..	16
Figure 2. This is the four main subcomponents of the FrED system.....	18
Figure 3. This is the complete assembly of the low-cost FrED system that was achieved by the MEng team of 2022.	20
Figure 4. This is the inventory space for the FrED factory showing some of the component used to make the extrusion sub system of FrED.	21
Figure 5. This is the 4010 12V DC axial fan which is was used in the cooling system for the old version of the low-cost FrED system.....	26
Figure 6. This is the fan that is mounted on the frame angles at 45 degrees to help cooling efficiency of the fiber.	27
Figure 7. This is the axial fan and its frame mounted to the FrED system to face the fiber from the front to cool the fiber as its being extruded.	27
Figure 8. This is the axial fan mounted to the frame angled at 75 degrees to help have better contact along the length of the fiber.....	29
Figure 9. This is a 75-degree angled fan, and its frame is mounted on the FrED system to enhance cooling of the fiber.....	29
Figure 10. This is the hood that is attached to the axial fan to help concentrate the cooled are directly to the fiber for efficient cooling.....	30
Figure 11. This shows the orientation of hood on the axial fan frame to help it direct the cooled air to the fiber.....	30
Figure 12. This is the working principle of the Peltier board.	31
Figure 13. This is the individual components that make up the system, which include the Peltier board, heat sink, and the axial fan.....	32
Figure 14. This shows the assembly of the Peltier-heat sink system with axial fans to help have an efficient transfer of cooled air on the fiber.....	32
Figure 15. This is the DC 12V 0.10A~0.15A 50mmx15mm centrifugal fan that will aid concentrate the fiber to the fiber.	33
Figure 16. This shows a right sided cross-sectional view of the frame design to allow it to rotate at an angle possible after being 3D printed.....	35
Figure 17. This shows the M3 screw that is used to secure the angled fan after it has been adjusted.	35
Figure 18. This is the design of the frame that will be used to help mount the centrifugal fan to the FrED system.....	36

Figure 19. This is the centrifugal fan with its frame mounted to the FrED system close to the extrusion subsystem to allow early cooling of the fiber right from extrusion. 37

Figure 20. This is to show the wide range of orientation of the fan frame can be angled to aid efficient cooling and concentration on the fiber. 37

Figure 21. This is the fan frame with hood to help prolonged concentration of the cooled on the fiber length. 39

Figure 22. This is the fan frame with the hood mounted to the FrED system to enhance cooling of the fiber. 39

Figure 23. This are the square washers that were used to help figure the right length of the fan frame to allow the air to continually target the fiber during the drawing process. 40

Figure 24. This shows how the washers were used to find the right length that the frames needed to be to continually target cooled air to the fiber. 41

Figure 25. This is an example of how the length of the frame (without the hood) change after determining the right legth for the frames to enhance it cooling capabilities. 42

Figure 26. Thus is the configuration of the pulleys for the old version of the FrED system. 44

Figure 27. This is the 3D printed pulleys that was used for guiding the fiber during the drawing process in the old version of the FrED system. 44

Figure 28. This shows the sticky nature of the fiber on the pulley as well as the vibrations that the fiber experiences during the drawing process. 45

Figure 29. This shows the residual fiber left behind when it sticks on the pulley. 46

Figure 30. This is the pulley support that can be moved anywhere along the frame of the FrED system. 47

Figure 31. This is the pulley with little bumps int eh internal groove to reduce stickiness of the fiber. 48

Figure 32. This the two aerosol sprays used to reduce the stickiness between the 3D printed pulleys and the fiber. 49

Figure 33. This is the different plastic pulleys that were used to enhance stability of the fiber. .. 50

Figure 34. This is the different metal pulleys that were used to enhance stability of the fiber. 51

Figure 35. This shows the fiber very stable with the eye when the steel pulley was being implemented. 52

Figure 36. This shows an example of the gap with the pulley and the pulley support. 53

Figure 37. This is an example of the pulley when it tightly connected to the pulley support. 53

Figure 38. This is the design of the pulley support with the ball (left side) and also shows it being connected to the pulley (right side)..... 54

Figure 39. This shows a 0.3 mm stable fiber with the naked eye. 55

Figure 40. This shows the instability of the fiber when looked through the microscope lens causing the diameter to increase to 0.6 mm.	55
Figure 41. This shows the path of the fiber to achieve optimal stability for both the eye and lens of the microscope.	56
Figure 42. This shows the range at which the pulleys can be modulated.	57
Figure 43. This shows the modulation of the gear system with the ball bevel gears.	58
Figure 44. This is the gear system with the gear box cover on top of it.	59
Figure 45. This shows the different dampening alternatives that were used to reduce the noise and vibrations.	60
Figure 46. This shows different mount iterations and the final design with the DC motor connected to it.	62
Figure 47. This shows the position of the laser micrometer on the research based FrED.	64
Figure 48. This shows the IG-028 T and the IG-028 R connected to the 3D printed support for proper alignment of the two component for proper reading.	64
Figure 49. This shows the IG-1000 amplifier with its connection to the receiver (red strip) and the receiver (blue strip).	65
Figure 50. Schematic of the wires for the amplifier with the brown and blue wires necessary for powering the laser micrometer.	65
Figure 51. This shows the other wires that were insulated by tape to avoid any shocks when current runs through the amplifier.	66
Figure 52. This the settings process to access the measurement mode.	66
Figure 53. This shows the final display on the amplifier screen to be in the right mode for diameter measurement.	67
Figure 54. This shows the protrusion of the laser micrometer to detect to provide diameter readings on the amplifier.	67
Figure 55. This is the Jiuson 40-100x digital microscope used on the low-cost FrED system. ...	68
Figure 56. This is the cooling setup used for complete solidification of the fiber before the drawing process.	69
Figure 57. This is the three-pulley system that aids in the guidance and stability of the fiber during the drawing process.	70
Figure 58. This shows the different background that was used as background for the microscope.	71
Figure 59. This shows the different preforms that were used for the fiber production.	71
Figure 60. This shows the set up for attaining images with eh groove of the pulley as a potential background for image processing.	72

Figure 61. This shows the fiber in the middle and the multiple machine lines which makes it difficult to implement edge detection on the fiber.	72
Figure 62. This is the images captured by the microscope when the dark back screen is close to the fiber showing more internal structures of the board and making edge detection difficult.	73
Figure 63. This is the images captured by the microscope when the dark back screen placed out of focus to the fiber to see the edges of the fiber better.	73
Figure 64. This shows the whole camera setup of the microscope to aid in the quality images for image processing.....	74
Figure 65. This shows the line of code to import the libraries needed for the computer vision for pictures.....	75
Figure 66. This shows the line of code for mounting the google drive and displaying the images captured by the microscope.	76
Figure 67. This shows the lines of code used to implement the gray scale, canny edge and the hough lines functions.	77
Figure 68. This shows the line of codes that is used to detect the lines on the original image. ...	78
Figure 69. This shows the line of code to calculate the diameter with and print out the value....	78
Figure 70. This is the line of code to tackle scenarios where the image being process is tilted..	79
Figure 71. This shows the code for the lines to attain the two lines for the fiber.	79
Figure 72. This is the final display of the image after the code has been executed to find the edges of the fiber and print the output value.....	80
Figure 73. This shows the edge case where the groove of the pulley is used as a back screen and the image is tilted.....	80
Figure 74. This is the lines of code for mounting, retrieving frames and then initializing variables.....	81
Figure 75. This the initial lines of codes int eh while loop to get the lines for the edges of the fiber with gray scale, canny detection and hough lines.	82
Figure 76. This is the line of code to filter and detect the two lines that are the edges of the fiber.	83
Figure 77. This is the line of code to display, calculate the width of the fiber and print out the output value.....	84
Figure 78. This is the line of code to display the video properties.	84
Figure 79. This is the final display of the image after the code has been executed to find the edges of the fiber and print the output value.....	85
Figure 80. This shows the output of an edges case of when the video is too blurry, the fiber shaking too much or when the back screen shows multiple lines.....	86

Figure 81. This is the line of code help provide the serial connection between the Teensy Arduino and the device being used.	87
Figure 82. This is the line of code that calculates and sends the values to the Teensy Arduino in real time.	87
Figure 83. This is the Raspberry Pi Camera 3 SC0872 11.9 MP that is used in image processing to attain 120 fps.....	90
Figure 84. This is the White LED Backlight Module that is used to provide contrast with the fiber for better image quality.....	91
Figure 85. This is the first prototype to house and attach the Raspberry pi to the FrED system.	92
Figure 86. This is the second prototype to house and attach the Raspberry pi to the FrED system.	92
Figure 87. This shows protrusion that serves as a protective buffer, shielding the camera's sensitive elements from any pressure or damage.....	93
Figure 88. This shows the setup of the camera system for the Raspberry Pi camera module 3 for image processing.....	93
Figure 89. This is the line of code to import the library, make a picamera2 instance and setting the window to display processing images.....	94
Figure 90. This is the line of code to capture the frames and employ grayscale, canny edge detection and hough line algorithm on the frames.....	95
Figure 91. This is the line of codes that draw the lines detected from the frame and cluster lines to make two distinct lines.	96
Figure 92. This is the line of code to calculate and print the fiber diameter as an output.	97
Figure 93. This is the heating area for melting the preform.	99
Figure 94. This is extrusion system support of the old version of the FrED system holding the stepper motor.....	100
95. This is the whole assembly of the extrusion system of the old version of the FrED system.	100
Figure 96. This shows the whole setup of the extrusion system for the old version.	101
Figure 97. This shows the lever for the old extrusion system vs the lever of the new extrusion system with the protrusion to hold the perform.	102
Figure 98. This shows the spring from the previous design and the new stronger spring to aid in alignment and prevent slippage of the preform.	102
Figure 99. Cross sectional view of the lock nut showing the serrations and the nylon insert (blue piece).....	103
Figure 100. This is the low-strength steel nylon-insert locknut that was used to ensures that the spring remains in place for proper alignment of the preform.	104

Figure 101. This is the PC alloy heater block that is used in place of the heat sink to avoid hazardous scenarios and reduce cost.....	105
Figure 102. This shows the leakages of the preform when the heat sink is in use.	106
Figure 103. This is the excess mounting plate that support the stepper motor that was eliminated.	106
Figure 104. This shows the other excess blocks and features that are reduced and eliminated.	107
Figure 105. This the redesign of the extrusion system support.	107
Figure 106. This shows the difficult in attaching the nut when fastening the screw.	108
Figure 107. This shows the extrusion system with the new support and the easiness in attaching the nut when fastening the screw.	108
Figure 108. This is the full assembly of the fame of the FrED system.....	109
Figure 109. This shows the T-slot system to lock the back and rib, and then to the base of the frame.	110
Figure 110. This shows the different rubbers that were used to aid in elasticity and vibration isolation in the FrED system.....	112
Figure 111. This shows the size of the old version of the FrED frame (right frame) and the new version with reduced size (left frame).	112
Figure 112. This is the Teensy Arduino that is used in the FrED system for deep reinforce learning control.	113
Figure 113. This is the schematic of the Teensy Arduino I/O.....	114
Figure 114. This is the 4 Model B, 8GB of RAM Raspberry Pi with a 128GB SD card for the integration of the running the FrED system and image processing.	115
Figure 115. This is the schematic of the GPIO pins of the Raspberry Pi.	116
Figure 116. This shows the use of the SD formatter to clean out the SD card.	117
Figure 117. This shows using the Raspberry Pi Imager to download the requisite operating system on the SD card.....	118
Figure 118. This shows the use of the jumper wire used for the Raspberry Pi and PCB integration.	119
Figure 119. This is the Adafruit ADS1115 chip that is used as the analog to digital converter for the thermistor.	121
Figure 120. This is the schematic of the Adafruit ADS1115 chip.....	122
Figure 121. This is the old version of the PCB used for testing of the new component and Raspberry Pi.....	123
Figure 122. This is the new PCB that will be used for the new low-cost FrED system.	124

Figure 123. This is a rendered picture of the current low cost FrED system. 133
Figure 124. This is the actual image of the new low cost FrED system. 134

List of Tables

Table 1. This is the different cameras and the important properties consideration for image processing of the fiber.....	89
Table 2. This shows the GPIO ports from Figure 114 that are used to connect to the PCB ports for output for the seamless integration of the FrED component.....	120
Table 3. This shows the number of parts used in each sub part and the cost for each of them. .	125
Table 4. This is the bill of material for the fiber extrusion subsystem for the new version of the low cost FrED system.	126
Table 5. This is the bill of material for the cooling subsystem for the new version of the low cost FrED system.....	127
Table 6. This is the bill of material for the diameter measurement subsystem for the new version of the low cost FrED system.	128
Table 7. This is the bill of material for the fiber collection subsystem for the new version of the low cost FrED system.	129
Table 8. This is the bill of material for the frame subsystem for the new version of the low cost FrED system.....	130
Table 9. This is the bill of material for the electronics subsystem for the new version of the low cost FrED system.	131
Table 10. This shows the number of parts used in each sub part and the cost for each of them.	132

1. Introduction

This chapter introduces the background and overview of the Desktop Fiber Extrusion Device, (FrED). It presents a comprehensive product overview, project scope, timeline, and stakeholders involved. The methodology employed in the research is briefly outlined, highlighting the chosen techniques and their rationale. The chapter also acknowledges the team's collective efforts and contributions. Finally, a concise thesis outline is provided, offering a glimpse of the forthcoming chapters' content and structure.

1.1 Background

Optical fiber manufacturing forms the cornerstone of modern internet communication technology. This intricate process involves fiber extrusion, a continuous manufacturing method, where various parameters are crucial in achieving precise dimensions, particularly the fiber's diameter. Taking inspiration from desktop 3D printers, a pioneering effort by Kim et al [1]. led to the development of a revolutionary desktop fiber extrusion device known as FrED. The primary objective behind FrED's creation was to offer students an experiential understanding of manufacturing and feedback control systems in a hands-on and heuristic manner. Through FrED, students gained the ability to manipulate process parameters, thereby varying the diameter of the extruded fiber.

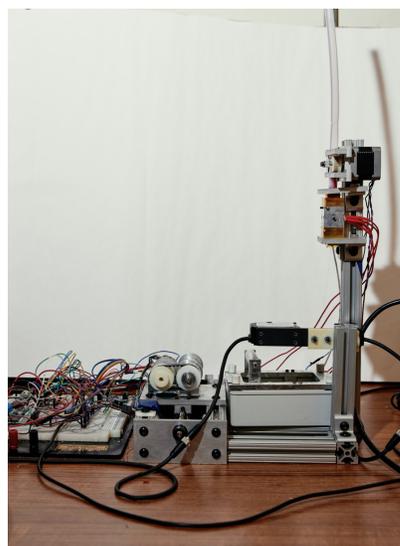


Figure 1. This is the initial researched-based version of FrED developed by D. D. Kim, et al.

The success of initial prototypes at MIT prompted a collaboration with Monterrey Tech, Mexico, to introduce FrEDs in their manufacturing courses, extending its reach to a wider audience. As the demand for a cost-effective FrED variant grew, an ambitious endeavor was undertaken to establish an assembly line at MIT for large-scale production. This not only addressed the need for affordability but also created an opportunity to impart manufacturing knowledge to students at MIT, particularly those pursuing the Master of Engineering in Advanced Manufacturing and Design. Consequently, this collaboration with Tec de Monterrey aimed to introduce a similar course centered around FrED in their curriculum, thereby facilitating a comprehensive and globally relevant manufacturing education program. The ultimate goal was to democratize access to FrED and foster a new generation of skilled professionals driving the advancement of optical fiber manufacturing and related technologies.

As part of a larger endeavor to enhance the learning experience for online students, the development of a cost-effective FrED variant became crucial. This involved reimagining the design of the device hardware and utilizing affordable materials, all while reducing manufacturing complexity and streamlining production and assembly processes to improve functionality. Additionally, an important focus was on incorporating advanced sensors to enable efficient and precise data collection [1]. Notably, computer vision technology was integrated into the system to facilitate diameter measurements with unprecedented accuracy. By harnessing these technological advancements, the new iteration of FrED aimed to provide a more comprehensive and immersive learning experience for students, further empowering them to explore the intricacies of fiber extrusion and manufacturing control systems.

1.2 Product Overview

Fiber Extrusion Device (FrED) represents a smart desktop fiber extrusion system developed with a primary focus on educational applications. Originally created in 2017 by David Kim and Brian Anthony, FrED was specifically designed for professional education at MIT [1][2]. Subsequently, in 2021, Cuiffi et al. expanded its utility to train members of the manufacturing workforce. Distinguishing itself from large and expensive industrial setups, FrED was intentionally engineered to be compact, safe, and cost-effective, while also providing students with ample opportunities to explore the realms of smart manufacturing and feedback control systems [2]. The

utilization of glue sticks as the preform material, heated, and pulled to create fibers with desired diameters, offers a practical and secure learning experience. The incorporation of various sensors ensures an abundance of data for data analytics and process control. Beyond its pedagogical purpose, FrED serves as a valuable tool for new fiber design research and small-batch fiber prototyping, supporting the exploration of different materials and extrusion diameters.

FrED's core components encompass the extruder, cooling system, spool, and sensors, collectively forming a comprehensive fiber extrusion setup [1].

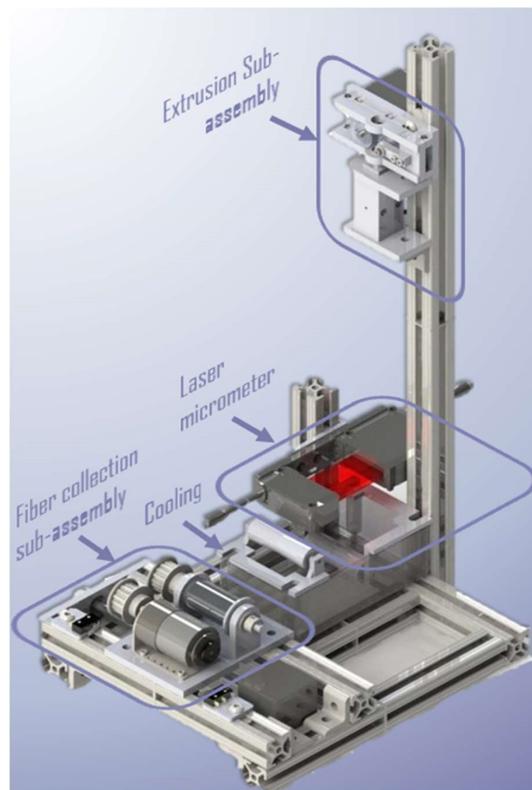


Figure 2. This is the four main subcomponents of the FrED system.

The extruder employs stepper motors as actuators to drive heated glue sticks through a narrower exit, controlling the extrusion rate via stepper motor velocity [1]. Subsequently, the fiber proceeds to the cooling system, comprised of a cooling tank with water as the cooling agent, preventing adhesion to other FrED components. The fiber is then pulled and collected by the spool, employing

a DC motor for rotation and a stepper motor in conjunction with a lead screw to ensure even fiber collection, thus regulating the fiber diameter.

Key to the successful fiber diameter control is the integration of three essential sensors: the resistance temperature detector (RTD), placed within the extruder's heating chamber for precise temperature measurement and extrusion control [1][3]. The laser micrometer gauges the fiber diameter before entering the cooling system, supplying vital data for the closed-loop feedback control system, an integral aspect of fiber diameter regulation. Additionally, a pair of limit switches are incorporated to prevent the fiber from straying off the spool, though they do not contribute to diameter control data. Initially, a simple mass conservation model was used to control fiber diameter based on extrusion speed and spool dimensions, but this empirical approach alone proved insufficient due to temperature inconsistencies and stage motion time-mismatching [5]. Consequently, closed-loop feedback control was introduced, employing the proportional controller (P-controller) and the proportional integral controller (PI-controller), respectively, to take spool rotational velocity and the error between measured and target diameters as control feedback [4]. This control system effectively ensures precise fiber diameter regulation. Furthermore, a noteworthy advancement in control methodology is the utilization of deep reinforcement learning (DRL) as a numerical model-free control system [1][5]. DRL has demonstrated improved performance in tracking diameter error, and ongoing research seeks to enhance its effectiveness through data pre-training to reduce online training time.

The FrED project was initiated with the successful development of prototypes from the research point of view fulfilling the required technical specifications. However, the cost of producing and deploying these prototypes on a larger scale posed a significant challenge. Therefore, the primary goal of the project was to drastically reduce the cost of FrED from its original \$5428 to less than \$200 per unit, while simultaneously establishing a scalable manufacturing process [6][8]. The key methodologies to overcome this challenge included redesigning FrED using design for manufacturing and assembly principles, building functional prototypes, defining a process plan for large-scale production, setting up a supply chain, and establishing an assembly line for pilot production [7]. The project successfully achieved its goals by the end of Summer 2022.

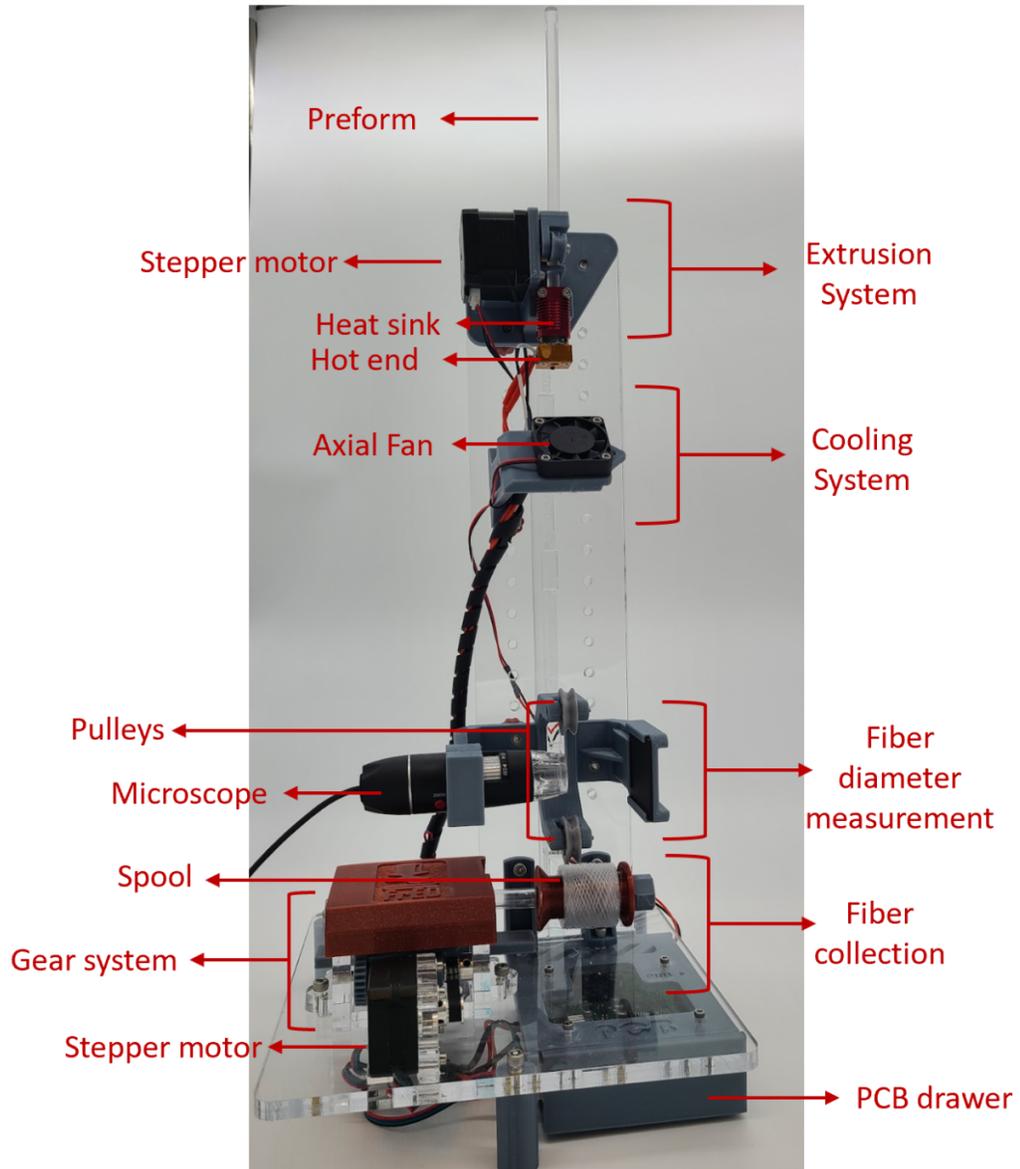


Figure 3. This is the complete assembly of the low-cost FrED system that was achieved by the MEng team of 2022.

FrED's applications expanded to smart manufacturing education, and it found practical use in various manufacturing courses at MIT and Tec de Monterrey [7]. Throughout the design process, the principles of Design for Manufacturing and Assembly (DFMA) guided decisions, focusing on standard parts, reduced components, modularity, and integration. The project demonstrated

significant advancements in affordable and scalable fiber extrusion technology as well as a pilot production of 25 FrED units was set as the target to be completed within a 6-month timeframe [9].



Figure 4. This is the inventory space for the FrED factory showing some of the component used to make the extrusion sub system of FrED.

1.3 Project Overview

The initial FrED prototype demonstrated technical success. Subsequently, achieving the milestone of reducing the system cost from \$5428 to approximately **\$270** per unit and enabling large-scale manufacturing was also accomplished. However, this project aims to further enhance the FrED system by significantly reducing the production cost to less than \$200 per unit while introducing substantial improvements to its functionality. To have better understanding of the project goal, the FrED device was assembled from scratch, and operated to observe the current issued with the latest design. With the help of Russell Bradley, who was part of the team with the latest design, these were the general enhancements that were discussed, as outlined below:

- Enhance the cooling system to ensure efficient cooling of the preform from the extruder to a solid state to improve the drawing process of the fiber.

- Improve the stability and uniformity of the FrED-produced fiber to enhance the accuracy of diameter measurements.
- Optimize the gear system to reduce noise and eliminate vibrations that contribute to fiber instability.
- Integrate cutting-edge computer vision technology for precise measurement of the fiber's diameter.
- Implement comprehensive hardware improvements to enhance functionality and reduce costs.
- Upgrade the micro processing system of FrED to provide users with a better experience, comprehensive control capabilities, and advanced image processing functionalities.

1.4 Methodology overview

The redesign and manufacturing of FrED followed a systematic product development life cycle, comprising the following steps:

- Initial identification of functional requirements for the system.
- Detailed analysis of the existing system to identify components or aspects that fell short of meeting the specified requirements.
- Comprehensive material research and exploration of potential redesign options to enhance system functionality.
- Creation of prototypes based on the proposed improvements, followed by rigorous testing to evaluate their performance against the specified requirements.
- Continuous iteration through the design, prototyping, and testing phases to refine and optimize the system until the most suitable solution for meeting the functional requirements was achieved.

Throughout the design process, the principles of DFMA (Design for Manufacturing and Assembly) were adhered to [20], with a focus on the following aspects:

- Utilization of standard parts whenever possible to simplify sourcing and assembly.

- Streamlining the total number of parts while ensuring that all functional requirements were met.
- Optimizing hardware designs for enhanced deployability and seamless integration within diverse systems and scenarios, catering to educational requisites, facilitation of repairs, as well as preemptive provisioning for component substitution.
- Integration of parts wherever feasible to enhance the overall efficiency and reliability of the system.

1.5 Stakeholders

The development of FrED involved a diverse group of key stakeholders, each playing a crucial role in the project's success. At the heart of the initiative was the FrED Factory Founding Team, consisting of four dedicated Master of Engineering in Advanced Manufacturing and Design students. This team took on the primary responsibility of designing, developing, and manufacturing the low-cost FrED, making it all possible [6][7][8][9]. Guiding and supporting them was Brian Anthony, PhD, an esteemed MIT Faculty specializing in data science and smart manufacturing. Brian, as the Principal Investigator for the FrED initiative, provided essential advice and mentorship.

The Master of Engineering in Advanced Manufacturing and Design Office, led by Professor David Hardt, Jose Pacheco, MBA, and Brian Anthony, PhD, played a pivotal role in shaping the vision for the FrED ecosystem. Their dedication to transforming on-campus manufacturing education was a driving force behind the project's goals. Additionally, faculties from Tec de Monterrey, Erick Guadalupe Ramirez-Cedillo and Adriana Vargas-Martinez, served as important customers for the project. Their involvement in parallel development and utilization of FrED for teaching activities added valuable insights.

MIT Professional Education (PE) also played a significant role as a provider of online and on-campus professional courses in Smart Manufacturing. Their vision for FrED as an educational kit that could be dispatched to learners enrolled in their classes expanded the project's reach and impact. Lastly, the MIT Mechanical Engineering Department provided a platform for students to learn manufacturing through programs like the Undergraduate Research Opportunities Program

(UROP). FrED Factory's collaboration with the department enabled aspiring engineers to gain valuable hands-on learning experiences.

Through collaboration, expertise, and dedication, these stakeholders collectively contributed to the successful development and implementation of FrED. Their combined efforts made a meaningful impact in the field of advanced manufacturing and design education, providing a transformative learning experience for students and professionals alike.

1.6 Thesis Outline

The following sections aim to focus details of enhancing the functionality and cost-effectiveness of the FrED (Fiber Extrusion Device) through a systematic product development approach addressing the areas of optimizing the cooling system for efficient preform cooling, improving fiber stability for precise diameter measurements, optimizing the gear system to reduce noise and vibrations, integrating computer vision technology for accurate diameter measurement, implementing hardware improvements to enhance functionality and reduce costs, and upgrading the microprocessor system for better user experience and control capabilities. The methodology involves rigorous research, iterative design, prototyping, and testing to achieve the best functional solution. This will also include a comprehensive analysis of the proposed improvements, their effectiveness, contributions, implications, and future directions for further advancements.

2. Enhancing the Cooling System:

Ensuring that the preform of the fiber is fully cooled before moving to the drawing process is of paramount importance in the production of fibers, as this critical step in the manufacturing process offers several key benefits. The drawing process involves the elongating preform moving through a series of pulleys enabling meticulous control, ensuring uniformity and optimal properties throughout the fiber. [20].

First and foremost, proper cooling of the preform is essential for achieving high-quality and uniform fibers. By allowing the preform material to solidify evenly, it helps prevent the formation of internal defects and inconsistencies that could adversely affect the fiber's properties. Additionally, adequate cooling enhances the mechanical strength and integrity of the resulting fiber, ensuring that the drawn fiber possesses the necessary tensile strength and resistance to bending and tearing during the drawing process. Moreover, the cooling process plays a crucial role in controlling the diameter of the fiber. The rate of cooling can directly influence the final fiber diameter, which is crucial for maintaining consistent properties.

In addition to improving fiber quality, cooling the preform gradually and uniformly reduces the residual stress and strain within the material. This reduction in internal stresses is critical to prevent micro-cracks and fiber breakage during the drawing process and subsequent handling. Proper cooling also helps to avoid fiber deformation and distortion. If the preform is not adequately cooled, it may deform during the drawing process, leading to irregular shapes and undesirable.

2.1 Previous Design

In the previous iteration of the low-cost Fiber Extrusion Device (FrED), the cooling method employed a 4010 12V DC axial fan which directed airflow downward toward the fiber. The axial fan has a wider area at which the airflow disperses which ensures a large area coverage when cooling.



Figure 5. This is the 4010 12V DC axial fan which is was used in the cooling system for the old version of the low-cost FrED system.

The fan was also strategically positioned directly beneath the extrusion subassembly to expedite the cooling process as soon as the preform was introduced into the system. This immediate cooling upon the preform's descent optimizes the fiber's thermal properties and contributes to enhanced overall efficiency. Furthermore, the axial fan was oriented to face the front of the fiber. By the face-front position and directing the airflow toward the fiber's frontal surface, the cooling efficiency is enhanced as it targets the region where heat accumulation is most prominent during extrusion.

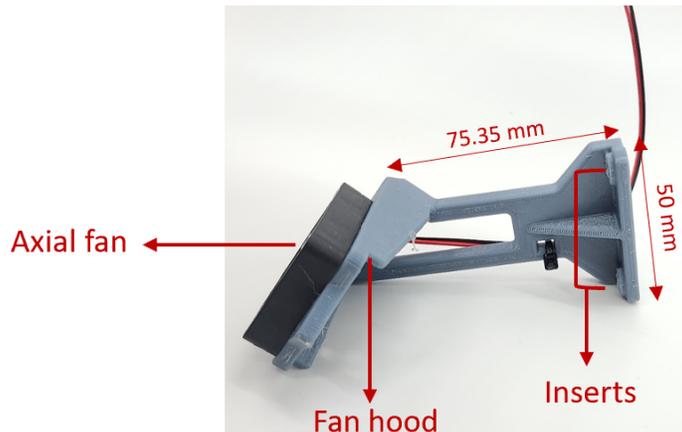


Figure 6. This is the fan that is mounted on the frame angles at 45 degrees to help cooling efficiency of the fiber.

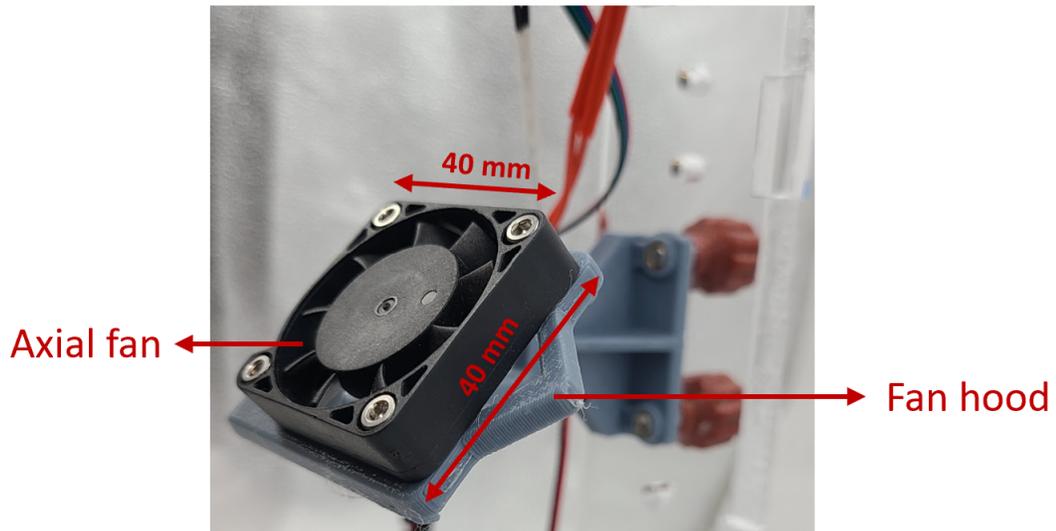


Figure 7. This is the axial fan and its frame mounted to the FrED system to face the fiber from the front to cool the fiber as its being extruded.

The design of the fan support structure also played a crucial role in maximizing cooling efficiency. The fan support was intentionally crafted to maintain an angle with respect to the fiber's axis. This inclination of the fan ensured that the cooling airflow was directed at an oblique angle, rather than purely vertical. The incorporation of an angled fan facilitates increased contact between the cooling airflow and the entire length of the fiber, surpassing what would be achieved with a purely vertical orientation. Moreover, this advantageous angle effectively guides most of the airflow downward, contributing to more efficient cooling, while simultaneously mitigating the incidence of airflow hitting the back of the device and diverting it away from the intended directional flow of the fiber.

2.2 Issues

The inadequacy of the previous cooling system is evident in its inability to effectively cool the fiber, resulting in issues such as fiber sticking on the pulleys and susceptibility to breakage during

high-speed spooling at the bottom of the FrED system. After in-depth analysis of the root cause of these cooling inefficiencies it was realized that despite the implementation of an axial fan with a wide spread of cooling air, the inherent thinness of the fiber poses a significant challenge. The fiber's diameter, being approximately 100 times smaller than the diameter of the area from which the cooling air emanates, leads to considerable air wastage. As a consequence, only around 5% of the cooling air is effectively directed to the fiber itself, with the remaining 95% dissipating into the surrounding environment. This imbalance in air distribution significantly compromises the cooling process and results in inadequate heat dissipation, hindering the fiber's solidification before descending through the FrED system. In addition to that, the suboptimal speed of the fan further exacerbates the cooling inefficiency. A low fan speed inherently translates to a reduced airflow rate, resulting in a diminished amount of cooling air reaching the fiber as it progresses downwards. The insufficient flow of cooling air over the fiber's surface leads to prolonged exposure to high temperatures, preventing the fiber from attaining the required solidification state within the desired timeframe. Consequently, the fiber remains vulnerable to the issues of sticking and breakage during the spooling process.

2.3 Design Changes

2.3.1 Axial facing front at a 75-degree angle:

The initial iteration involved employing the original design with a slight increase in the angle of tilt from 45 degrees to 75 degrees. This adjustment aimed to enhance the cooling process by enabling the cooled air to cover a greater length of the fiber, resulting in a more rapid cooling rate compared to the previous design. However, despite this modification, a persistent challenge persisted wherein a significant portion of the air dissipated into the surrounding environment rather than directly cooling the fiber. Consequently, even with the tilted angle, achieving complete cooling of the fiber remained difficult.

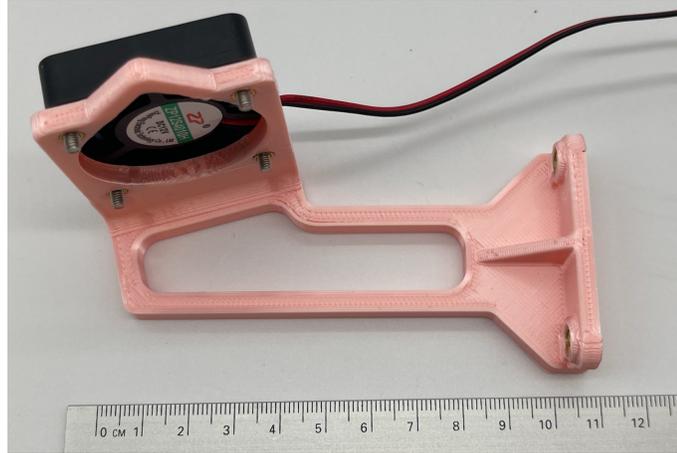


Figure 8. This is the axial fan mounted to the frame angled at 75 degrees to help have better contact along the length of the fiber.



Figure 9. This is a 75-degree angled fan, and its frame is mounted on the FrED system to enhance cooling of the fiber.

2.3.2 Axial facing front at an angle and hood:

The subsequent design iteration was to address the cooling inefficiencies encountered in the previous design. This incorporates a designed hood around the axial fan, which strategically directed the cooled air exclusively towards the fiber, rather than allowing it to disperse into the surrounding environment. This deliberate redirection of the airflow was intended to increase the

cooling effectiveness by concentrating the cooling stream directly onto the fiber surface. Although the airflow velocity aimed at the fiber was elevated, it did not yet attain the desired level necessary for optimal cooling prior to the fiber drawing process.



Figure 10. This is the hood that is attached to the axial fan to help concentrate the cooled air directly to the fiber for efficient cooling.

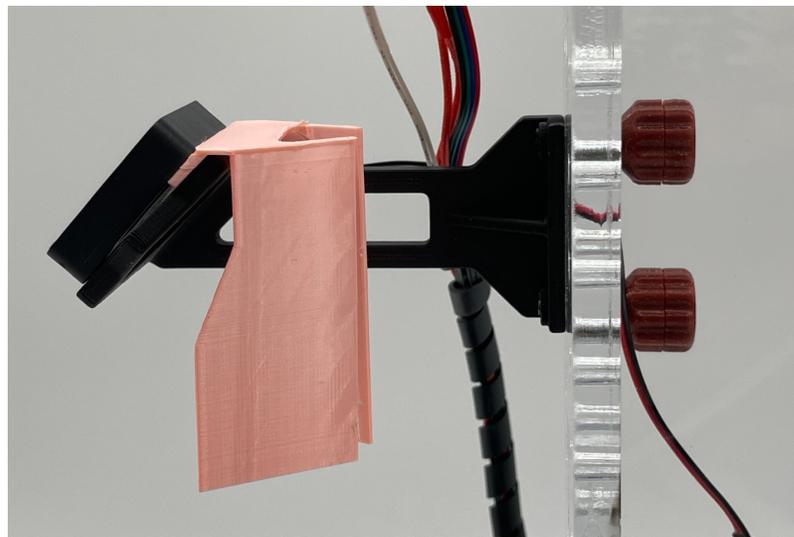


Figure 11. This shows the orientation of hood on the axial fan frame to help it direct the cooled air to the fiber.

2.3.3 Using the Peltier and heat sink system:

The subsequent design phase involved integrating a heat sink cooling system with the incorporation of a Peltier board to further enhance the cooling efficiency directed towards the fiber. The Peltier board, known for its exceptional ability to achieve extremely low temperatures through the phenomenon of thermoelectric cooling, played a pivotal role in this solution which is based on the principle of the Peltier effect, can create a substantial temperature differential across its surface, with one side becoming significantly cold while the opposite side generates heat [10].

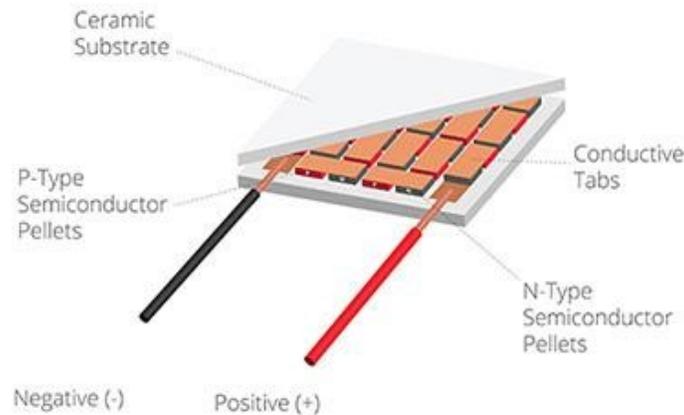


Figure 12. This is the working principle of the Peltier board.

The design involved positioning the Peltier material between two heat sinks and corresponding fans. The primary objective of the first heat sink was to transfer the cold region generated by the Peltier board to the fan responsible for directing cooled air onto the fiber. This approach aimed to maximize the cooling potential and facilitate the delivery of colder air to the fiber surface. Conversely, the second heat sink was situated on the hot region of the Peltier board, where excess heat accumulated during the cooling process. An additional fan, connected to this heat sink, was tasked with dissipating the accumulated heat away from the system, ensuring that the air blowing towards the fiber would consistently remain cold.

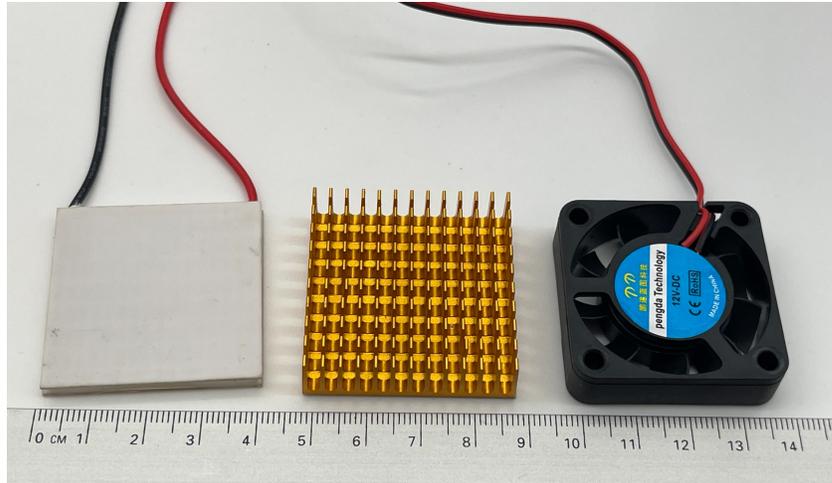


Figure 13. This is the individual components that make up the system, which include the Peltier board, heat sink, and the axial fan.

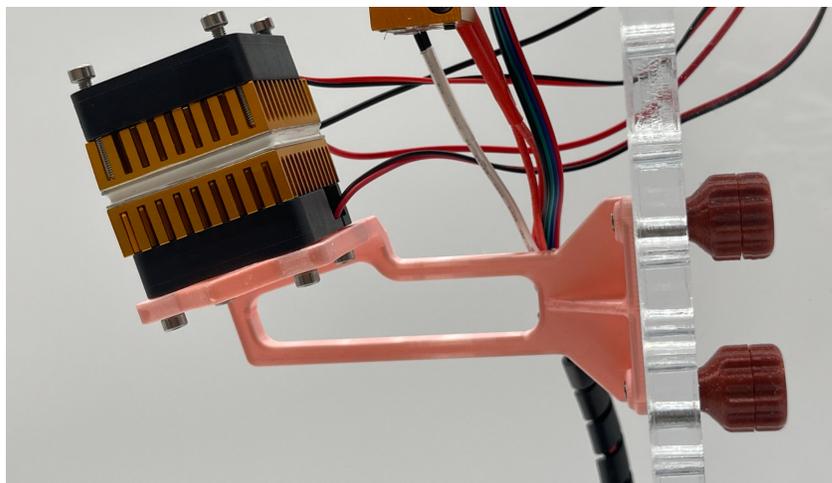


Figure 14. This shows the assembly of the Peltier-heat sink system with axial fans to help have an efficient transfer of cooled air on the fiber.

However, despite the promising theoretical aspects of this design, practical challenges arose during the implementation and testing phases. The primary concern was the insufficient cooling capacity, as the cooled air directed towards the fiber did not reach the desired level necessary for effective fiber cooling. Moreover, the integration of the heat sink and fan system resulted in a substantial increase in bulkiness, making the overall assembly complex. Furthermore, the heat sink connected to the Peltier's hot region exhibited a rapid temperature rise, surpassing the cooling capacity of the

corresponding fan. As a consequence, the system became impractical, with the risk of user injury due to excessive heat generation, making it hazardous to touch the overheated section.

2.3.4 Using centrifugal fans:

In the subsequent phase of the design process, a centrifugal fan was introduced as a key component to enhance the cooling system. A centrifugal fan is a type of fan that operates by drawing air into the center of the fan, and then expelling it outward perpendicular to the axis of rotation. This design of the fan creates a more focused and directed airflow compared to conventional axial fans, due to the fact that the air from the axial fan is channeled through a larger area while the centrifugal fan channels the air through a small exit, resulting in higher airspeeds and increased air pressure. By utilizing a centrifugal fan, the cooling effect generated became more concentrated and centric to the fiber [11]. This directed airflow allowed for a more efficient and precise cooling process, ensuring that the cooling effect was applied directly to the fiber's target area.

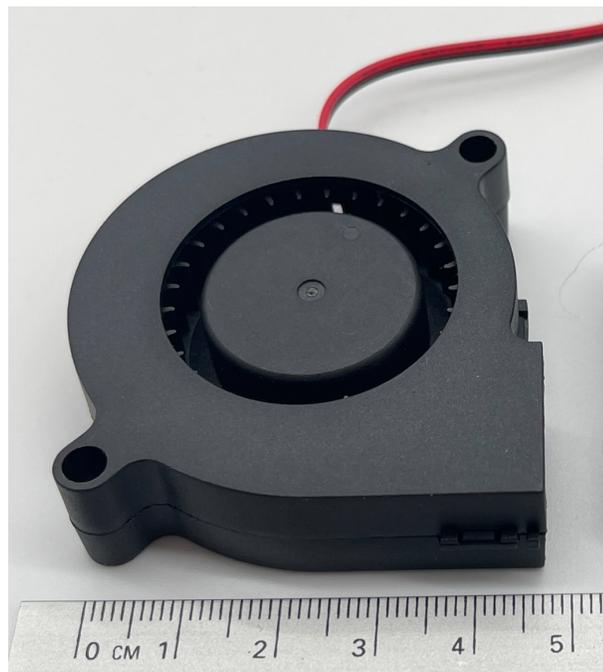


Figure 15. This is the DC 12V 0.10A~0.15A 50mmx15mm centrifugal fan that will aid concentrate the fiber to the fiber.

Experimental testing was conducted by manually holding the centrifugal fan, while monitoring the cooling impact on the fiber. The results were promising, as the fiber experienced a significant

improvement in cooling efficiency. Consequently, the fiber no longer exhibited issues such as sticking on the pulleys, and its overall durability was enhanced, reducing the likelihood of premature tearing during the drawing process. Building upon the positive outcomes observed, further advancements were sought to optimize the cooling rate. To achieve this objective, the decision was made to incorporate two centrifugal fans in the cooling system. This configuration aimed to create a more robust and comprehensive cooling effect, ensuring that the fiber could attain complete cooling before proceeding to the subsequent drawing process.

2.3.5 Made a modular frame for the centrifugal fan:

One of the crucial considerations in the design process was to ensure that the angle at which the centrifugal fan is tilted towards the fiber provided optimal cooling. Recognizing the significance of various angles to achieve this objective, a modular approach was adopted to allow for adjustability in the fan's frame. This modular design enabled the frame for the centrifugal fan to be adjusted according to the desired angle, thereby ensuring precise and targeted cooling. To achieve this adjustability, the frame for the centrifugal fan was designed with a gap slot, situated between the fan's mounting section and the support element that is screwed to the back of the FreD system, as illustrated in Figure 16. The supporting element is outfitted with a rear aperture, positioned to accommodate M3 screws. This configuration allows for secure fastening of the fan when the desired angle is adjusted.

In the manufacturing process, the frame with its gap slot was fabricated using 3D printing technology. However, to optimize the functionality of the frame's adjustable feature, additional post-processing was required. After 3D printing, manual twisting of the frame was necessary to break any residual strings within the gaps. This step was vital in enabling the full range of motion of the frame, ensuring smooth and reliable adjustment of the fan's tilt angle. Moreover, the modular design also facilitated ease of maintenance and future upgrades. In the event of component wear or system optimization, the frame could be conveniently adjusted or replaced without the need for significant structural changes or system disassembly.

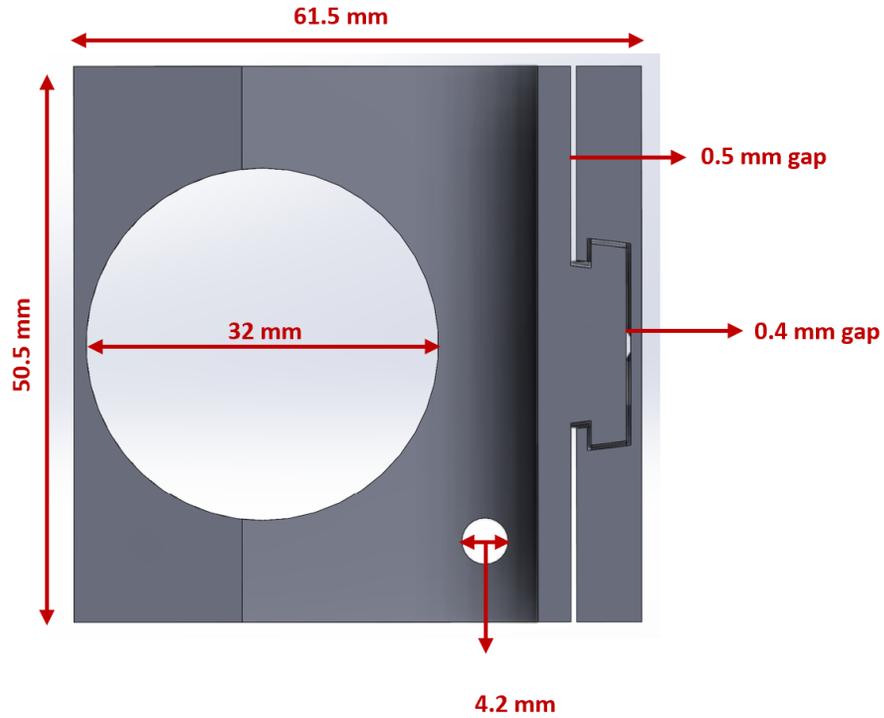


Figure 16. This shows a right sided cross-sectional view of the frame design to allow it to rotate at an angle possible after being 3D printed.

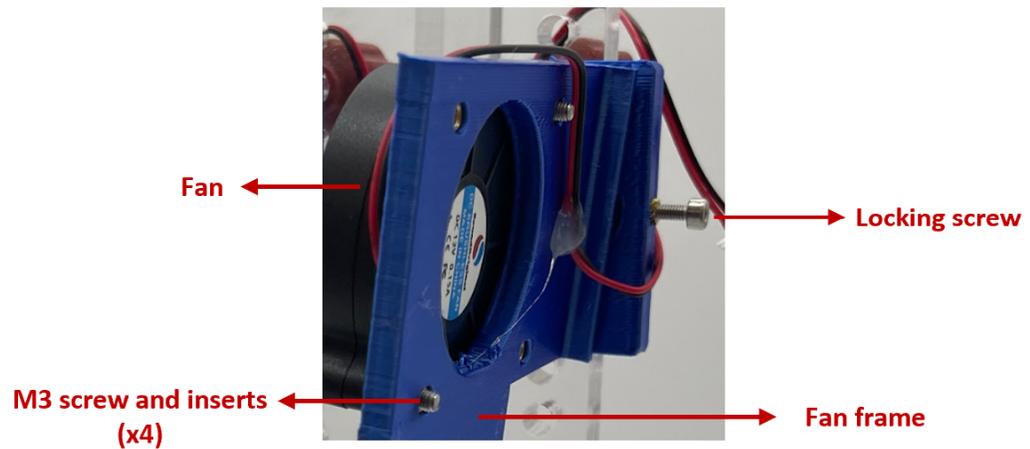


Figure 17. This shows the M3 screw that is used to secure the angled fan after it has been adjusted.

To integrate the fans with the FrED system and ensure superior cooling performance, the frame was positioned in close proximity to the extrusion system, right after the fiber drops. To achieve this, a square support frame was conceptualized, featuring a central hole to accommodate the fan area, thus enabling proper and efficient air circulation. The frame's design ensured a precise fit with the centrifugal fan, and it was secured to the fan using M3 screws.

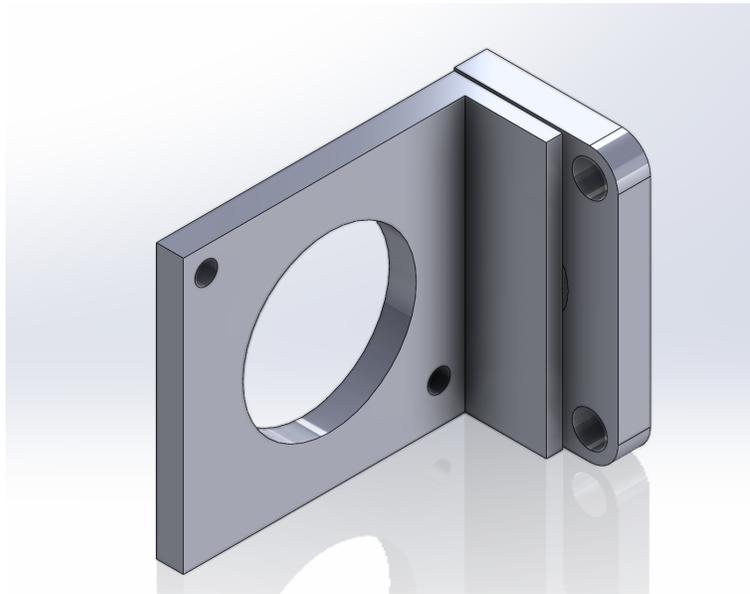


Figure 18. This is the design of the frame that will be used to help mount the centrifugal fan to the FrED system.

To enable easy integration with the FrED system, a rectangular base was incorporated on the side of the frame, serving as the connector to the back of the FrED system. For secure fastening, M5 screws were employed to connect the frame and base to the back of the FrED system. The gap allowed the frame to be angled at various degrees, helping to achieve the most effective cooling angle for the fiber.

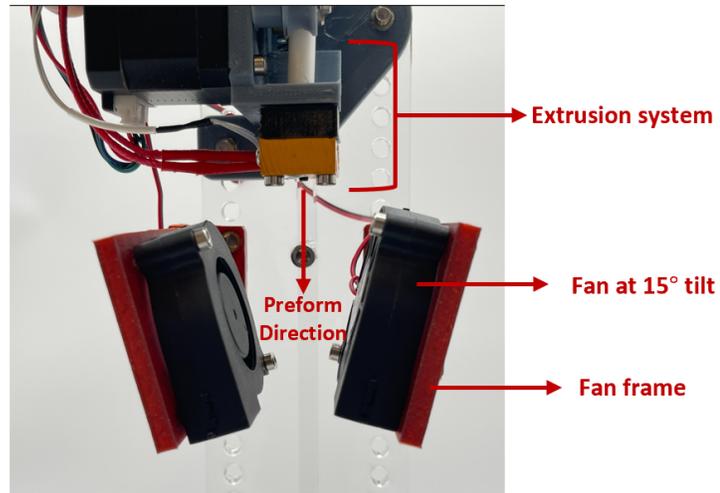


Figure 19. This is the centrifugal fan with its frame mounted to the FrED system close to the extrusion subsystem to allow early cooling of the fiber right from extrusion.

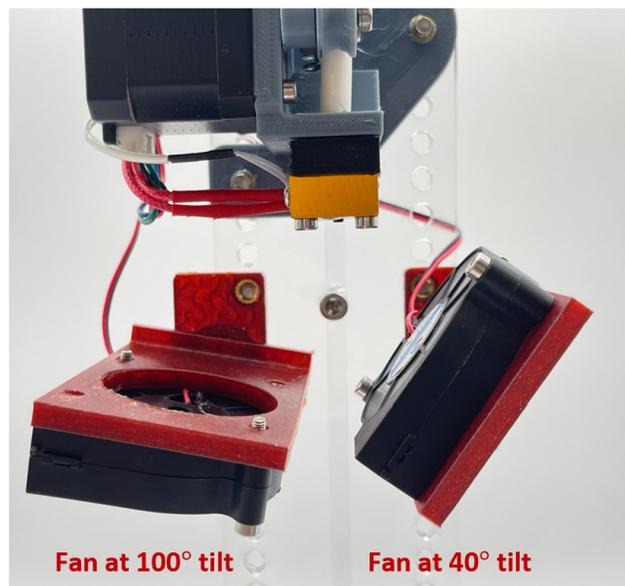


Figure 20. This is to show the wide range of orientation of the fan frame can be angled to aid efficient cooling and concentration on the fiber.

2.3.6 Made it with a hood:

Following the successful creation of the centrifugal fan's mounting frame, the initial cooling process yielded encouraging results, yet there remained an opportunity for further enhancement to achieve a higher degree of fiber solidification. Despite the concentrated cooling effect directed towards the fiber, a residual softness persisted, prompting the need for optimization to attain a more robust and solid fiber structure. As the cooled air traversed the system, it was observed that a certain degree of air dissipation occurred, particularly evident as the airflow reached the end of the fiber just prior to the first pulley in the drawing process. This dissipation hindered the air's direct impact on the fiber at this stage.

To tackle this challenge, an additional refinement was introduced to the centrifugal fan's frame in the form of an extended hood. This hood was designed to descend towards the bottom and placed to interact with the fiber's path. The extended hood's positioning ensured that the cooling air remained engaged with the fiber throughout its trajectory, allowing the cooling effect to persist even as the fiber approached the first pulley in drawing process. The substantial cooling effect imparted by the directed airflow served to solidify the fiber significantly, resulting in a marked reduction in stickiness to the pulleys and ensuring the fiber retained its structural integrity as it was coiled up during the spooling process.



Figure 21. This is the fan frame with hood to help prolonged concentration of the cooled on the fiber length.



Figure 22. This is the fan frame with the hood mounted to the FrED system to enhance cooling of the fiber.

The modularity and versatility of the fan holder enables the comprehensive evaluation of different cooling configurations using the same fan holder, an aspect that is significant from an educational standpoint. The user-centric design approach allowed for user-friendly experimentation with the cooling system, offering a unique opportunity for users to personally manipulate the fan holder and witness firsthand the resultant alterations in the cooling process. This educational dimension of the design affords users an interactive platform to gain insights into the intricacies of fiber cooling dynamics and empowers them to optimize the cooling process to suit their specific requirements. Empowering users with the capability to fine-tune the cooling system not only contributes to their experiential learning but also encourages a deeper understanding of the underlying principles governing efficient fiber cooling.

2.3.7 Used the 3D printed square washer to identify the right fan frame length:

Upon implementing the cooling approach with the directed airflow towards the fiber, another observation surfaced that necessitated further refinement. It was observed that the airflow being generated interacted with the backside of the fiber as it traversed the FrED system. This interaction resulted in the dissipation of the airflow behind the fiber, rather than being optimally directed onto the fiber itself.

To achieve consistent contact between the cooling airflow and the fiber throughout its trajectory, square washers, varying in size (1mm, 2mm, and 5mm) were 3D printed. These washers were designed to serve as spacers, enabling the controlled adjustment of the fan's distance from the system.

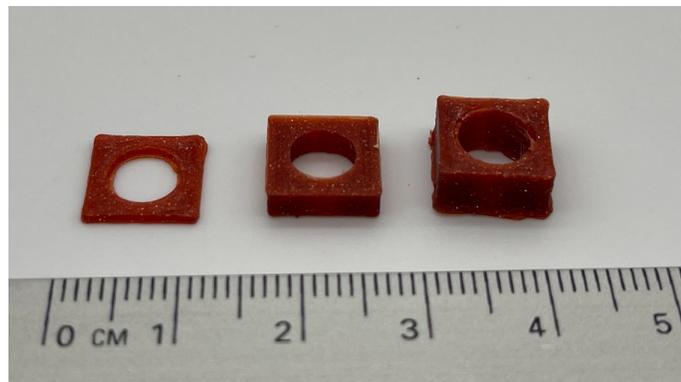


Figure 23. These are the square washers that were used to help figure the right length of the fan frame to allow the air to continually target the fiber during the drawing process.

The process of experimenting with washers of different dimensions allowed for precision in fine-tuning the cooling setup. By altering the fan's proximity to the system, the goal was to identify the ideal separation distance that would enable continuous and effective contact between the cooling airflow and the fiber.

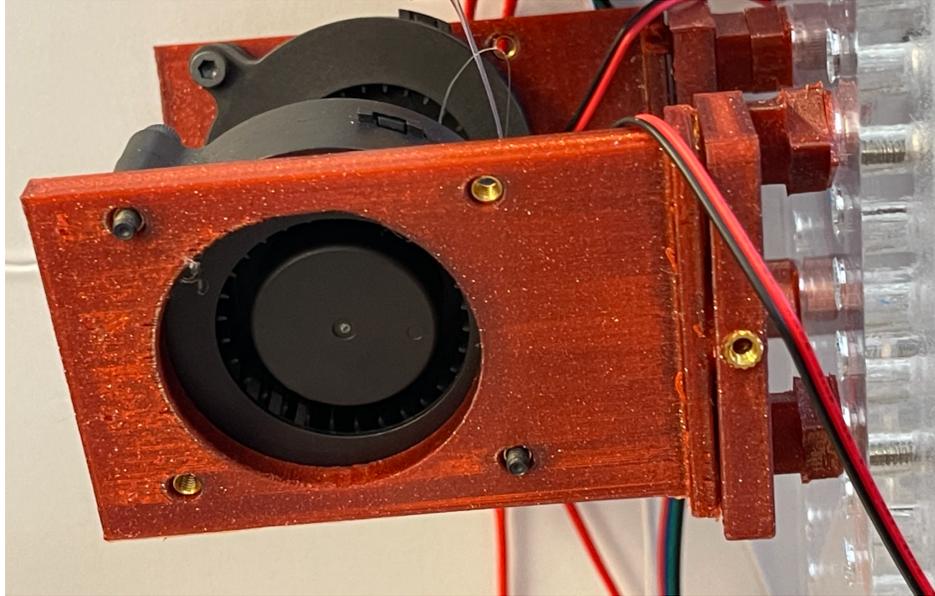


Figure 24. This shows how the washers were used to find the right length that the frames needed to be to continually target cooled air to the fiber.

This methodological approach proved to be efficient because introducing an entirely new fan holder frame with different lengths would have incurred significant material waste. The modular integration of square washers offered a versatile means to validate the optimal length configuration. This approach not only minimized material wastage but also facilitated a practical means of assessing the required length. Upon the identification of the optimal distance using the square washers, a final iteration of the fan holder frame was produced.

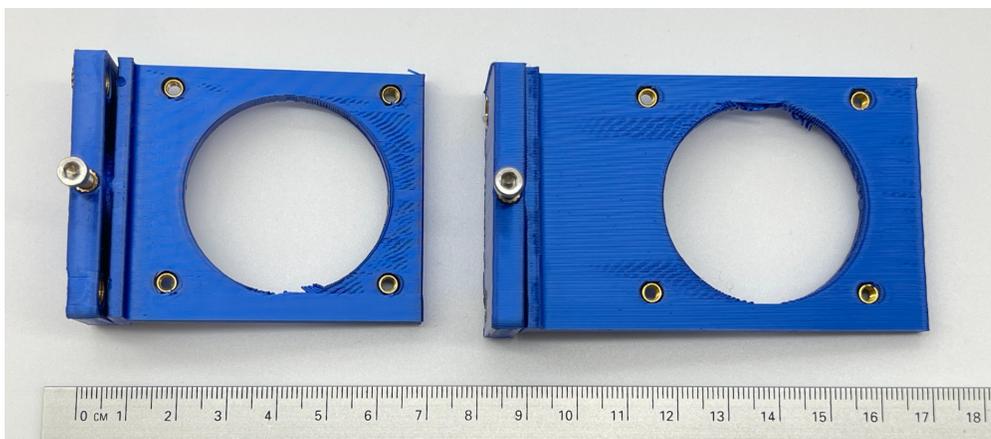


Figure 25. This is an example of how the length of the frame (without the hood) change after determining the right length for the frames to enhance its cooling capabilities.

The overall dimensions of the FrED system underwent reduction, particularly to the redesign of the back rib and base components, as outlined in section 6.2. Given the compacted dimensions, the smaller-sized FrED system exhibited an effective and complete cooling performance even in the absence of the hood.

By removing the hood attachment on the fan holder, several significant advantages were realized. Primarily, it reduced the material usage for the hood itself as well as a decrease in the overall print time for the fan holder. Additionally, the absence of the fan hood made the setup and manual maneuvering of the fiber around pulley systems more straightforward.

3. Improving Fiber Stability and Uniformity:

The stability and uniformity of fibers are very important in the fiber production process of FrED, and achieving these qualities is made possible through the strategic implementation of a pulley system.

Fibers are highly delicate and sensitive to tension variations during production. Pulleys are strategically positioned along the fiber's path to provide precise tension control, safeguarding against fiber breakage, maintaining uniformity, and enhancing the overall mechanical properties of the fiber. They also aid in aligning and guiding the fiber accurately during the drawing process. These pulleys are meticulously designed with precise dimensions and alignment to ensure uniform fiber diameters and minimize variations.

Furthermore, pulleys aid in cooling the fiber during the fiber drawing process, helping in efficient dissipation of the excess heat remaining after the cooling section of FrED. This ensures proper solidification of the fiber and preserves its desired physical and optical properties. It also helps in the controlled and precise winding to the spool to prevent fiber entanglement and tangling.

Most importantly, it aids in the stability of the fiber during the drawing process, enabling real-time measurement and evaluation of the fiber's diameter. This is crucial for low-cost FrED, as there is no privilege of using expensive devices for diameter measurement of such thin fibers.

3.1 Previous Design:

The initial design used for the diameter measurement holder incorporated two pulleys into the system, ensuring the fiber's straightness and stability while feeding it to the spool. The pulleys were 3D printed using PLA material and securely connected to their support using screws. The second pulley was positioned very close to the spool to provide a wider guidance area.

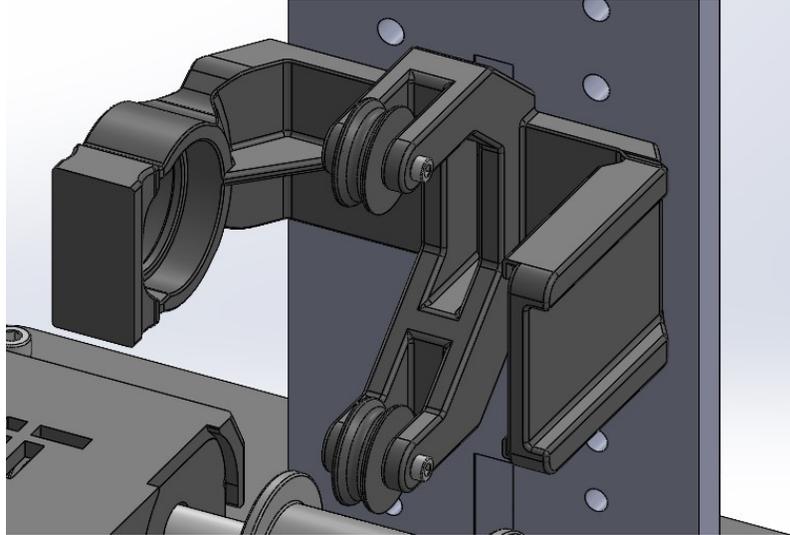


Figure 26. Thus is the configuration of the pulleys for the old version of the FrED system.

The 3D printed pulley featured an internal groove approximately 9 mm wide, which effectively guided and maintained the fiber within the groove during its transit to the spool, accommodating both translational and rotational movements.



Figure 27. This is the 3D printed pulleys that was used for guiding the fiber during the drawing process in the old version of the FrED system.

3.2 Issues:

The pulley presented a significant challenge due to persistent fiber sticking during the drawing process. This stickiness caused the fiber to adhere to the pulleys and move along with them as they rotated, resulting in tension build-up until the force was sufficient to tear off the sticky section. Consequently, in some cases, the entire fiber would break, leading to unpredictable occurrences and hindering the attainment of stable and uniform fibers during the drawing process.

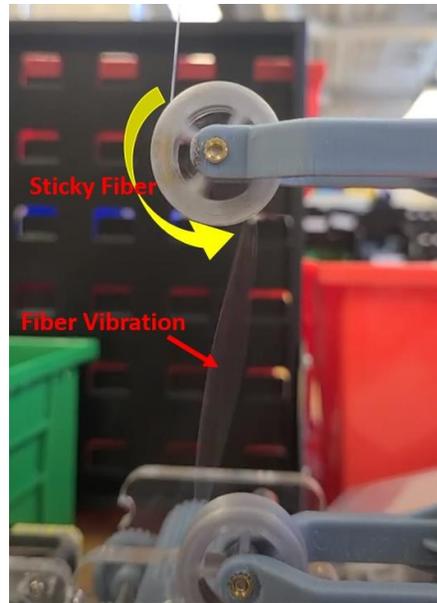


Figure 28. This shows the sticky nature of the fiber on the pulley as well as the vibrations that the fiber experiences during the drawing process.

The fixed distance between the two pulleys further compounded the issue, making it challenging to adjust and modulate the distance between them to alleviate the sticking problem. Additionally, when the fiber tore off from the pulleys, small residues of the fiber would be left behind, becoming lodged in the pulley's groove. Over time, these accumulated residues disrupted the smooth connection between the fiber and the groove, exacerbating the sticking issue.



Figure 29. This shows the residual fiber left behind when it sticks on the pulley.

In addition to that, attempts to remove these residues by scratching them off resulted in markings on the pulley's surface, rendering the groove uneven and diminishing its uniformity. This marked an additional obstacle in achieving the desired level of precision and consistency in fiber production.

3.3 Design Changes:

3.3.1 Change it to pulley support that can be moved up and down:

As stated above, the initial configuration of the pulley holders exhibited an inherent limitation, whereby the interconnection of the pulley holders remained fixed. In response to this constraint, a revised design with the primary objective of enabling dynamic repositioning of the pulley holders along the system. The design included the integration of a foundational base element affixed to the system's rear, accompanied by two extension arms that held the pulleys. The ends of the arms had holes for insertion of M3 screws to fasten the pulleys from one side to the other.

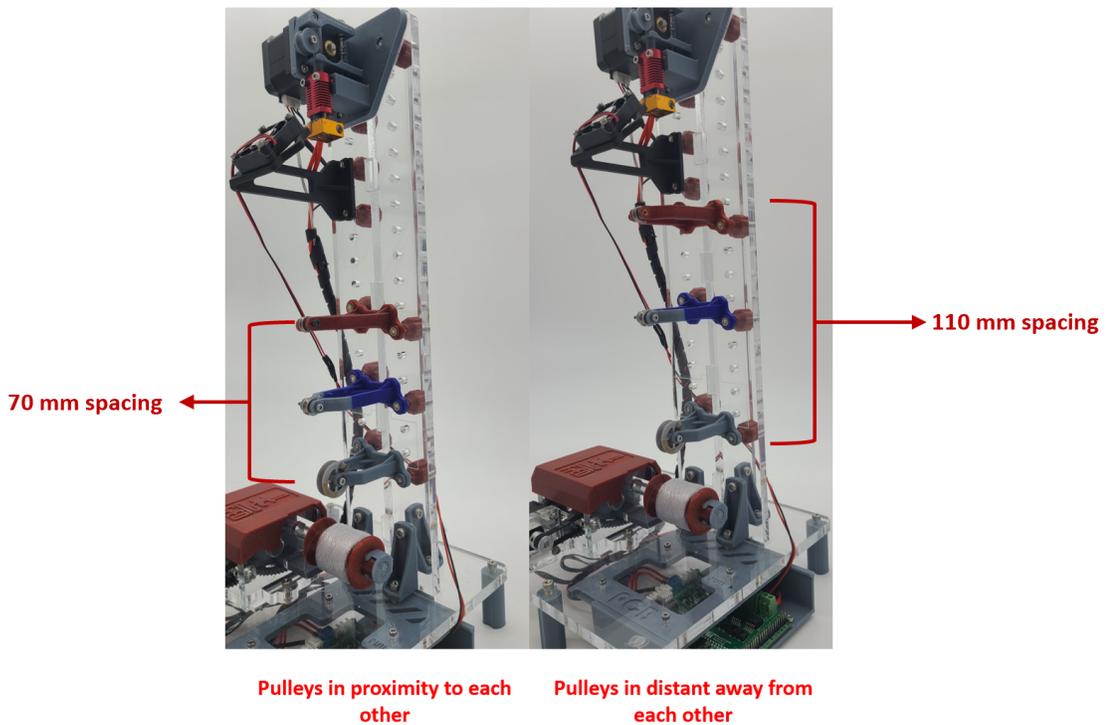


Figure 30. This is the pulley support that can be moved anywhere along the frame of the FrED system.

The importance of the design was being able to manipulate and fine-tune the spatial relationships between the pulleys, as well as with the extrusion system. This benefit of maneuverability lay in its role in accommodating various configurations, eliminating any constraint on the number of pulleys essential for ensuring the stability of the fiber during the drawing process. Due to the number of components and space on the frame of the FrED system, it was decided to use three pulleys to attain stability of the fiber.

The first pulley was positioned at a distance ranging from 100 to 120 mm from the extrusion system to allow the fiber an extended window for the fiber cooling process. The remaining pair of pulleys underwent iterative adjustments, to determine optimal spacing that would effectively underpin the fiber's stability during the drawing process. However, regardless of the positioning of the pulley, the fiber was still sticking to the 3D printed pulley and was still unstable.

3.3.2 Tried different PLA printed pulleys with different grooves:

This design iteration was a 3D-printed pulley with a wider internal groove, approximately 0.3 mm larger than the previous pulley. The primary modification was introduced in the groove of the pulley. Diverging from the previous iteration's uniformly smooth groove, this iteration was characterized with intricate bump formations within the pulley's groove. The reason behind this design variance stemmed from the former smooth-surfaced groove, which was susceptible to adhesive interactions with the fiber, impeding its smooth traverse through the pulley. The integration of these minute bumps should allow the fiber with a series of discrete points of contact with the groove, effectively reducing the extent of surface adhesion of the fiber.

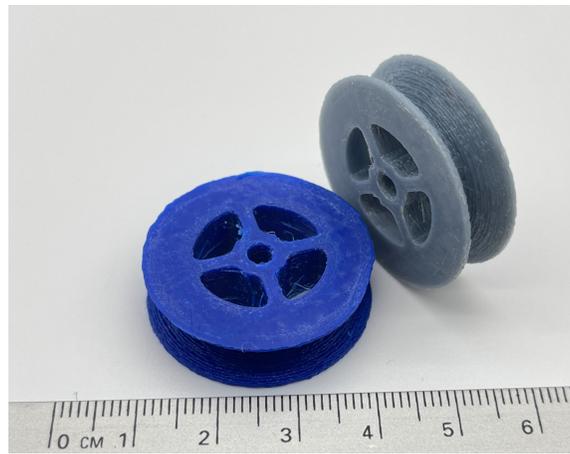


Figure 31. This is the pulley with little bumps in the internal groove to reduce stickiness of the fiber.

The residual adhesive tendencies persisted and there were still vibrations experienced by the pulley contributing to the instability of the system. Due to that, aerosol-based surface sprays were used on the pulleys to mitigate the adhesion-related concerns. Specifically, two aerosol sprays were a silicon-based lubricant spray and a Teflon-based dry lubricant spray.



Figure 32. This the two aerosol sprays used to reduce the stickiness between the 3D printed pulleys and the fiber.

The outcomes of the lubrication were unable to wholly remove the adhesive tendencies inherent to the fiber-pulley interface. Notwithstanding the utilization of these lubricants, the stickiness persisted.

3.3.3 Plastic pulleys:

Due to the issue of fiber adherence to the pulleys, a decision was made to explore alternative non-3D printed pulley options. With that, low-cost plastic pulleys of three different kinds were used each varying in size, with the objective of mitigating stickiness and augmenting system stability as seen below.

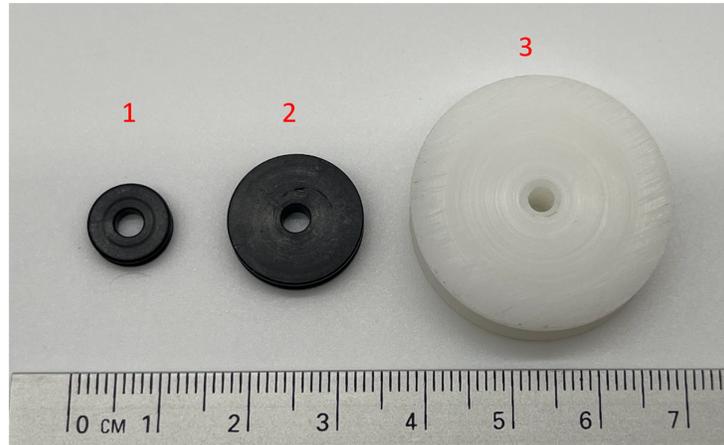


Figure 33. This is the different plastic pulleys that were used to enhance stability of the fiber.

Initial trials commenced with the use of the smaller pulley (1), serving to validate the reduction in stickiness. While an observable decrease in stickiness was noted, there were instances where the fiber was unstable. A notable improvement was the absence of residual fiber remnants on the pulleys post-detachment after they stick to them. A slightly larger pulley was introduced to increase the tension in the fiber (2). This adjustment was done to cause the fiber traverse around a greater circumference, thereby fostering more stability. Despite that, persistent vibrations within the fiber persisted, preventing it from consistently remaining within the confined internal groove.

In response, a larger pulley with a wider internal groove was used (3). This implementation aimed to not only amplify fiber tension for stability but also to afford greater lateral maneuvering room for the fiber within the groove. This adjustment ensured the fiber within the groove during lateral displacements. However, none of the employed pulley configurations entirely mitigated the issue of stickiness, even though it displayed a noticeable improvement over the performance of the 3D printed parts.

3.3.4 Metal pulleys:

Since the plastic pulleys weren't providing sufficient stability and were unable to prevent stickiness issues, metal pulleys were used.



Figure 34. This is the different metal pulleys that were used to enhance stability of the fiber.

Metal pulleys offer notable advantages over their plastic pulleys, particularly in terms of reduced stickiness and improved overall performance (1 and 2). However, challenges arose when dealing with larger fiber diameters, where a degree of stickiness persisted. Complicating matters was the presence of a small internal groove, which posed difficulties in consistently maintaining the fiber within the groove during the drawing process.

Addressing these concerns, a pivotal factor contributing to the enhanced stability of the fiber within the groove was the incorporation of ball bearings within the metal pulleys (3). This integration of the bearings facilitated a smooth and seamless rolling movement of the pulley, mitigating disruptions caused by stickiness and contributing to the stability of the fiber. To further refine the system, a larger pulley featuring a more intricate internal groove was introduced to better secure the fiber within the groove and promote stability.

Nevertheless, despite these efforts, challenges persisted, with the fiber intermittently straying from the groove. The breakthrough came with the development of a final pulley design from steel, boasting both an extensive depth in its groove and a wider internal groove with ball bearings (4). This steel pulley overcame issues related to fiber stickiness while simultaneously delivering unwavering stability. Even when dealing with larger fiber diameters, the resulting stability was visually evident, with the fiber exhibiting minimal vibrations.

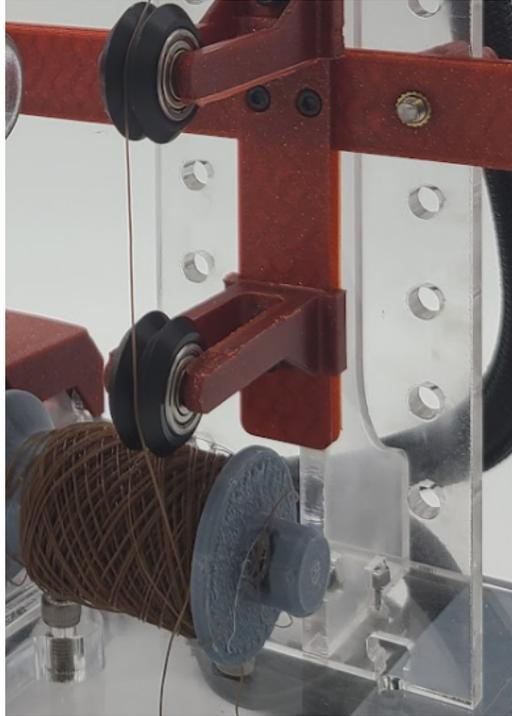


Figure 35. This shows the fiber very stable with the eye when the steel pulley was being implemented.

The choice of using aluminum for the last pulley design was motivated by its cost-effectiveness and lightweight nature (5). This combination rendered the pulley both economical and maneuverable, ensuring smooth and faster rolling and movement, even when accommodating fibers with smaller diameters.

3.3.5 Support with the ball design:

When the FrED is operated over an extended period, minute vibrations become noticeable and lead to the shaking of the fiber. The underlying issue here is that the pulley itself experiences significant oscillations due to its connection with the holder. The problem arises from the screw's inability to fit flush within the internal hole of the pulley. This misalignment prevents the pulley from moving smoothly and makes it susceptible to easy tearing.

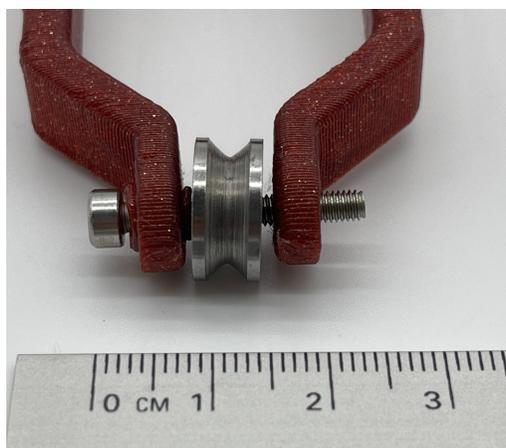


Figure 36. This shows an example of the gap with the pulley and the pulley support.

Furthermore, even a slight gap between the screw and the pulley causes the pulley to exhibit slight movements as the fiber interacts with its surface. Compounding the issue, the screw employed to secure the pulley introduces complications when tightened excessively because it then restricts the movement of the pulley especially those without bearings.



Figure 37. This is an example of the pulley when it tightly connected to the pulley support.

In retrospect, the use of screws as connectors for pulleys was the best idea because the threading of the screws disrupts the pulley's symmetric movement and, in some instances, worsens vibrations within the pulley assembly. This design flaw prompted a shift towards using ball connectors instead. These new connectors are integrated into the holes of the pulley as seen below. The spherical design of the ball connectors enables the pulley to roll smoothly and seamlessly. This

modification has effectively addressed the vibrations caused by screws and ensures a more stable and balanced operation of the pulley mechanism.



Figure 38. This is the design of the pulley support with the ball (left side) and also shows it being connected to the pulley (right side)

3.3.6 Alternating the fiber direction (one behind-two in front), shortening one support, and making it one-sided:

During this stage, the fiber exhibited stable behavior visible to the naked eye, even when the FrED was operated for extended periods. However, upon mounting the microscope, vibrations were still evident when observed through the lens. Following discussions with Prof. Brian Anthony, a method was devised to introduce greater tension to the fiber, thereby enhancing its stability.

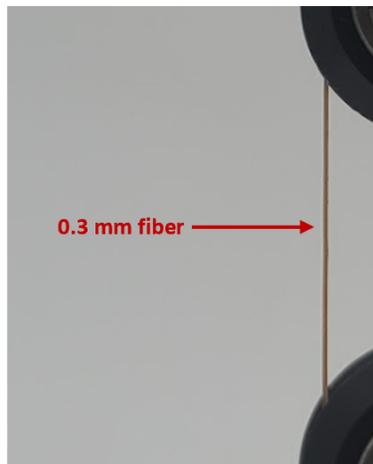


Figure 39. This shows a 0.3 mm stable fiber with the naked eye.



Figure 40. This shows the instability of the fiber when looked through the microscope lens causing the diameter to increase to 0.6 mm.

This approach involved altering the fiber's path around the pulleys. Specifically, the fiber was routed behind the first pulley and then positioned in front of the second and third pulleys. This adjustment proved effective as it was observed that the pulley stability improved significantly after the fiber passed behind the first pulley. This sequence allowed for a remarkably smooth and stable passage of the fiber over the second and third pulleys before being spooled.

Furthermore, in pursuit of augmenting fiber stability through increased tension, adjustments were made to shorten the length of the first pulley support. Additionally, a one-sided design was implemented for the support of the first pulley, achieved through the incorporation of a snap-lock pin system to anchor the pulley securely. These modifications collectively contributed to a notable improvement in overall system stability.

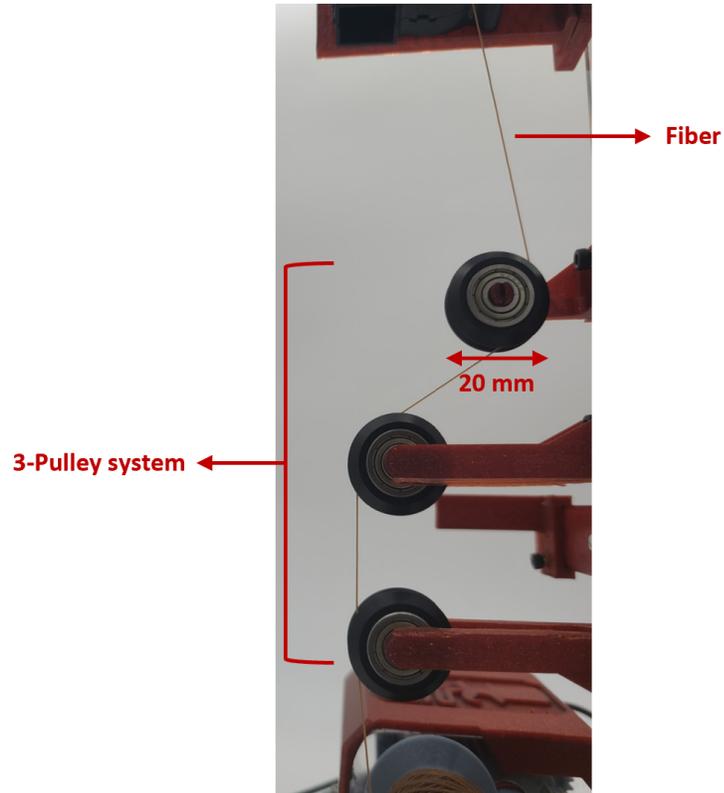
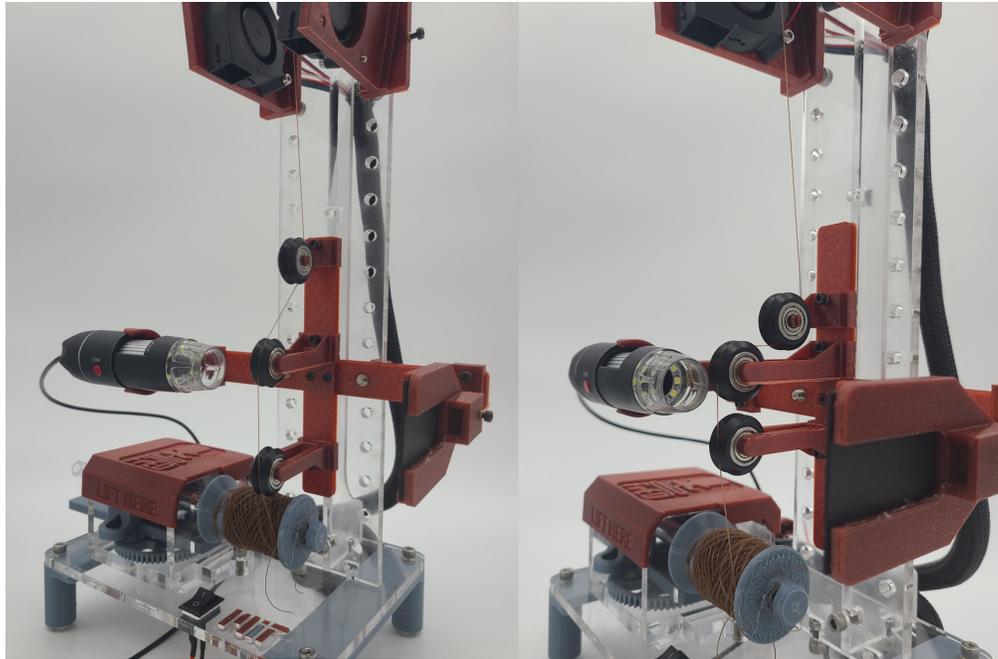


Figure 41. This shows the path of the fiber to achieve optimal stability for both the eye and lens of the microscope.

3.3.7 Fully modulated pulley system to go up and down:

For enhanced and convenient adjustment of the pulley system, a fully modulated system was designed and 3D-printed. In this system, the pulleys have the ability to slide along a rack vertically and are secured using M3 screws when adjusting distances accordingly. The comprehensive design details are available in Wenhao Xu's thesis paper. This development was of paramount importance as it simplified the process of fine-tuning the pulley system, accommodating variations in fiber diameter and system vibrations while maintaining stability. Moreover, this advancement holds educational value, enabling users to make adjustments that showcase the varying degrees of fiber stability in different scenarios.



Wide range

Close range

Figure 42. This shows the range at which the pulleys can be modulated.

4. Refining the Gear System:

4.1 Previous Design:

In the previous design, the gear system was implemented with the scotch-yoke mechanism, a solution that entails the interconnection of two distinct sets of gears. One set of gears facilitates rotational motion, while the other set governs the reciprocating movement. This arrangement is implemented with specially adapted and customized bevel gear in conjunction with a standard bevel gear which were 3D printed. The system was also designed to ensure both repairability and replaceability. This included a modular plate mechanism, effectively connecting the side walls of the gearbox. This enables users to extract the entire gear box as a unified assembly for simplified repair and replacement operations. Moreover, expedited spool removal during data collection sessions was achieved by outfitting one end of the spool with an intuitive sliding lock mechanism. A gearbox cover, adorned with the FrED logo, was developed to enclose the gearbox.

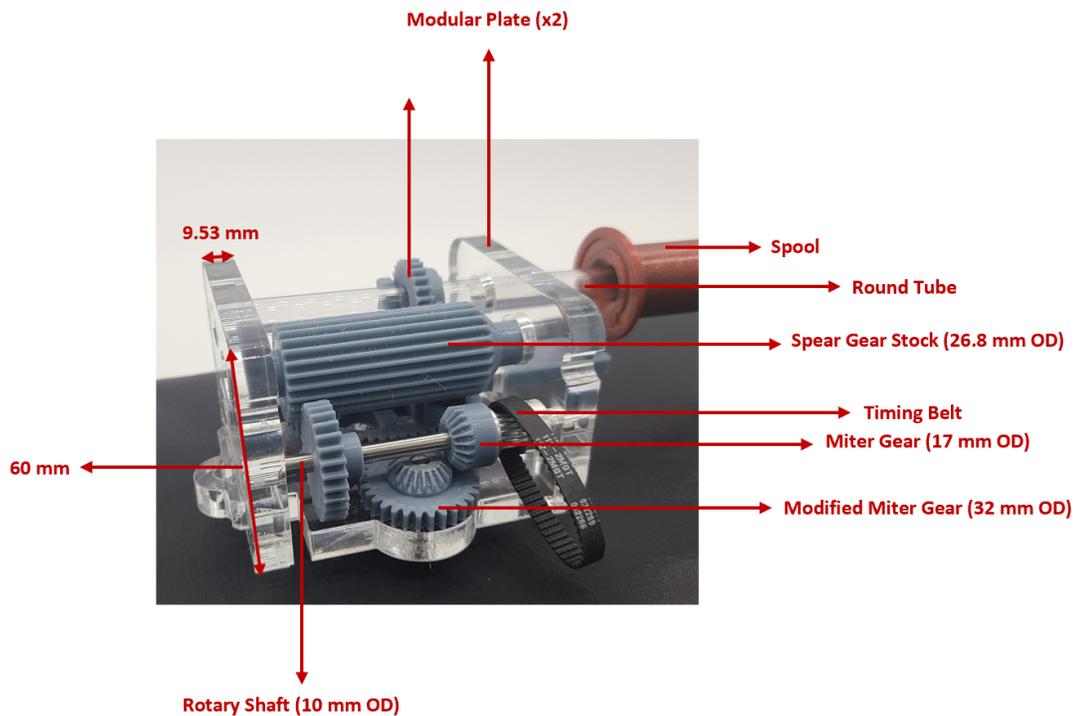


Figure 43. This shows the modulation of the gear system with the ball bevel gears.

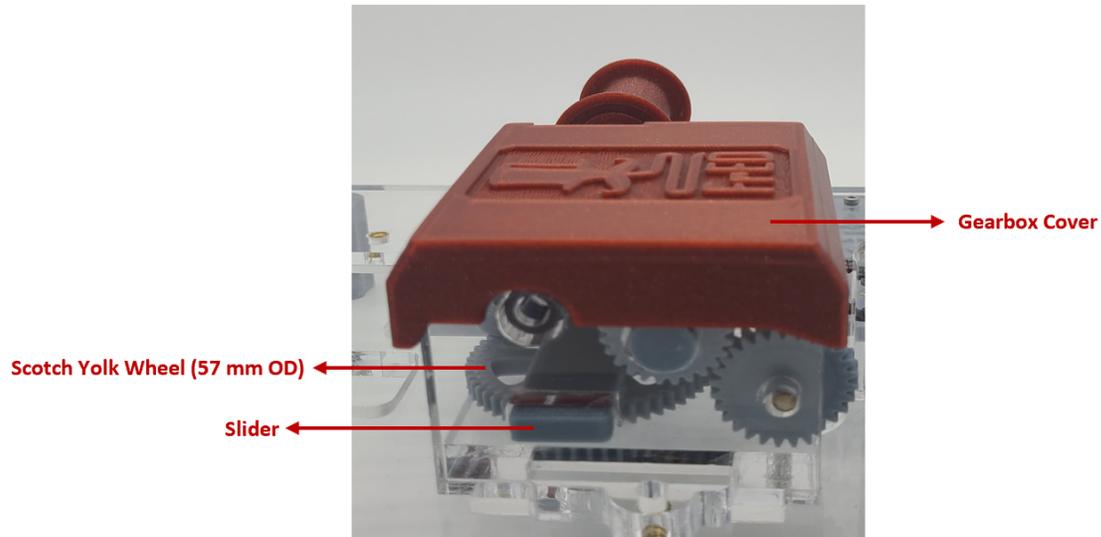


Figure 44. This is the gear system with the gear box cover on top of it.

The functionality of the gear system in the previous design demonstrated satisfactory performance concerning the rotational and translational motion of the spool during the fiber drawing process. Extensive testing of the system revealed that the 3D printed gears and pulleys exhibited impressive durability, with an average operational lifespan of approximately 2 months. The system underwent continuous operation for approximately 1.5 hours per day before any noticeable signs of wear were observed.

4.2 Issues:

Despite the gear system's performance, a notable drawback arose in the form of excessive noise generation during operation. The noise levels reached approximately 85-95 dB, which proved to be highly disruptive and over extended periods, became practically unbearable. Furthermore, this noise was transferred into the FrED system, adversely affecting fiber stability and uniformity during the drawing process.

Another issue stemmed from the utilization of a stepper motor in the previous design. The motion from the stepper motor resulted in noticeable steps felt at the base of the system, impacting the overall smoothness of the operation. Additionally, prolonged operation of the stepper motor caused it to heat up significantly, with temperatures reaching around 110°C, posing potential hazards if accidentally touched.

4.3 Design Changes:

4.3.1 Different materials used to reduce noise and vibration:

To tackle the persistent noise and vibration issues, there was a thorough exploration of various damping options. Among the alternatives considered and tested, a significant focus was placed on employing specialized vibration dampeners designed for stepper motors. These dampeners demonstrated their effectiveness by significantly reducing both motor-generated vibrations and the associated noise. The incorporation of these dampeners was a promising step towards achieving a quieter and more stable system.

Furthermore, we ventured into using cork dampeners and rubber push-in bumpers for the screws, seeking to address multiple sources of vibration simultaneously. The advantage of these dampening materials lay in their pliable nature, which enabled them to absorb vibrations more efficiently than the stepper motor dampeners alone. This strategic choice allowed us to tackle both the motor's vibrations and the mechanical vibrations transmitted through the screws as well.

Despite our dedication to these damping solutions, the resultant noise levels were reduced to a range of 80-86 decibels. However, it became apparent that although the noise was mitigated, the system remained audibly disruptive, particularly during extended operational periods. This realization spurred us to continue our exploration in pursuit of a more comprehensive solution that could ensure not only reduced noise but also enhanced stability for prolonged operations.



Figure 45. This shows the different dampening alternatives that were used to reduce the noise and vibrations.

4.3.2. Moving the gear box inwards to have less tension on the rubber:

Another approach implemented to tackle this issue involved elevating the stepper motor to establish clearances between the motor and the base. This adjustment was intended to prevent the transmission of vibrations from the motor to the base of the FrED system and subsequently to other interconnected components. While this elevation did contribute to vibration reduction, it was discovered that the board securing the motor still facilitated vibration transfer.

In the original design, the board was positioned on the base through a slot and subsequently glued, resulting in a substantial surface area of contact between the board and the base, conducive to the transfer of vibrations. To address this concern, the board was now situated atop the base, secured using a T-slot system that limited the contact area between the board and the base. This alteration effectively curtailed vibration transmission.

Furthermore, attention was drawn to the belt connecting the stepper motor to the gear system, responsible for enabling rotational and reciprocating motion. It was observed that the belt was excessively tight, generating increased tension between the components and contributing to the overall noise level. This was rectified by slightly pushing the stepper motor to slacken the belt tension. This tweak led to a reduction in noise, but despite these efforts, noise levels persisted at approximately 78-83 decibels.

4.3.3 Changing hanging motor to DC motor:

After many different alternatives to reduce the noise and vibration, including taking out individual components of the gear to identify the issue, the stepper motor was the major reason for the noise and vibration, in addition to that the heat as well. With this stepper motor, movement occurs in predefined steps with a known number of steps per revolution as well. A DC motor was used instead of the stepper motor since it moves smoothly, and its lack of discrete steps. However, the use of a DC motor necessitates additional coding and a supplementary general-purpose input-output port for its encoder.

The initial step after procuring a DC motor involves testing for vibration and temperature. Running the DC motor continuously on a simple circuit for 1.5 hours reveals that the temperature remains

stable at around 36 degrees Celsius, while the noise level is significantly reduced compared to the stepper motor. The change to DC motor also caused a change in the mount.

Sashlin Jagdessi, one of the UROPs, designed a 3D printed mount, iterating through multiple versions to achieve the desired form to hold the DC motor secure. To alleviate noise issues caused by slot inserts in previous motor mounts, the decision was made to position the DC motor mount directly onto the frame base using alignment features, secured by screws.

The DC motor's relatively shorter shaft posed additional challenges, leading to a thinner mounting plate. However, the DC motor mount underwent rigorous stress testing, totaling over 100 hours at speeds twice the normal operation range, with no observable damages. The current noise level is maintained at 65 to 69 decibels. The primary sound emanates from the gear box, falling within an acceptable range of noise even after operating the system for prolonged hours. Furthermore, Dr. Brian Anthony was in support of the DC motor as it was an excellent educational tool for learning feedback controls.



Figure 46. This shows different mount iterations and the final design with the DC motor connected to it.

5. Integrating Computer Vision Technology:

In fiber production, maintaining consistent diameter measurements is a critical factor that directly impacts the quality of the fibers. Ensuring accurate and precise fiber diameters throughout the production process is of utmost importance for several reasons. Firstly, uniform fiber diameters ensure consistent mechanical and optical properties in the final product, leading to enhanced performance and reliability. Secondly, in applications where fibers are utilized for data transmission, such as in telecommunications or optical networks, variations in fiber diameter can significantly affect signal propagation, resulting in data loss and signal degradation.

5.1 Laser Micrometer:

To achieve real-time diameter data for feedback control and ensure the production of accurate diameter fibers during the drawing process, the research phase involved the utilization of an IG-028 laser micrometer. The IG-028 laser micrometer consists of two main parts: the IG-08T transmitter and the IG-028 R receiver. The system employs lasers to measure the diameter of the fiber by detecting the changes in intensity as the fiber passes through the measurement area [12].

Initially, the laser micrometer was placed beneath the extrusion sub-system to measure the fiber diameter directly from the preform for immediate feedback control.

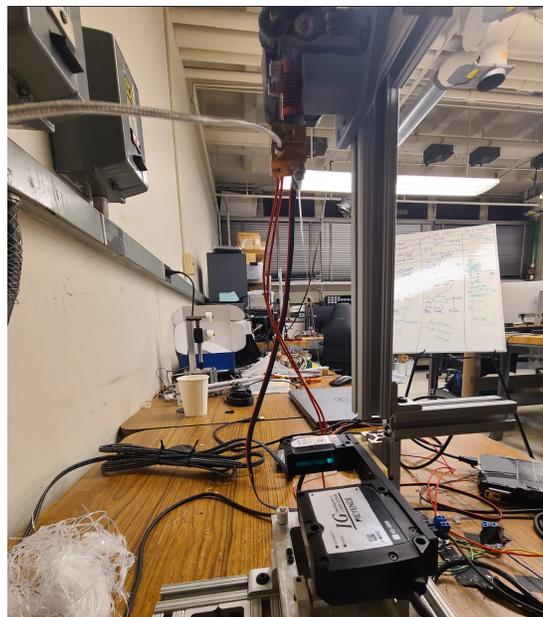


Figure 47. This shows the position of the laser micrometer on the research based FrED.

However, this setup proved impractical for low-cost FrED due to the significant cost of the laser micrometer, which amounted to about \$3,000. As the target price for the low-cost FrED was set to be below \$30, an alternative approach had to be devised.

However, the low-cost FrED started with the laser micrometer to help gain knowledge of the precise diameter and its variation of the fiber being produced. It was integrated into the system by developing a support structure through 3D printing. Careful alignment of the two main parts of the laser micrometer was crucial to ensure accurate readings and avoid erroneous measurements caused by tilts in the internal components.



Figure 48. This shows the IG-028 T and the IG-028 R connected to the 3D printed support for proper alignment of the two component for proper reading.

The connection process involved linking the laser micrometer to the IG-1000 amplifier through a head connection cable that connects the transmitter and the receiver. Proper guidance was provided using color-coded (blue and red) strips on the cables, and lock covers were employed at the tips for secure connections [12].



Figure 49. This shows the IG-1000 amplifier with its connection to the receiver (red strip) and the receiver (blue strip).

The wiring cable included brown and blue wires, which were essential for powering the laser micrometer, while the other wires were insulated to prevent any sparks or shocks.

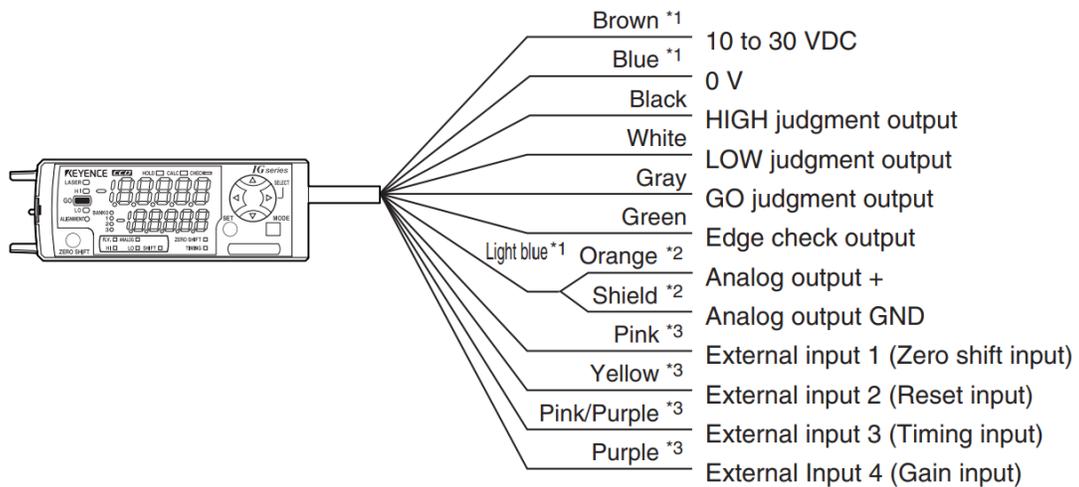


Figure 50. Schematic of the wires for the amplifier with the brown and blue wires necessary for powering the laser micrometer.

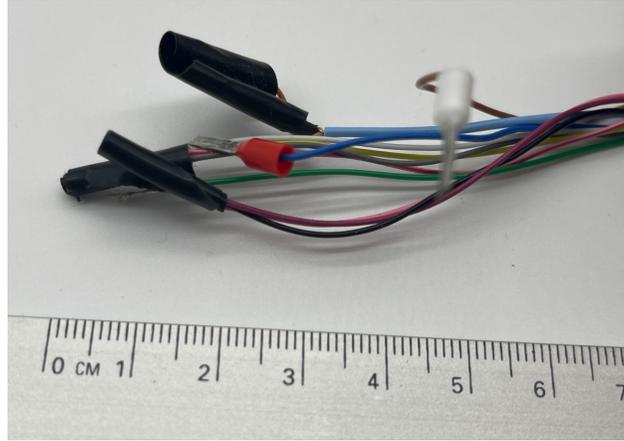


Figure 51. This shows the other wires that were insulated by tape to avoid any shocks when current runs through the amplifier.

After the laser micrometer and the amplifier are on, the MODE button is pressed and held for approximately two seconds to ensure the mode setting is switched to the measurement mode, denoted by "1. Fnc" as displayed on the interface [12]. To navigate through the various setting items, the user has the option to utilize either the left/right buttons or the MODE button, particularly useful if the measurement mode is inadvertently skipped.

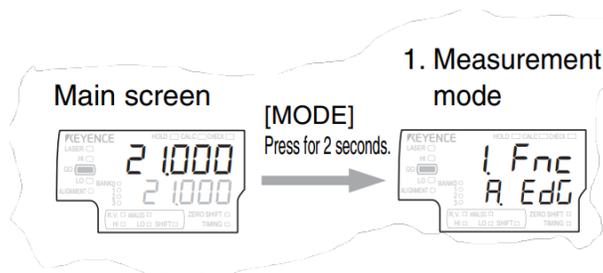


Figure 52. This the settings process to access the measurement mode.

Once in the measurement mode, the outer diameter/width measurement mode is activated by using the up/down button to the mode that is indicated by "b. diA" on the interface [12]. Selecting this mode configures the micrometer to precisely measure the diameter of the fiber as it passes through the measurement area providing for real-time and accurate fiber diameter readings during the drawing process, contributing to the production of consistent and high-quality fibers.

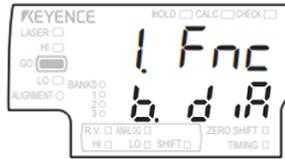


Figure 53. This shows the final display on the amplifier screen to be in the right mode for diameter measurement.

To enable fiber detection and read the fiber diameter on the low cost FrED, the laser micrometer needed to be positioned outwardly from the system's back to accommodate the fiber at a slightly greater distance. This involved a rapid prototyping technique utilizing long screws and nuts to push the laser micrometer outward effectively.

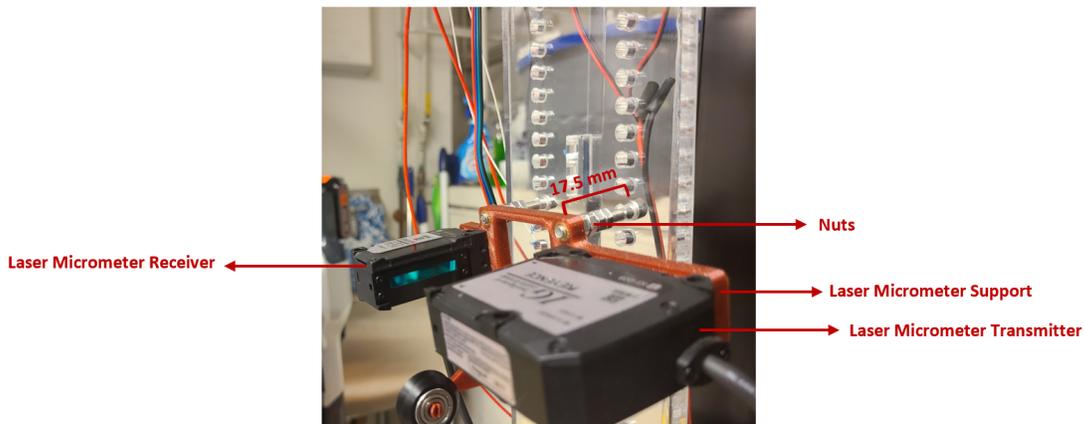


Figure 54. This shows the protrusion of the laser micrometer to detect to provide diameter readings on the amplifier.

The laser micrometer served as a valuable tool for observing the width of the fibers produced by the low cost FrED, as well as identifying variations in diameter. Furthermore, it played a crucial role in comparing the accuracy of the low-cost camera and its subsequent computer vision algorithm for diameter measurements.

5.2 Low cost FrED diameter fiber reading set up:

In the pursuit of achieving cost-effective and stable fiber diameter readings, the FrED system utilized the Jiuson 40-100x digital microscope as the primary camera for image acquisition in the

computer vision algorithm. The Jiuson 40-100x Magnification Endoscope Microscope is a portable and versatile digital microscope offering adjustable magnification levels from 40x to 100x. It features a 2.0-megapixel CMOS image sensor, allowing for high-resolution imaging with a maximum resolution of up to 1920x1080 pixels (1080P). The microscope is equipped with adjustable LED lights for illumination and has a built-in USB 2.0 interface for seamless connectivity to various devices. With its advanced optical components, it ensures accurate and detailed visualization of the fiber during the drawing process.



Figure 55. This is the Jiuson 40-100x digital microscope used on the low-cost FrED system.

The Jiuson digital microscope features a compact design with dimensions of 115mm long and 33mm wide, making it suitable for integration into the FrED system. Additionally, it incorporates front lighting, which illuminates the fiber surface directly, ensuring clear visibility of the fiber structure during the image capture process .

To enhance the stability of the fiber during the drawing process and ensure accurate diameter readings, several critical factors were considered and addressed in the design. First, the fiber's solidity and stability were crucial aspects that required attention. Unlike the laser micrometer, which is sensitive enough to detect the fibers even when they are vibrating and shaking during the drawing process, the FrED system focused on maintaining a stable environment for the fiber drawing process.

For a quick recall, this was achieved by implementing centrifugal fans on both sides of the extrusion system. The fans were angled to have direct contact with the fiber, and they directed air

downwards in the fiber's drawing direction throughout the process. This cooling mechanism significantly improved the fiber's solidity, leading to more reliable and consistent diameter readings.

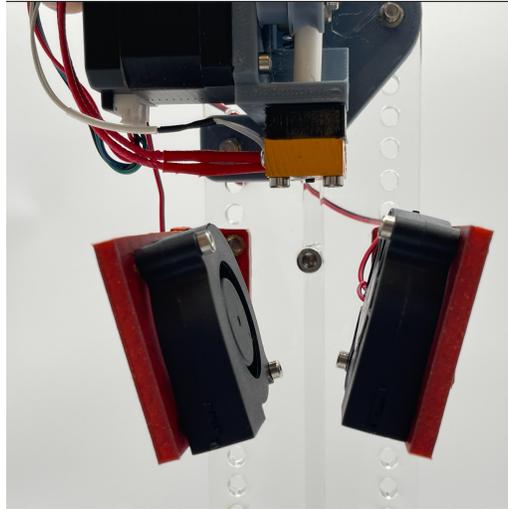


Figure 56. This is the cooling setup used for complete solidification of the fiber before the drawing process.

Another pivotal element in stabilizing the fiber was the use of a three-pulley system. The fiber was routed from the back of the first pulley to the front of the other two pulleys, creating sufficient tension that ensured the fiber's stability during the drawing process. This stability not only benefited visual inspection but also enhanced the camera's ability to capture high-resolution images and videos, leading to improved image processing results.

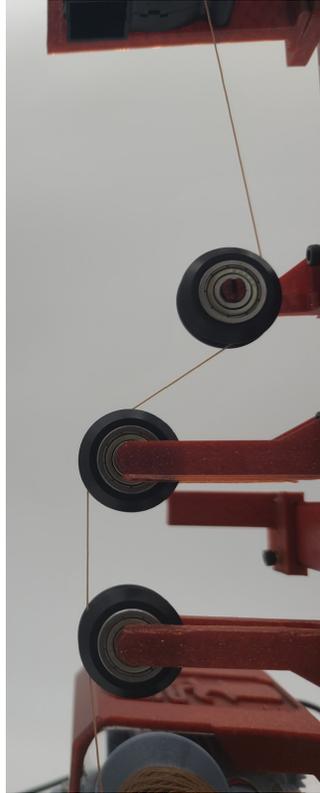


Figure 57. This is the three-pulley system that aids in the guidance and stability of the fiber during the drawing process.

Furthermore, the choice of an appropriate back screen and fiber color for the fiber played a crucial role in the image processing process. Various color options, including white, blue, and black screens, were tested to identify the most suitable option for different fiber colors. Ultimately, darker screens were found to provide the best contrast and reduced light reflections, particularly for transparent and white-colored preforms.



Figure 58. This shows the different background that was used as background for the microscope.

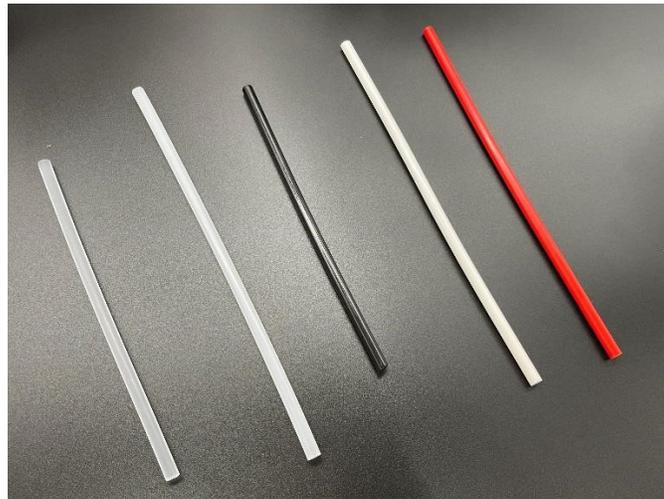


Figure 59. This shows the different preforms that were used for the fiber production.

Despite considering the groove of the pulley as a potential background for the camera's image processing, the high resolution of the camera caused issues due to capturing multiple machine lines during manufacturing and internal structures, leading to difficulties in diameter assessment.



Figure 60. This shows the set up for attaining images with eh groove of the pulley as a potential background for image processing.

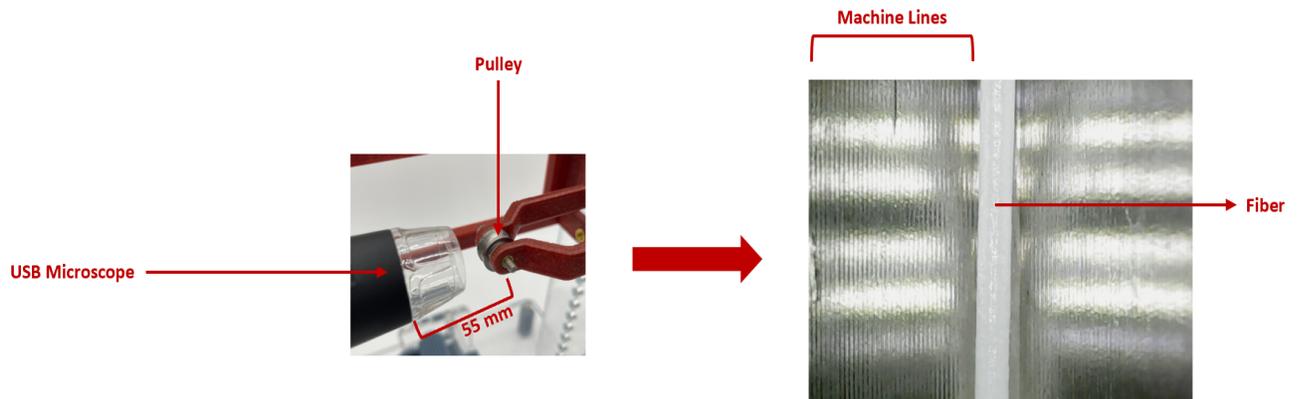


Figure 61. This shows the fiber in the middle and the multiple machine lines which makes it difficult to implement edge detection on the fiber.

This was similar with the dark back screen, however for better images, the dark back screen was utilized, placing the board out of focus from the camera, resulting in clear and distinct fiber edges being captured by the camera during image processing.

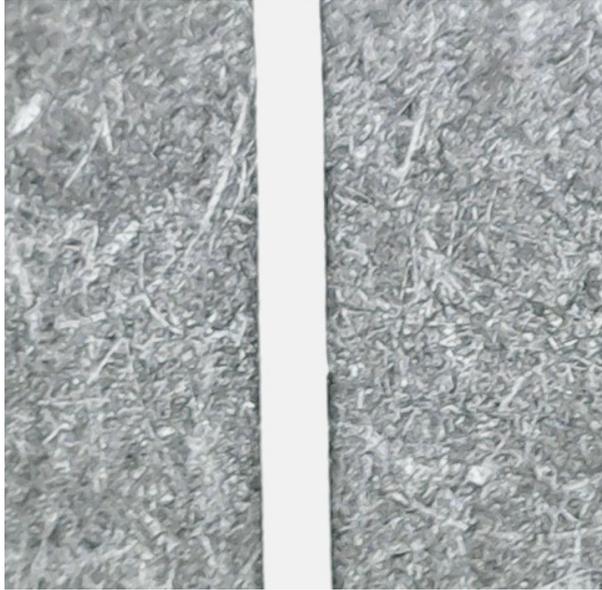


Figure 62. This is the images captured by the microscope when the dark back screen is close to the fiber showing more internal structures of the board and making edge detection difficult.

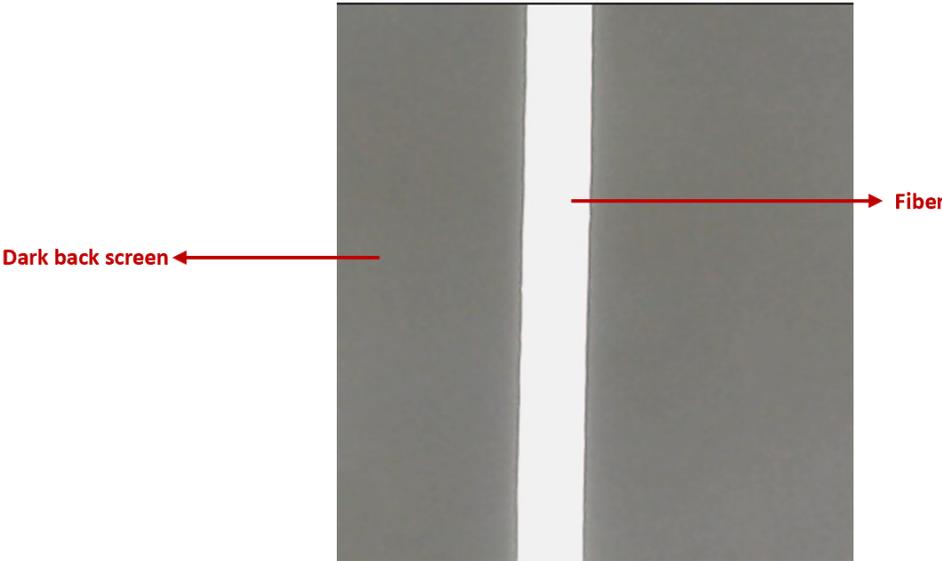


Figure 63. This is the images captured by the microscope when the dark back screen placed out of focus to the fiber to see the edges of the fiber better.

Through trial and error, various parameters such as camera distance, resolution, and pulley spacing were optimized to ensure the stability of the fiber and improve the overall system performance. Wenhao Xu's comprehensive design and fully integrated pulley and camera support system

enabled flexible movements of both the pulleys and the camera's back screen, allowing precise positioning for effective fiber diameter readings.

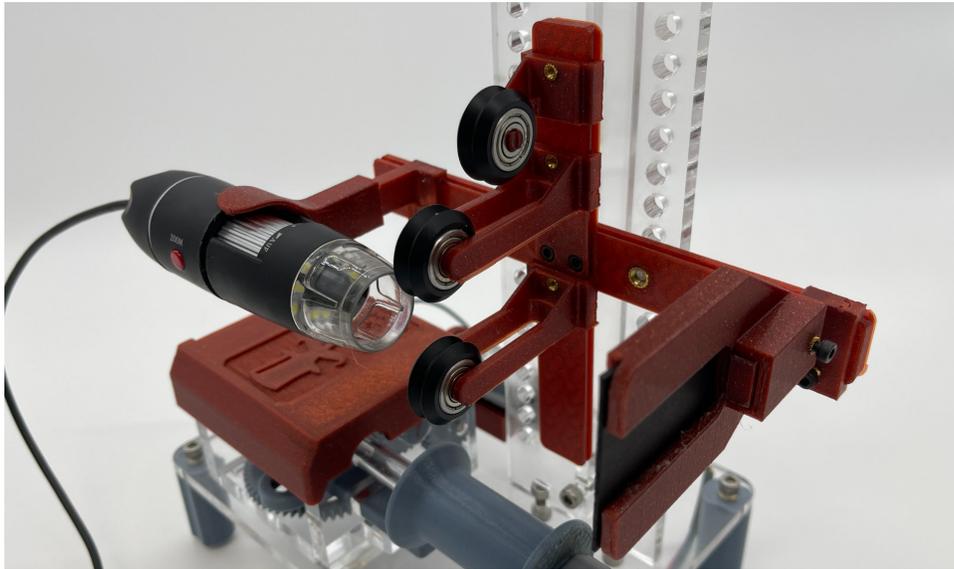


Figure 64. This shows the whole camera setup of the microscope to aid in the quality images for image processing.

For further details on the design and implementation of the FrED system, including technical specifications, and comprehensive results, refer to Wenhao Xu's thesis paper.

5.3 Image processing for pictures:

The image processing of the fiber was approached as an independent and critical component separate from the FrED system to ensure its proper functioning. The processing was conducted on Google Code lab, utilizing Google Drive as the storage repository for the images and videos used in the development of the code. The primary objective of the image processing algorithm was to perform edge detection of the fiber from the acquired images or frames.

Edge detection, in the context of image processing, refers to the identification of boundaries or edges of objects within an image. It involves analyzing pixel intensity variations to determine regions where sharp changes occur, indicating the presence of object edges. This technique plays a fundamental role in detecting the contours and outlines of the fiber, enabling precise measurements and analysis.

The image processing code was initially tailored for processing static images of the fiber. The primary focus was to ensure that the code effectively captured and delineated the edges of the fiber in each frame. By achieving this, the algorithm would provide accurate and reliable data for subsequent analyses.

The code's structure can be broken down into several key steps, which are the full code will be found in Appendix. It involved pre-processing the images, applying edge detection techniques, and post-processing the results to refine the detected edges and remove any noise or artifacts.

It's important to note that the code developed for static images served as the foundation for subsequent adaptations to process recorded videos and real-time video feeds. This modularity and scalability allowed for a cohesive approach to image processing across different data sources, ensuring consistency and accuracy in the analysis.

To begin with is to make sure the right libraries for the code to function properly. The code imports the necessary libraries: `cv2` for computer vision operations, `numpy` for numerical processing, and specific functions from `google.colab` for displaying images.

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from google.colab import drive
```

Figure 65. This shows the line of code to import the libraries needed for the computer vision for pictures.

After importing the requisite libraries, it is imperative to mount Google Drive to enable access to the image file situated in a designated directory. Subsequently, the code proceeds to read the input image from Google Drive and presents it visually using the `cv2_imshow` function, ensuring that the image under processing possesses the desired resolution. Following successful execution, the display function can be omitted to enhance processing efficiency.

```
# Mount Google Drive
drive.mount('/content/drive')

# Read image from Google Drive
img = cv2.imread('/content/drive/MyDrive/2.THG FrED/Fiber 6.15.2023/Big Roller Far.jpg')
cv2.imshow(img)
```

Figure 66. This shows the line of code for mounting the google drive and displaying the images captured by the microscope.

After the image is read into the algorithm, the next crucial step is to prepare the image for edge detection. This is achieved by converting the image to grayscale, resulting in a single-channel image where each pixel represents the intensity or brightness of the corresponding pixel in the original color image. Grayscale conversion simplifies image processing, as it removes color information, focusing solely on the intensity variations within the image. This ensures that the subsequent edge detection algorithm operates efficiently and accurately.

The Canny edge detection algorithm, implemented using the `cv2.Canny` function, is then applied to the grayscale image. The Canny edge detector is a well-known edge detection technique that identifies areas of significant intensity gradients within the image. These areas usually correspond to the boundaries or edges of objects present in the image. During the Canny edge detection process, the algorithm applies a series of steps, including Gaussian blurring to reduce noise, gradient calculation to determine pixel intensity variations, non-maximum suppression to thin out detected edges, and hysteresis thresholding to identify strong and weak edges. By leveraging the Canny edge detection algorithm, the code successfully identifies and highlights the edges of the fiber within the image.

With the edges of the fiber heightened, The Hough Line Transform (`cv2.HoughLines`) is utilized to detect straight lines in the edge-detected image. The Hough Line Transform is a technique used to detect straight lines in an image where the previously obtained edge-detected image. By applying this transformation, the Hough Line Transform is able to detect lines as points of intersection in polar space. The parameters used in the Hough Line Transform function (`cv2.HoughLines`) include the edge-detected image, the resolution of ρ (distance from the origin to the line) and θ (angle of the line), and a threshold value. The threshold determines the minimum

number of votes required for a line to be considered a detected line. Following successful execution, the display function can be omitted to enhance processing efficiency.

```
# Convert image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Apply Canny edge detection
edges = cv2.Canny(gray, 20, 200, apertureSize=3)

# Apply Hough Line Transform
lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)
print(lines)
```

Figure 67. This shows the lines of code used to implement the gray scale, canny edge and the hough lines functions.

Once the Hough Line Transform is performed, the code extracts the detected lines' information which corresponds to the straight edges detected in the original image. With this information, the code proceeds to draw the detected lines on the original image using `cv2.line`. Each detected line is visualized as a red line on the image, overlaying the original fiber image. This provides a visual representation of the straight edges detected through the Hough Line Transform. This step helps to verify the accuracy of the line detection process and allows for a visual assessment of the effectiveness of the image processing algorithm in identifying the fiber's edges. However, following successful execution, the display function can be omitted to enhance processing efficiency.

```
# Draw detected lines on the original image
for line in lines:
    rho, theta = line[0]
    a = np.cos(theta)
    b = np.sin(theta)
    x0 = a * rho
    y0 = b * rho
    x1 = int(x0 + 1000 * (-b))
    y1 = int(y0 + 1000 * (a))
    x2 = int(x0 - 1000 * (-b))
    y2 = int(y0 - 1000 * (a))
    cv2.line(img, (x1, y1), (x2, y2), (0, 0, 255), 2)
cv2.imshow('img')
```

Figure 68. This shows the line of codes that is used to detect the lines on the original image.

After drawing the detected lines on the original image, the code proceeds to calculate the distance between these two lines. These lines correspond to the edges of the fiber that were successfully identified through the Hough Line Transform and filtering process. This distance measurement directly corresponds to the width of the wire or fiber in the image, expressed in pixels. Having calculated the distance between the two detected lines, the code proceeds to display the final result. The original image, with the detected lines drawn as red lines, is displayed using the `cv2_imshow` function. In addition to the visual representation, the code prints the calculated width of the wire or fiber in pixels and millimeters as the output by multiplying with a calibrated multiplier which is attained by finding known diameter of a similar size of the diameter, which in this case was the diameter of a fishing line. This output serves as a value that is going to be used for the feedback control of the diameter of the fiber.

```
# Calculate the distance between the two detected lines
print (line)
distance = abs(line1[0] - line1[1])

# Display the result
cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()

print("Width of wire:", distance, "pixels")
print("Width of wire:", distance*0.00732142857, "mm")
```

Figure 69. This shows the line of code to calculate the diameter with and print out the value.

An additional refinement in the image processing methodology involved considering potential edge cases where the acquired images or frames might be captured at specific angles. This proactive approach aimed to address any potential loopholes or biases in the code's performance. To evaluate the algorithm's robustness, images obtained from Google Drive were intentionally tilted manually, for instance, at an angle of 22 degrees, to simulate non-vertical or non-horizontal frames. The rotation process is facilitated using the `cv2.getRotationMatrix2D` function, which calculates the transformation matrix required for the rotation. After obtaining the transformation

matrix, the actual rotation is performed using `cv2.warpAffine`, which applies the calculated matrix to the image.

```
# Define the rotation angle in degrees
angle = 22

# Calculate the image center
height, width = img.shape[:2]
center = (width // 2, height // 2)

# Perform the rotation
rotation_matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
img = cv2.warpAffine(img, rotation_matrix, (width, height))
cv2.imshow('img')
```

Figure 70. This is the line of code to tackle scenarios where the image being process is tilted.

It was identified with the angled frame, there were more lines detected after applying the Hough Line Transform to detect straight lines in the edge-detected image. Due to that, the code below proceeds to filter the detected lines to identify the two lines that correspond to the edges of the fiber. To achieve this, the code compares the angles (θ values) of the detected lines with the specified rotation angle (22 degrees in this case). Since the image was rotated to align the fiber edges horizontally, the two relevant lines should have angles close to 90 degrees (vertical lines) in the original, non-rotated image. The code iterates through the detected lines and checks their θ values. If a detected line's angle is approximately equal to 90 degrees (within a certain tolerance), it is considered as one of the two relevant lines. This code is included at the end of the 'for' loop to filter all the unwanted lines before storage of the needed two lines is performed.

```
if str(round(theta, 5)) != (str(round((angle*np.pi)/180, 5))):
    if str(round(theta, 5)) != str(round(np.pi/2 - ((angle*np.pi)/180), 5)):
        if str(round(theta, 5)) != str(round((angle*np.pi)/180 + np.pi/2, 5)):
            if str(round(theta, 5)) != str(round(np.pi - ((angle*np.pi)/180), 5)):
                if str(round(theta, 5)) != (str(round(np.pi + ((angle*np.pi)/180), 5))):
                    if str(round(theta, 5)) != str(round(np.pi/2*3 - ((angle*np.pi)/180), 5)):
                        if str(round(theta, 5)) != str(round(np.pi/2*3 + ((angle*np.pi)/180), 5)):
                            if str(round(theta, 5)) != str(round(np.pi*2 - ((angle*np.pi)/180), 5)):
                                line1.append(rho)
```

Figure 71. This shows the code for the lines to attain the two lines for the fiber.

This practical validation ensured that the image processing code remained effective and reliable across a wide range of scenarios, providing increased confidence in its performance and reducing the risk of potential errors as seen in the resultant output images for the frame below.

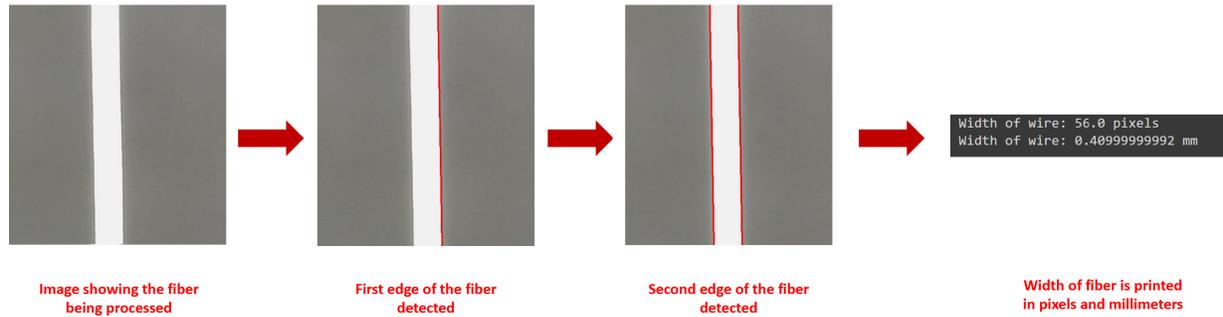


Figure 72. This is the final display of the image after the code has been executed to find the edges of the fiber and print the output value.

The image below vividly illustrates the vital importance of ensuring that no additional lines are visible to the microscope. Any extraneous lines could be mistakenly interpreted as the edges of the fiber, leading to incorrect outputs. This issue is demonstrated below, where the groove of the pulley serves as an example of such potential misinterpretation.

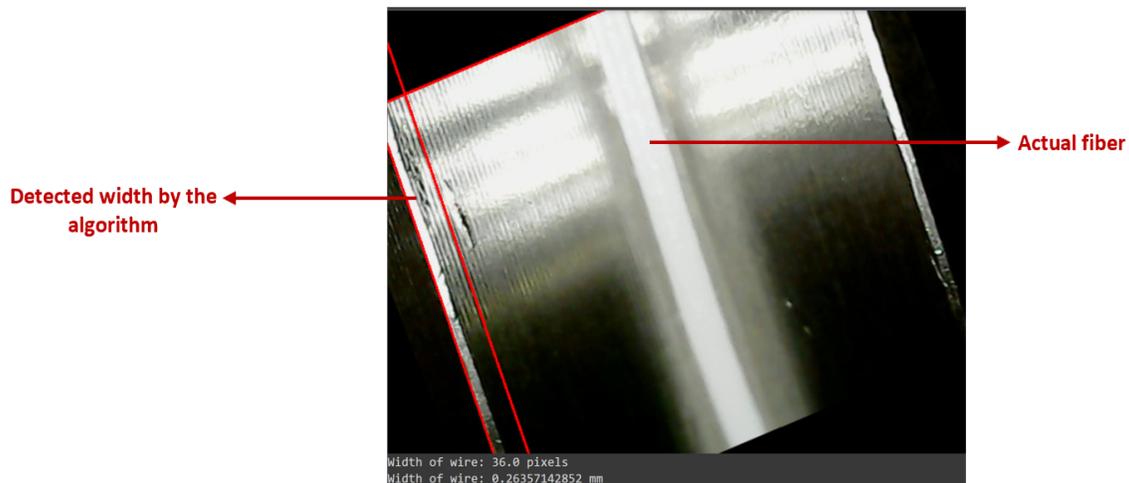


Figure 73. This shows the edge case where the groove of the pulley is used as a back screen and the image is tilted.

5.4 Image processing for recorded video:

Upon successfully obtaining the image and frame values, the subsequent endeavor involved extending the code's capabilities to process recorded videos. This crucial advancement aimed to enable image processing for dynamic scenarios, specifically capturing a moving fiber. As previously stated, the core elements of the code remained consistent, requiring only minor adjustments to accommodate video data. The code continues to rely on the same libraries and Google Drive mounting procedure, ensuring a seamless transition between image and video processing functionalities. To facilitate video processing, the code utilizes the `cv2.VideoCapture` function, that serves as a gateway to the video data, allowing the code to retrieve individual frames and pertinent video properties. It then initializes two lists, `line_distances` and `line_final`, to store the distance values calculated between pairs of detected lines in each frame.

```
# Mount Google Drive
drive.mount('/content/drive')

# Read video from Google Drive
video_path = '/content/drive/MyDrive/2.THG FrED/6.28.2023/WIN_20230628_17_15_09_Pro.mp4'
cap = cv2.VideoCapture(video_path)

# Define a list to store line distances
line_distances = []
line_og = []
line_final = []
```

Figure 74. This is the lines of code for mounting, retrieving frames and then initializing variables.

The code iterates through each frame of the video with a while loop, ensuring a sequential traversal through the video frames, and executes the next steps for each frame. For each iteration, the code employs the `cap.read()` function to extract the next frame from the video stream. The function returns two values: a boolean variable indicating whether the frame was successfully read (`ret`) and the actual frame data (`frame`). The code proceeds to the next steps only if the frame is successfully read (i.e., `ret` is True), ensuring the entire video is traversed until the last frame is reached. The frame is converted to grayscale using `cv2.cvtColor`, performs edge detection using the Canny edge detection algorithm (`cv2.Canny`), and use the Hough Line Transform

(cv2.HoughLines) to detect straight lines in the edge-detected frame which is same for the image processing code for the pictures. Detected lines are stored in the lines variable.

```
while True:
    # Read each frame of the video
    ret, frame = cap.read()

    if not ret:
        break # Break the loop if there are no more frames

    #cv2_imshow(frame)

    # Convert frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply histogram equalization to enhance contrast
    #frame_equalized = cv2.equalizeHist(gray)

    # Apply Canny edge detection
    edges = cv2.Canny(frame, 50, 150, apertureSize=3)
    #print(edges)

    # Apply Hough Line Transform
    lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)
```

Figure 75. This the initial lines of codes in the while loop to get the lines for the edges of the fiber with gray scale, canny detection and hough lines.

With the detected lines, the code checks if 'lines' is not None and contains at least two detected lines. This is to help filter which frames have the lines that have been able to detect both edges of the fiber. It then proceeds to calculate the distances between the two detected lines using the Hough Line Transform results. The distances are stored in the line_distances list. The script iterates through the detected lines, calculates the distance between each pair of lines, and appends the result to line_final.

```

if lines is None or len(lines) < 2:
    continue

line_distances = []

for line in lines:
    rho, theta = line[0]
    #print (rho, theta, line)
    a = np.cos(theta)
    b = np.sin(theta)
    x0 = a * rho
    y0 = b * rho
    x1 = int(x0 + 1000 * (-b))
    y1 = int(y0 + 1000 * (a))
    x2 = int(x0 - 1000 * (-b))
    y2 = int(y0 - 1000 * (a))
    cv2.line(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)

    line_distances.append(rho)
    print ("line distance = ", line_distances)
line_final.append(line_distances)

```

Figure 76. This is the line of code to filter and detect the two lines that are the edges of the fiber.

The script displays each frame with the detected lines using `cv2_imshow`. It continuously processes the frames until the user presses the 'q' key. After processing all frames, the code calculates the final distance between the pairs of lines stored in `line_final`. It prints the distance values in pixels and the corresponding width of the fiber in millimeters for each pair of lines.

```

# Display the frame with detected lines
cv2.imshow(frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break # Break the loop if 'q' is pressed

cap.release()
cv2.destroyAllWindows()

# Calculate the distance between the two detected lines
distance_final = []
for j in line_final:
    print(j)
    #print (line_final)
    distance = abs(j[0] - j[1])
    #print (line_distances[0], " ", line_distances[1])
    distance_final.append(distance)

for index, distance in enumerate(distance_final):
    print("Distance", index+1, ":", distance, "pixels")
    print("Width of wire", index+1, ":", distance * 0.00732142857, "mm")

```

Figure 77. This is the line of code to display, calculate the width of the fiber and print out the output value.

An additional feature to this code is that extracts and prints vital video properties, including the total number of frames, the frames per second (FPS), and the video's overall duration in seconds. This insightful output provided crucial information about how many frames could be captured for a specific duration of time. In this the testing of the code, it was able to attain about 97 frames in the 3 seconds video which approximately translate to 30fps.

```

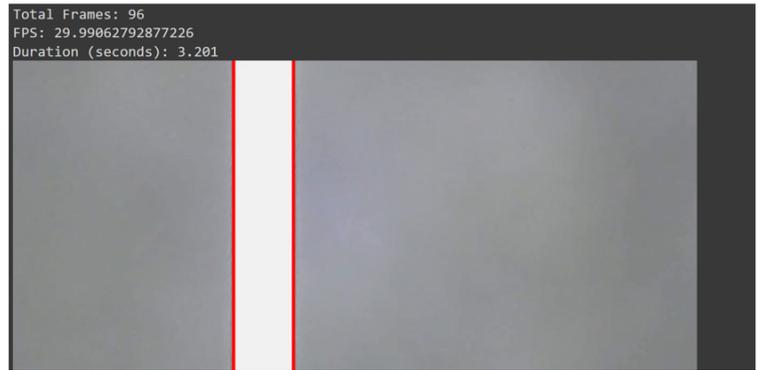
# Get video properties
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)
duration = frame_count / fps

# Print video properties
print("Total Frames:", frame_count)
print("FPS:", fps)
print("Duration (seconds):", duration)

```

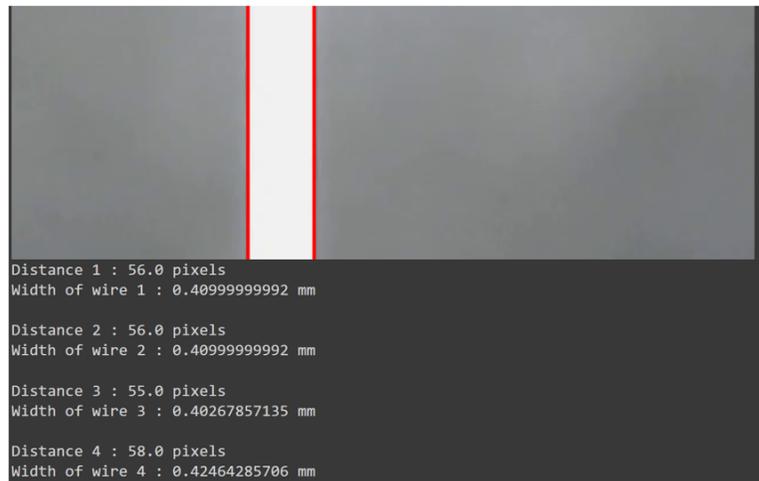
Figure 78. This is the line of code to display the video properties.

This practical validation ensured that the image processing code remained effective and reliable for a recorded video. The next step was to make sur that the code can work on live video to produce real time data for the feedback control of the fiber. The images below show some of the successes achieved using the code for the recorded video.



•
•
•

→ Show the other frames with the detected lines



•
•
•

→ Prints the rest of the fiber width in pixels and millimeters

Figure 79. This is the final display of the image after the code has been executed to find the edges of the fiber and print the output value.

There were instances highlighting that the code might fail to detect any lines under certain conditions, such as when the video is excessively blurry, the fiber experiences excessive shaking, or the background screen displays multiple lines. These scenarios were crucial to address, ensuring that the calculated fiber width does not originate from flawed frames within the video recording.

```
Mounted at /content/drive
Total Frames: 159
FPS: 19.20224919930244
Duration (seconds): 8.28028
line = None
line = None
line = None
line = None
line = [[[3.180000e+02 5.235988e-02]]]
line = None
```

Figure 80. This shows the output of an edges case of when the video is too blurry, the fiber shaking too much or when the back screen shows multiple lines.

5.5 Image processing for live video:

The successful adaptation of the code for recorded video processing was vital for the pursuit for real-time video for feedback control of fiber diameter. However, to achieve operability with live video input and to relay data for feedback control to the Teensy Arduino microcontroller, further enhancements and considerations were imperative. The Teensy Arduino lacks the native capability for image processing due to inherent hardware constraints. As a result, image processing tasks were exclusively performed on Google Colab.

In addition to that, the Teensy Arduino lacks direct USB connectivity, alternative communication mechanisms were adopted to bridge this gap. Specifically, a serial connection and the baud rate was established, enabling data transmission between the computer and the microcontroller. The Teensy Arduino was now primed to receive and interpret the diameter values inferred from the image processing algorithm.

The implementation for live video processing closely resembles that of the recorded video but to be able to attain the serial connection for the data gained from Google Colab to the Teensy Arduino, necessary package pyserial is needed using the "!pip install" commands, and add to the imports. With the imports, the code establishes a serial connection with the Teensy Arduino using the serial.Serial() function. The specific COM port and baud rate are configured accordingly.

```
!pip install pyserial

import cv2
import numpy as np
import serial
from google.colab.patches import cv2_imshow
from google.colab import drive

# Connect to the Teensy Arduino
ser = serial.Serial("COM6", 9600) # Replace <serial_port> and <baud_rate> with your Arduino's port and baud rate
```

Figure 81. This is the line of code help provide the serial connection between the Teensy Arduino and the device being used.

After the serial connection, the code follows the recorded video code, thus from the capturing the video, reading the frames and using the while loop for image processing to calculate the diameter of the fiber as seen in the figures 75-81. An addition to the while loop is that the calculated width of the wire (in millimeters) is sent to the Teensy Arduino using ser.write() after being encoded as a string. This facilitates the transmission of the width data to the Arduino for further processing or control actions.

```
# Calculate the distance between the two detected lines
distance = abs(line_distances[0] - line_distances[1])
width_of_wire_mm = distance * 0.00732142857
print("Width of wire :", width_of_wire_mm, "mm")
print() # Print an empty line to create space between each index

# Send the width of the wire to the Teensy Arduino
ser.write(str(width_of_wire_mm).encode())
```

Figure 82. This is the line of code that calculates and sends the values to the Teensy Arduino in real time.

To monitor that the code was working successfully, the processed frame, with detected lines drawn, is displayed using `cv2.imshow()` similar in the figure 66.

5.6 Image processing on Raspberry Pi Camera model 3:

Upon achieving successful integration of the microscope with real-time video processing of the fiber drawing process, the camera demonstrated commendable precision in attaining accurate data values for the fiber. The comparison of the camera-acquired data with those obtained from the laser micrometer was very similar to validating the precision of the imaging processing algorithm.

However, during the calculation of the frame rate, it was ascertained that the microscope-operated camera was limited to a modest 30 frames per second (fps). This limitation raised concerns about the camera's suitability for producing enough data for feedback control and the potential utilization for machine learning applications in the long term. A decision was made to explore alternative options, seeking a low-cost camera with a higher frame rate camera to bolster the system's capabilities, enabling real-time data acquisition at a more rapid pace. A higher frame rate equates to a greater volume of captured data, which is pivotal for fine-tuned feedback control algorithms and data-driven machine learning models. The table below shows the different cameras that were considered with more details of the camera search in Wenhao thesis.

Table 1. This is the different cameras and the important properties consideration for image processing of the fiber.

Name	Effective Video Resolution (Pixels)	FPS	Minimum Object Distance (cm)	Shutter Type	Focus Type
Raspberry Pi High-Quality Camera (HQ) 12.3 MP	2028 × 1520p 2028 × 1080p 1332 × 990p	30 56 120	20	Rolling	Manual
Arducam OV5647 V1 5 MP	1080p 720p 480p	30 60 90	8	Rolling	Fixed
Raspberry Pi Camera 3 SC0872 11.9 MP	2304 × 1296p 1536 × 864p	30 HDR, 56	10	Rolling	Motorized
Arducam IMX519 16 MP	1080p 720p	30 60	10	Rolling	Motorized
Arducam Auto Focus OV5647 5MP	1080p 720p 480p	30 60 90	8	Rolling	Motorized
Arducam OV9281 Monochrome Global Shutter Camera	1280×800 1280×720 640×480	120 144 260	50	Global	Fixed
CAM-MIPI9281RAW-V2	1280×800 1280×720 640×480	144 144 253	-	Global	Manual

Of the various cameras, the Raspberry Pi Camera 3 SC0872 11.9 MP emerges as the optimal choice due to its exceptional balance between its resolution and a substantial frame rate of 120. Despite being of the rolling shutter type, a physical tilt of the camera sideways allows for the fiber to be captured horizontally through the lens, which helps mitigate the risk of image being distorted or skewed [13]. Moreover, the camera's motorized focus rather than fixed provides the ability to finely adjust the camera's focus resolution enhances the quality of the captured images.

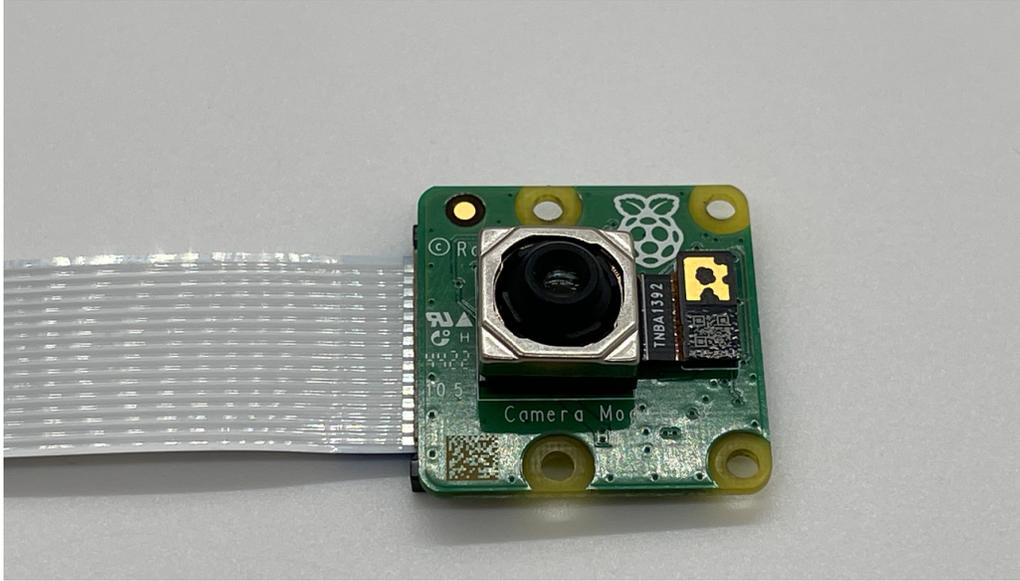


Figure 83. This is the Raspberry Pi Camera 3 SC0872 11.9 MP that is used in image processing to attain 120 fps.

Adopting the Raspberry Pi Camera Module 3, it was imperative to employ a more powerful Raspberry Pi microprocessor than the previously employed Teensy microcontroller due to its capabilities of image processing. Moreover, the shift from a USB-connected camera module to a ribbon connection on the Raspberry Pi instigated a complete revamp of the communication protocols and libraries employed which entailed a comprehensive reevaluation of the codebase. This included moving away from the Teensy's C++ coding environment to the code that aligns with the Raspberry Pi's Python-based architecture.

In addition to that, a distinctive aspect distinguishing the Raspberry Pi from the USB camera is the absence of front lighting. Consequently, a white LED backlight board was used to serve as the back screen for the Raspberry Pi camera. To enhance the visibility and clarity of the fiber's edges during image capture, the back lighting system was used to emit illumination from the rear, effectively illuminating the fiber against the contrasting background making the edges more discernible and facilitating precise image processing.

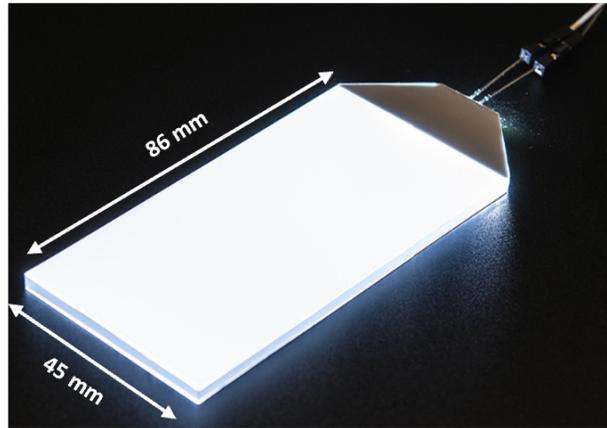


Figure 84. This is the White LED Backlight Module that is used to provide contrast with the fiber for better image quality.

Moreover, the integration of the Raspberry Pi Camera into the FrED system necessitated the development of a dedicated camera support mechanism. The design entailed a redesign of the existing setup employed for the USB microscope with its modulation system to go to and fro of the fiber and the back of the FrED system. The major change was to change the part that holds the camera to a board that serves as a housing to the Raspberry Pi where it is screwed to the board from the front.

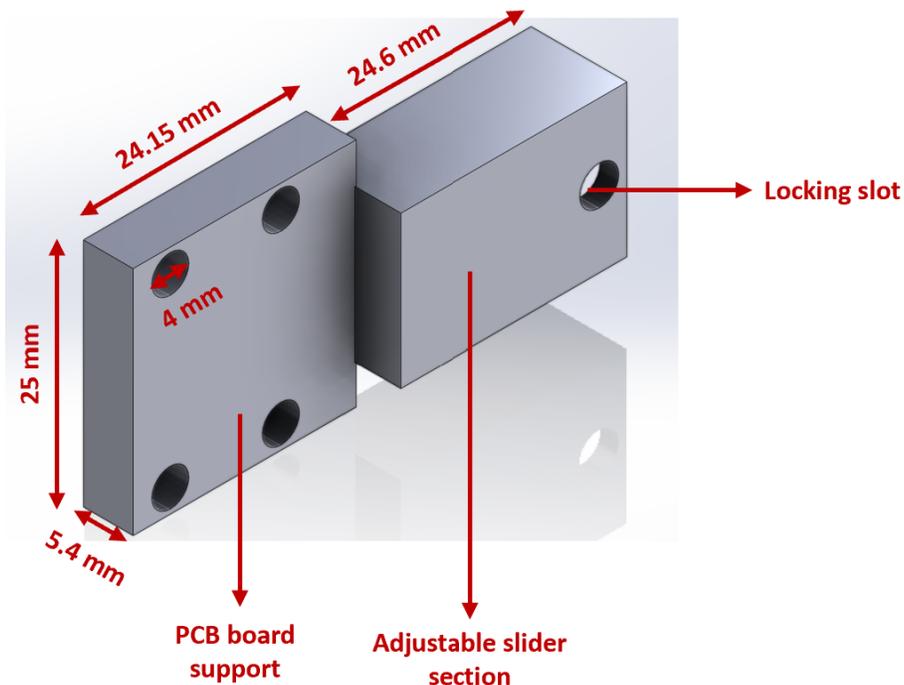


Figure 85. This is the first prototype to house and attach the Raspberry pi to the FrED system.

During the initial design iteration, certain considerations were overlooked which was recognizing the importance of safeguarding the delicate front-end components. In the second design iteration, a 2mm protrusion was incorporated at the rear of the camera support mechanism. By positioning the Raspberry Pi at the back of the system, the protrusion served as a protective buffer, shielding the camera's sensitive elements from any pressure or damage. Moreover, the design featured a square aperture aligned with the camera's lens, allowing the lens to subtly protrude to the board's surface helping to view of the fiber.

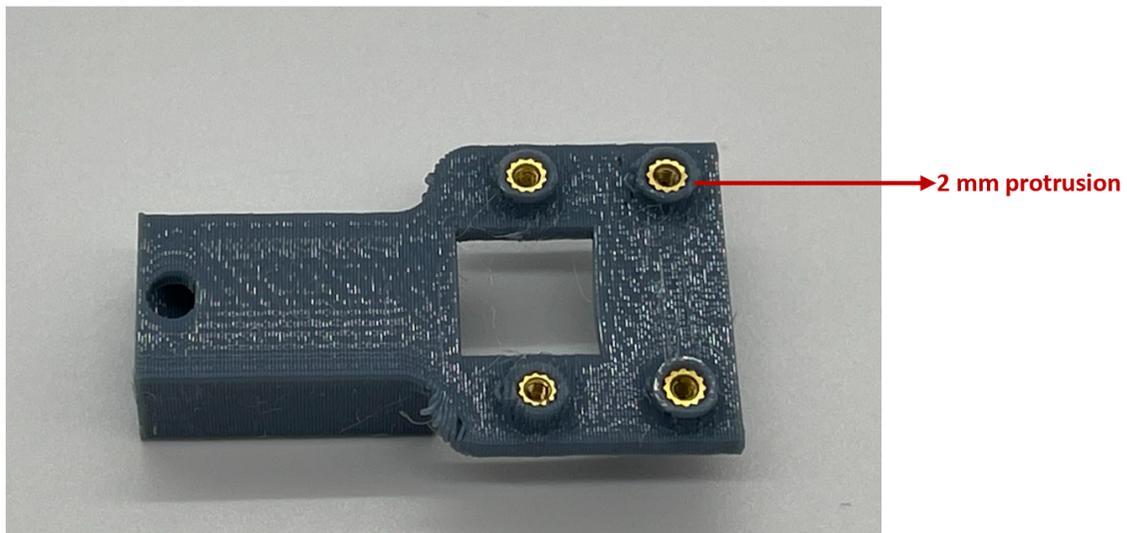


Figure 86. This is the second prototype to house and attach the Raspberry pi to the FrED system.

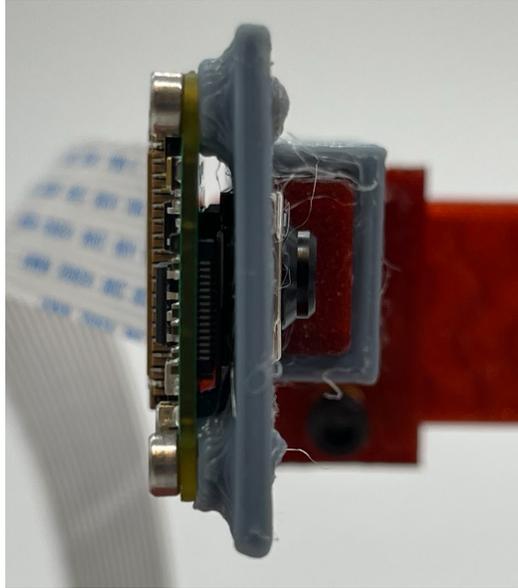


Figure 87. This shows protrusion that serves as a protective buffer, shielding the camera's sensitive elements from any pressure or damage.

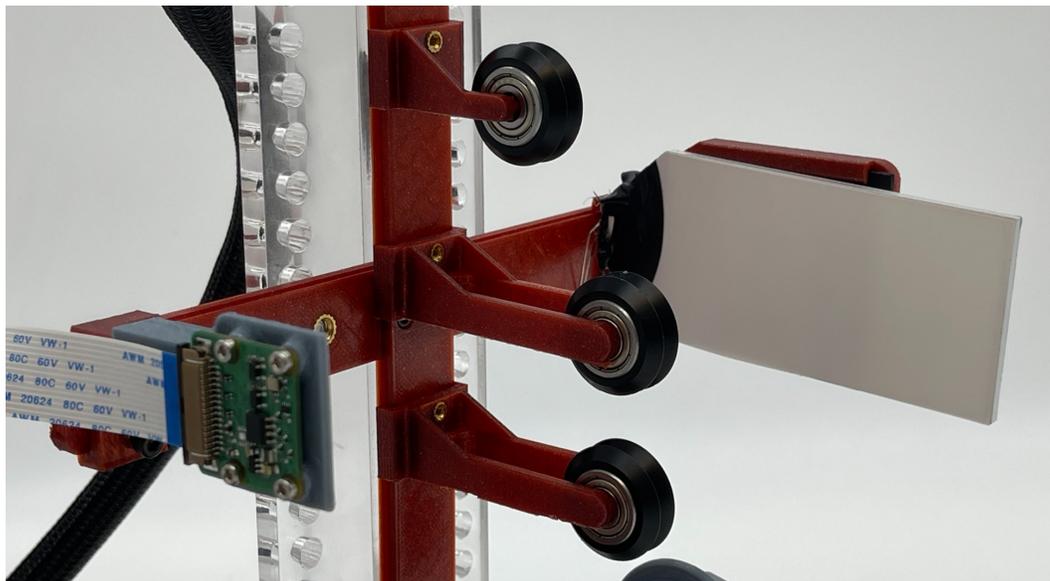


Figure 88. This shows the setup of the camera system for the Raspberry Pi camera module 3 for image processing.

With the Raspberry Pi camera setup, the next step was to utilize the "picamera2" library and adapt the code for USB functionality to work with the Raspberry Pi as well. The code was written and run on the Thonny IDE which was already embedded in the Raspberry Pi. The code begins by

importing the required libraries, including libraries for camera control (picamera2), computer vision (cv2 from OpenCV), array manipulation (numpy), visualization (matplotlib.pyplot), mathematical functions (math), clustering (KMeans from sklearn.cluster), and time manipulation (time). An instance of the Picamera2 class is then created and initialized with specific controls such as manually autofocus the lens as well as its position. The camera preview is displayed using the show_preview parameter. The window for displaying the processed image, and its position is created and adjusted on the screen. The IDE showing here is for Google Colab since it was difficult to attain screenshots from the Raspberry Pi but it is run on the Thonny IDE.

```
import picamera2
from libcamera import controls
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math
from sklearn.cluster import KMeans
import time

# Create a Picamera2 instance
cam = picamera2.Picamera2()
cam.start(show_preview=True)
cam.set_controls({"AfMode": controls.AfModeEnum.Manual, "LensPosition": 100})

# Create a named window
cv2.namedWindow("Lines", cv2.WINDOW_AUTOSIZE)
# Move the window to a specific position
cv2.moveWindow("Lines", 100, 100)
```

Figure 89. This is the line of code to import the library, make a picamera2 instance and setting the window to display processing images.

With the Raspberry Pi camera set to view the fiber, it then processes through the while similar to the code for the microscope. Within the loop, an image is captured from the ongoing camera stream. To facilitate efficient processing, the captured image is first converted from its original color representation to grayscale and then the Canny edge detection is employed as it is with the microscope code but with a more defined threshold. The outcome is a binary image where edge pixels are represented as white while the background remains black. This process essentially outlines the prominent features in the image, making it easier to identify the main lines or wires

present within it. The Hough Line Transform is applied to detect lines in the edge-detected image to attain the distance resolution (ρ), angular resolution (θ), and threshold of the line segments.

```
while True:
    # Capture an image into the stream
    image = cam.capture_array("main")

    # Apply Canny edge detection
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Canny edge detection
    low_threshold = -20
    high_threshold = 40
    edges = cv2.Canny(gray, low_threshold, high_threshold)

    # Apply Hough Line Transform
    rho = 1 # Distance resolution in pixels of the Hough grid
    theta = 0.2* np.pi / 180 # Angular resolution in radians of the Hough grid
    threshold = 120 # Minimum number of votes (intersections in Hough grid cell)

    # Run Hough on edge detected image
    # Output "lines" is an array containing endpoints of detected line segments
    lines = cv2.HoughLines(edges, rho, theta, threshold)
```

Figure 90. This is the line of code to capture the frames and employ grayscale, canny edge detection and hough line algorithm on the frames.

Still in the while loop and upon successful detection of lines in the edge-processed image, a pivotal step involves visually highlighting these lines on the original image. If lines are detected, they are drawn on the image using their mathematical representation (ρ and θ). These detected lines are also stored in an array. Following the drawing of detected lines, the code proceeds to implement a sophisticated data analysis technique known as K-Means clustering. This method facilitates the categorization of the detected lines into distinct clusters, with each cluster representing a specific pattern or subset of the data. In this context, K-Means clustering is employed to segment the detected lines' data into two distinct clusters, which align with the two primary lines present in the image and displayed in the window previously created. In essence, K-Means clustering enhances the precision and interpretability of the image processing results, yielding a clearer and more accurate depiction of the detected wire or line configurations.

```

if lines is not None:
    for i in range(0, len(lines)):
        rho = lines[i][0][0]
        theta = lines[i][0][1]
        a = math.cos(theta)
        b = math.sin(theta)
        x0 = a * rho
        y0 = b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
        cv2.line(image, pt1, pt2, (0,0,255), 3, cv2.LINE_AA)

lines_array = []
for i in range(len(lines)):
    lines_array += [[lines[i][0][0],lines[i][0][1]]]

# K-mean cluster algorithm depending on rho to end up with two lines
kmeans = KMeans(n_clusters=2, random_state=0, n_init="auto").fit(lines_array)

# Display the image
cv2.imshow("Lines", image)

```

Figure 91. This is the line of codes that draw the lines detected from the frame and cluster lines to make two distinct lines.

With the two primary lines clearly identified, the distance between these two lines is calculated. Drawing upon their polar coordinates and employing mathematical relationships involving sine and cosine functions, the code calculates the distance between the lines both in pixels and millimeters. The processed image with detected lines is displayed in the "Lines" window. The calculated diameter of the wire is printed as output. If no lines are detected, a message "No fiber is seen" is printed in order to know if there is an issue with the images that are fed to the Raspberry Pi camera. The loop continues to run until manually interrupted, and the camera is closed properly when done.

```

lines_array = []
for i in range(len(lines)):
    lines_array += [[lines[i][0][0],lines[i][0][1]]]

# K-mean cluster algorithm depending on rho to end up with two lines
kmeans = KMeans(n_clusters=2, random_state=0, n_init="auto").fit(lines_array)

# Display the image
cv2.imshow("Lines", image)

# Compute distance between the two lines
distance = abs(math.sin(kmeans.cluster_centers_[0][1])*kmeans.cluster_centers_[0][0] -
                 math.sin(kmeans.cluster_centers_[1][1])*kmeans.cluster_centers_[1][0])
width_of_wire_mm = distance * 0.0230659188788095

# Perform further processing or actions with the width_of_wire_mm value
print("Diameter : ", width_of_wire_mm,"mm")

else:
    print("No fiber is seen")

# Close the camera when done
camera.close()

```

Figure 92. This is the line of code to calculate and print the fiber diameter as an output.

With this code, an initial test involved using a stationary white cardboard sheet to ensure the code's proper functionality and its capability to accurately detect the edges of the cardboard. Once this preliminary test yielded success, the focus shifted to employing a stationary fiber to determine its diameter. Unlike the USB microscope, the Raspberry Pi lacks additional lenses that could facilitate zooming in on the fiber, leading to limitations in achieving clear edge demarcation. Another limitation was that there was a need for an extended distance between the camera and the fiber to prevent excessive blurriness, which was approximate distance of 80 mm. Unfortunately, this extended distance hindered the accurate measurement of the fiber's diameter as it diminishes the fiber's visibility and size in the image.

Consequently, a significant issue arises wherein the lines detected by the edge detection process overlap significantly with the fiber. This overlap impeded the code's ability to distinctly identify the edges of the fiber, ultimately resulting in considerable fluctuations in diameter measurements. The lines drawn on the fiber needs to be enhanced to attain precision with the image processing algorithm. Specifically, these lines need to be more refined and narrower, particularly to accommodate the minute size of the fiber that the Raspberry Pi camera captures.

6. Implementing Hardware Improvements:

Improving the FrED system involved a focus on hardware enhancements, aiming to elevate overall efficiency, streamline assembly processes, and reduce manufacturing costs. As discussed in the previous sections, comprehensive hardware improvements were implemented, spanning the cooling subsystem and three-pulley system configuration. Another of the major changes was the extrusion system and the frame of the FrED system.

6.1 Extrusion System:

6.1.1 Previous Design:

In the initial version, the extrusion system comprised two key sections: the heater block and the extruder. The development of the heating block for the FrED system involved a thorough exploration of standard 3D printer components, particularly focusing on the Creality Ender 3 hot end, which is specialized in extruding PETG polymers. To adapt this hot end for the FrED system's specific needs, necessary modifications were implemented to accommodate the 7mm diameter of the glue stick preform, surpassing the typical 1.75mm filament diameter used in standard 3D printers. This adaptation required precise machining of the heat sink and the incorporation of a steel pipe, which was cut using a band saw [6]. The utilization of a steel heat break pipe was instrumental in ensuring a robust and secure seal, complemented by the application of Marine Weld, a high-strength, heat-resistant adhesive.

The integration of the hot end with the steel pipe was accomplished using M3 screws, providing a firm and stable connection. Within the hot end, essential components such as the heater cartridge and thermistor were integrated to facilitate the controlled heating of the pre-form. The thermistor was used for accurate temperature reading to the Arduino to aid regulate the heater and prevent thermal overshoot in the heater cartridge.

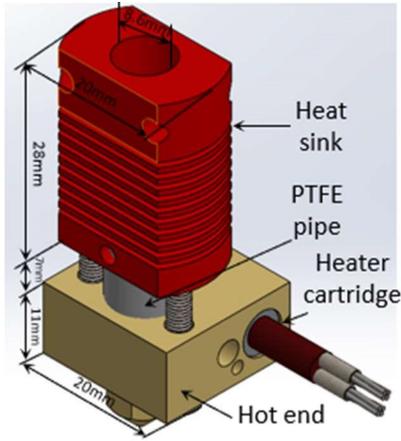


Figure 93. This is the heating area for melting the preform.

The extruder was designed by employing a 3D printed extrusion support part, featuring threaded inserts at its base to secure attachment of the hot end using M3 bolts. Positioned at the rear of the extruder was the stepper motor, with its driving gear adeptly positioned inward. To facilitate the guiding of the fiber, the lever was incorporated above the driving gear. The lever, when screwed in from the front, aligned with the driving gear, serving as a reliable guide to direct the fiber through the heat sink towards the hot end. The extrusion support part was printed using PETG, a material renowned for its high temperature resistance [6]. Moreover, the extruder design accounted for the variability in pre-form sizes, with the addition of an M5 bolt and nut mechanism, having the ability to adjust the spring compression, thereby accommodating different pre-form dimensions with ease.

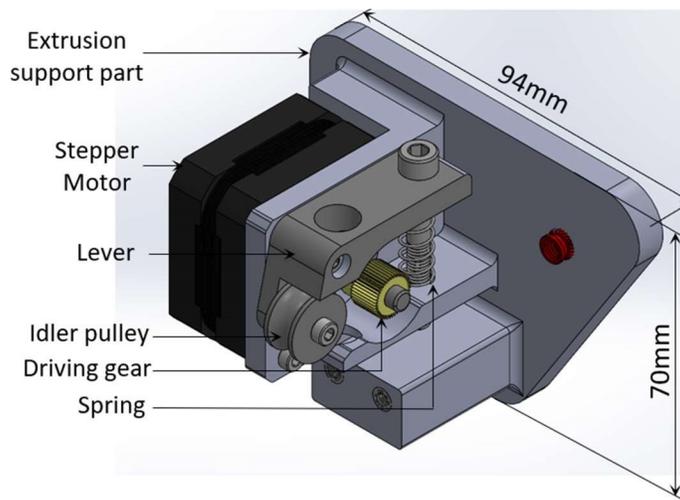
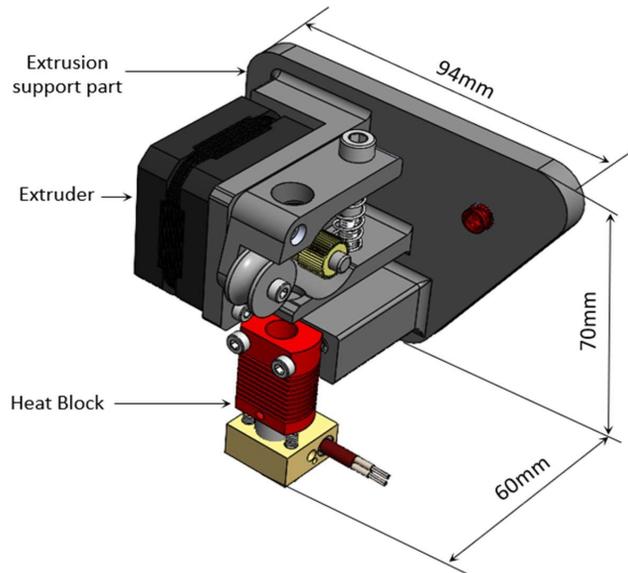


Figure 94. This is extrusion system support of the old version of the FrED system holding the stepper motor.

This design ensured the heating block and extruder worked effectively together, enabling the successful extrusion of glue stick pre-forms in the FrED system.



95. This is the whole assembly of the extrusion system of the old version of the FrED system.

The overall functionality of the old version was working, and the main issues was on the reducing parts needed to make it, reduce the number of material needed to make the sub system as well make it such a way is not hazardous to the user especially looking at how hot it can get hot and making its function better.

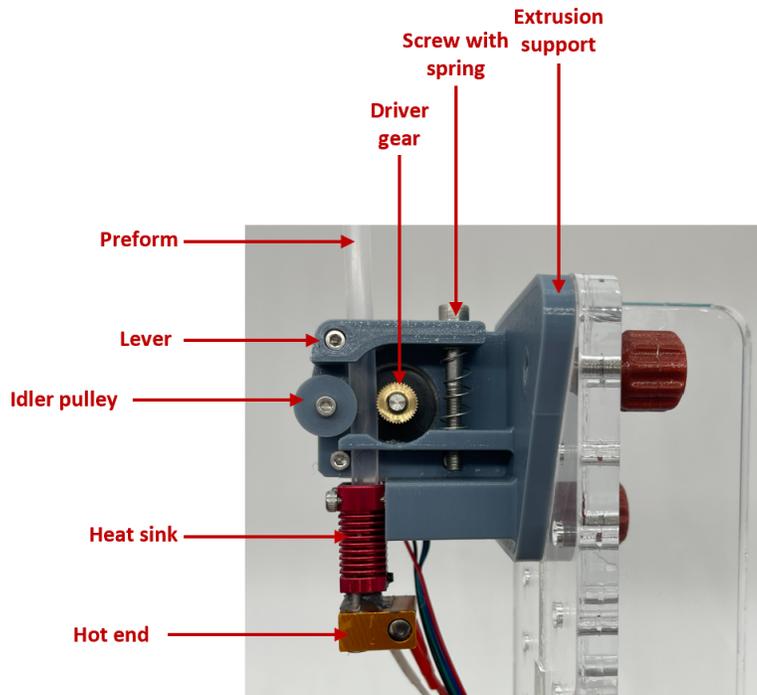


Figure 96. This shows the whole setup of the extrusion system for the old version.

6.1.2 Design Changes:

6.1.2.1 Lever:

Starting from the top, a lever was equipped with a holder designed to secure the preform as it's being placed. In the previous version, users needed to hold the preform in place for a certain amount of time until it was properly secured. To address this, a protrusion was added to the section where the preform enters. This extrusion allows the preform to rest securely as it's lowered into the heating section, eliminating the need for the user to continuously hold it.



Figure 97. This shows the lever for the old extrusion system vs the lever of the new extrusion system with the protrusion to hold the perform.

6.1.2.2 Spring and Lock nut:

Furthermore, an improvement was made with a stronger spring used to raise the lever, ensuring a tighter hold and precise alignment of the fiber. This alteration reduced the risk of misalignment during the descent of the preform into the heating zone. This upgrade also plays a key role in preventing any inadvertent slipping as the preform is pushed downward toward the heating area.



Figure 98. This shows the spring from the previous design and the new stronger spring to aid in alignment and prevent slippage of the preform.

Furthermore, there is an issue with the nut used to fasten the spring. This problem arises due to the vibrations coursing through the FrED system, with the pulsation from the stepper motor being one of causes of the vibration, which result in the spring becoming loose and causing the preform to slip when its being push down to the heating zone.

To address this challenge, a lock nut has been introduced as a solution. Unlike a regular nut, a lock nut features a unique design that incorporates a locking mechanism. The locking mechanism of a lock nut is specifically designed to provide superior fastening and resistance against loosening caused by vibrations, external forces, or other factors. This mechanism typically includes a series of grooves or serrations on the inner surface of the nut [14] . These grooves engage with the threads of the bolt or screw onto which the lock nut is being tightened.

As the lock nut is threaded onto the bolt, the grooves create additional friction between the nut and the bolt's threads. This increased friction generates a higher level of resistance to rotational

movement, effectively countering any attempts of the nut to loosen due to external forces. The engagement of the grooves with the threads creates a sort of locking effect, hence the name. The low-strength steel nylon-insert locknut that was used featured a nylon insert that further enhances the locking mechanism. This insert creates an interference fit with the threads, adding more resistance against rotational movement, and preventing the nut from backing off unintentionally. This enhanced security ensures that the spring remains in place, upholding proper preform alignment and preventing any unintended slipping during the preform heating phase [14].

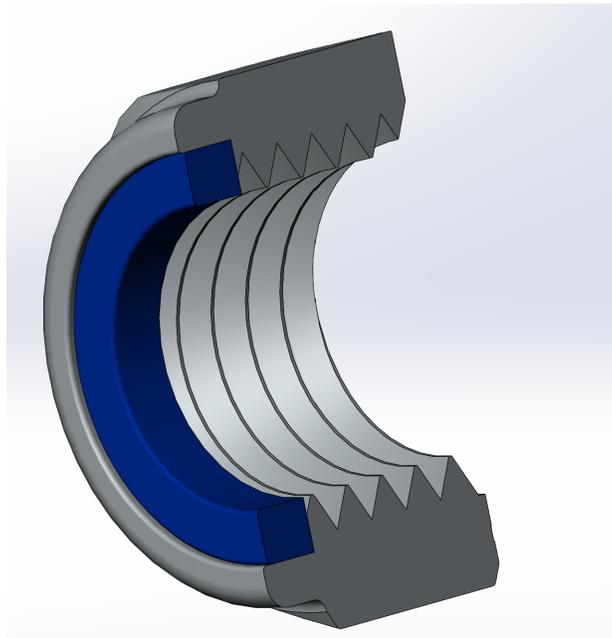


Figure 99. Cross sectional view of the lock nut showing the serrations and the nylon insert (blue piece).



Figure 100. This is the low-strength steel nylon-insert locknut that was used to ensure that the spring remains in place for proper alignment of the preform.

6.1.2.3 PC alloy heat block:

Another significant alteration to the extrusion system involved the removal of the heat sink that was connected to the hot end. The process of preparing the heat block section involved potentially hazardous manufacturing steps, such as enlarging the heat sink by drilling through it. This procedure also included pushing a rod through the heat sink, which was created by band sawing—a step that proved to be potentially unsafe during implementation. Additionally, the heat sink itself would become extremely hot during the preform heating process, posing a risk if a user were to touch it. Given its prominent location and size within the heating area of the extrusion system, accidental contact with the heat sink presented a genuine safety concern.

To address these issues, the heat sink was substituted with a 3D printed block made from a material known as PC alloy. This choice of material was particularly suitable due to its durability, heat resistance, and mechanical performance in varying environments, rendering it an ideal replacement for the heat sink. Rigorous testing confirmed its capability to withstand elevated temperatures, demonstrating resilience even at temperatures as high as 150 degrees Celsius—1.5 times higher than the average operational temperature of the hot end. Notably, the 3D printed heater block featured a protrusion of approximately 1mm, strategically designed to compress into the hole of the hot end, through which the preform melted through.

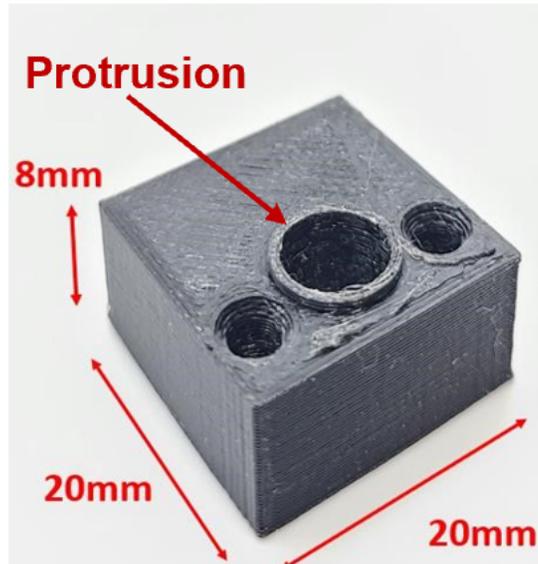


Figure 101. This is the PC alloy heater block that is used in place of the heat sink to avoid hazardous scenarios and reduce cost.

This protrusion serves a dual purpose: it effectively seals the junction between the heater block and the hot end, preventing any leakage of the preform material. As the hot end reaches its operational temperature, the tip of the protrusion melts and adheres securely to the hot end, creating an airtight seal that guarantees the containment of the preform material. This innovative extrusion design ensures a safe and effective preform heating process without the risk of leakage.

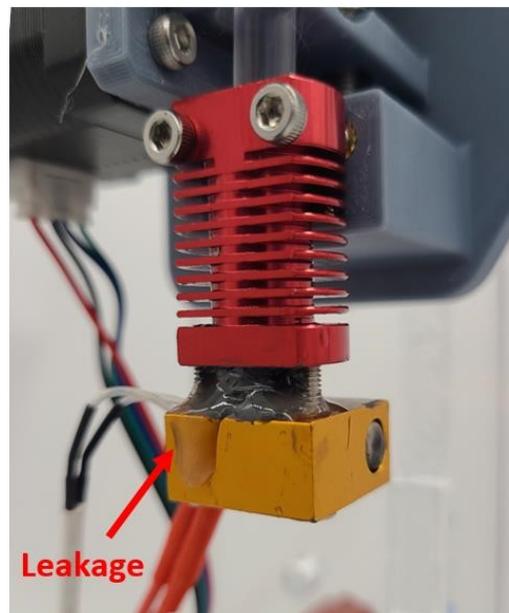


Figure 102. This shows the leakages of the preform when the heat sink is in use.

6.1.2.4 Extrusion system support:

Furthermore, a comprehensive revision was undertaken to enhance the support structure of the extrusion system. This revamping aimed to optimize material usage in the support components while streamlining assembly procedures. The unnecessary gap and protrusions between the mounting plate and the stepper motor's mounting location to provide reinforcement was eliminated since it was excess, and the stepper motor was to attach to the support without them.

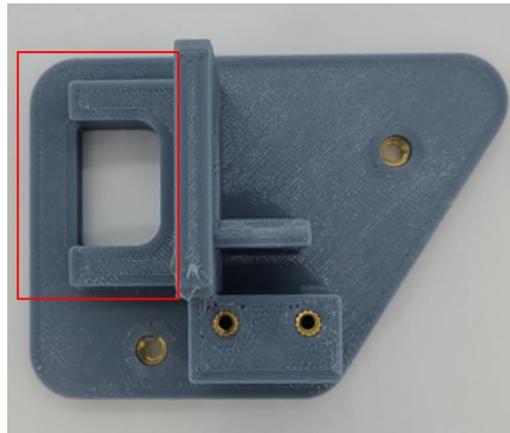


Figure 103. This is the excess mounting plate that support the stepper motor that was eliminated.

A reevaluation of the support's design also led to the removal of the heat block mount and the second guiding hole from the top. This presence of the hole proved redundant, as the stability of the preform was already well-maintained through other means—such as the lever's hole with protrusion, extrusion gears and pulley, and the heat block hole. Adjustments in the thickness of the support structure were also executed, focusing on optimizing structural integrity in relation to the substantial stepper motor. This optimization involved thinning areas away from the heated inserts and incorporating hollowed sections into the central part.

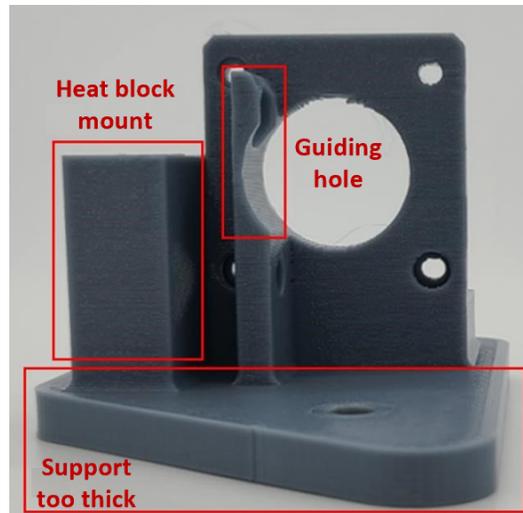


Figure 104. This shows the other excess blocks and features that are reduced and eliminated.

Notably, the revised support design yielded a size reduction by 43%, effectively diminishing both print time and resource consumption by 51%.

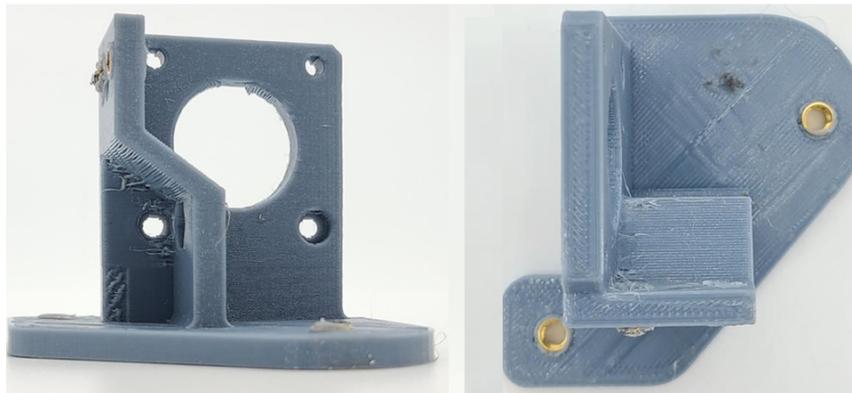


Figure 105. This the redesign of the extrusion system support.

Moreover, it was challenging during assembly, particularly with the constrained spaces in the old version of the extrusion support. It initial process of screwing the nut was holding the nut with a plier in place in the constrained space and using the Allen key to fasten the screw from the top. This procedure stripes teeth on the pliers, in addition to the assembly struggles, and ultimately wasting valuable time. With the removal of the heat block mount, attachment of the nut was using hex socket wrench to hold the nut in place win the ample space available and using Allen key to fasten the screw from the top.

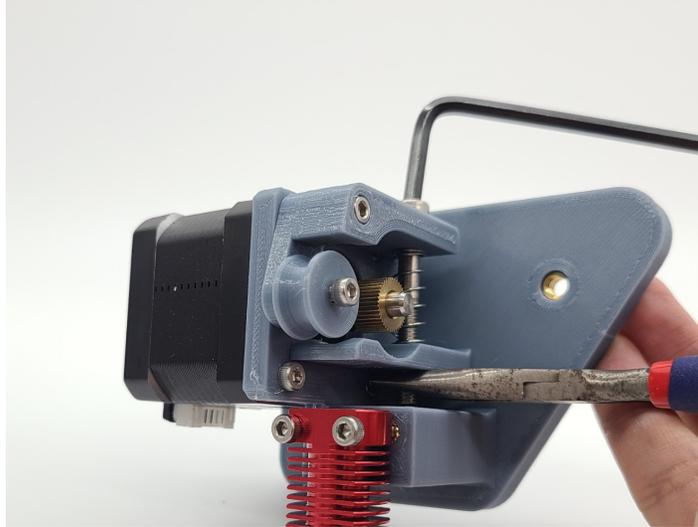


Figure 106. This shows the difficult in attaching the nut when fastening the screw.

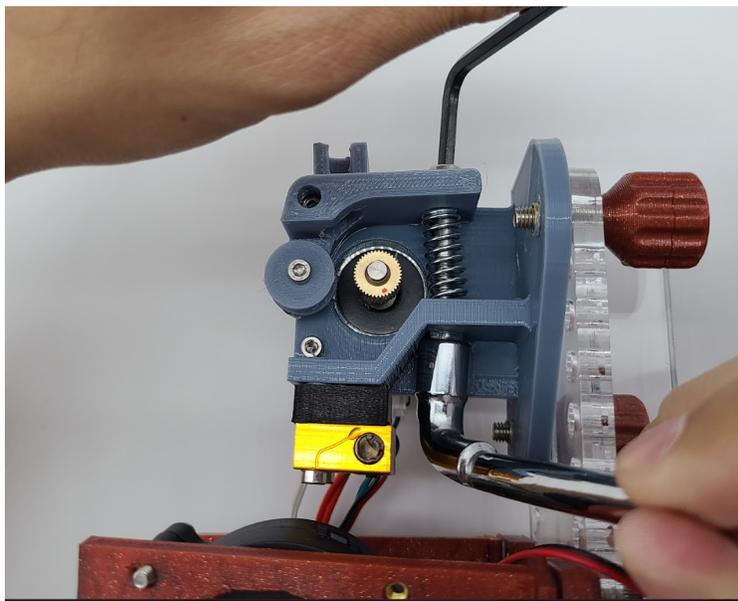


Figure 107. This shows the extrusion system with the new support and the easiness in attaching the nut when fastening the screw.

6.2 The Frame:

6.2.1 Previous Design:

The frame for the FrED system performs its functionality by holding all the sub system in place and allow seamless movement of the fiber through each section to the spool. The frame was

manufactured by using laser cutting techniques on acrylic sheets. It replicated two rows of vertical holes spaced 50mm apart horizontally and 20mm apart vertically between holes. In addition to that, different parts were fastened with thumb screws, with its casing made from 3D printing process. These thumb screws enable the convenient attachment and detachment of different parts along the vertical axis of the FrED. To establish the framework's back and support its structure, a rib was integrated, forming the distinctive T-tower shape. SciGrip Clear, a solvent cement known for acrylic bonding, was utilized to effectively join the rib to the back. While this cement initially sets within 10 to 15 minutes, it takes approximately 24 to 48 hours to fully cure and attain its maximum strength.

A trio of brackets made from 3D printing were used to fasten the T tower to base of the frame. This innovative bracket design offers the advantage of attaching the tower while the frame remains upright, thereby minimizing the need for reorienting the FrED during assembly. The frame's base is designed to accommodate the mounting of the T tower and gearbox on its upper side. On the underside, a drawer was incorporated to secure the printed circuit board (PCB) and electronics. Stabilizing the entire FrED setup are three feet attached with threaded inserts,

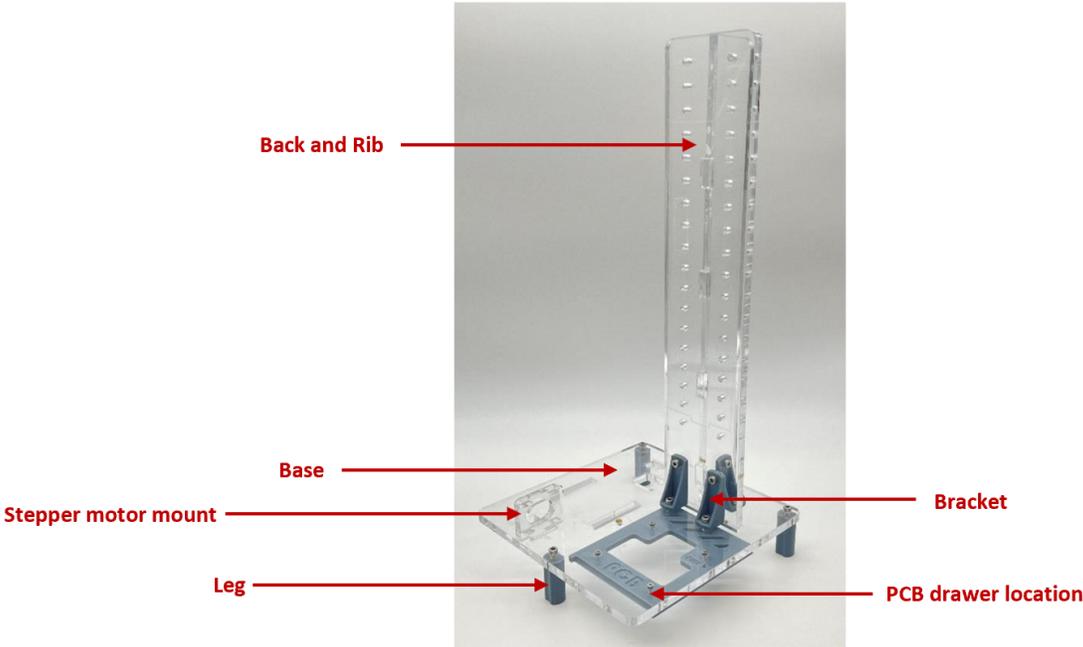


Figure 108. This is the full assembly of the fame of the FrED system.

6.2.2 Design changes:

6.2.2.1 T-slot system:

As stated above, the use of acrylic cement to bond the rib and backing of the frame takes an ample amount of setting time of 24 to 48 hours to attain full strength. To address this, a T-slot mechanical joint method was adopted to securely connect back and rib, and then to the base of the frame. The T-slot joint leverages the compressive force generated by a screw against a nut, delivering ample holding force. This is achieved by inserting the screw through a circular cutout on one panel and the nut through a corresponding T-shaped slot on the other panel [15]. The nut's width is designed to match the slot's dimensions, thereby preventing its rotation during the screw tightening process. It effectively eliminates the need for acrylic cement and bracket assemblies, leading to an expedited frame assembly process and a reduction of five parts from the entire FrED assembly.

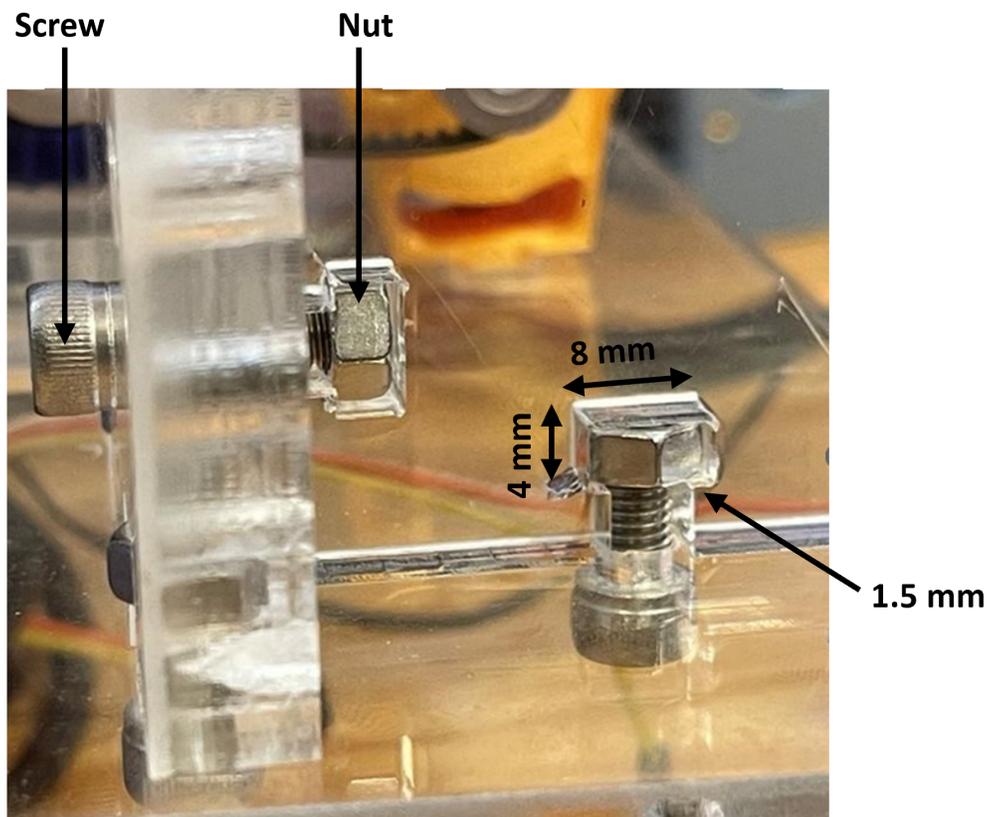


Figure 109. This shows the T-slot system to lock the back and rib, and then to the base of the frame.

However, the fastening process introduces the possibility of breaking the T slot space. To counter this potential issue, stress tests were conducted on the T slots to assess their resilience which the detail of the test is addressed in Wenhao Xu' thesis paper. This issues was tackled by non-recessed T slots with corner fillets could since it was able to withstand a 540-degree rotation before cracks emerged.

6.2.2.2 Frame size:

In alignment of reducing costs of the FrED system and making it lightweight, the frame of the previous FrED iteration was identified as a component with potential for cost reduction. The old version of the FrED frame incurs a cost of \$20.15 and offers scope for further optimization by reducing the width and thickness of the acrylic sheets.

There is potential to reduce the width in the top segment, although the bottom portion requires a certain width to facilitate proper mounting of the T tower onto the base. Concerning the back of the frame, the established geometry of two rows of mounting holes spaced 50mm apart horizontally and 20mm apart vertically has been deemed sufficient for modular component mounting. However, the width of the frame's back was reduced, while keeping it functional requirements.

Furthermore, the base of the frame underwent a reduction in size, resulting in a more compact configuration. This alteration necessitated a repositioning of the motor. Comprehensive information regarding the reduction and repositioning of frame components can be found in Wenhao Xu's thesis paper. These modifications collectively contributed to approximately a 20% decrease in the frame's overall dimensions. Consequently, the weight of the FrED was reduced, resulting in enhanced portability and easier handling. The new version of FrED frame incorporates the use of rubbers under the four feet of the frame. The resilient property of rubber effectively ensures that all feet make solid contact with the ground. Additionally, the rubber attachments contribute to vibration isolation to a certain extent, thereby aiding in noise reduction during operation. Numerous rubber feet options were subjected to experimentation, and among them, the one depicted in Figure X was selected. This particular choice strikes the optimal balance between factors such as vibration isolation, elasticity, aesthetic appeal, and cost-effectiveness.

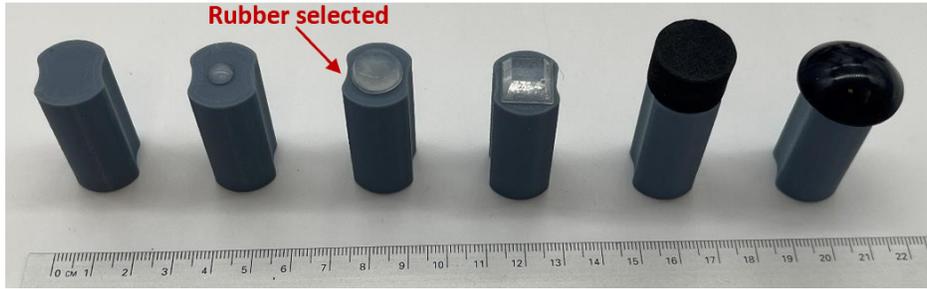


Figure 110. This shows the different rubbers that were used to aid in elasticity and vibration isolation in the FrED system.

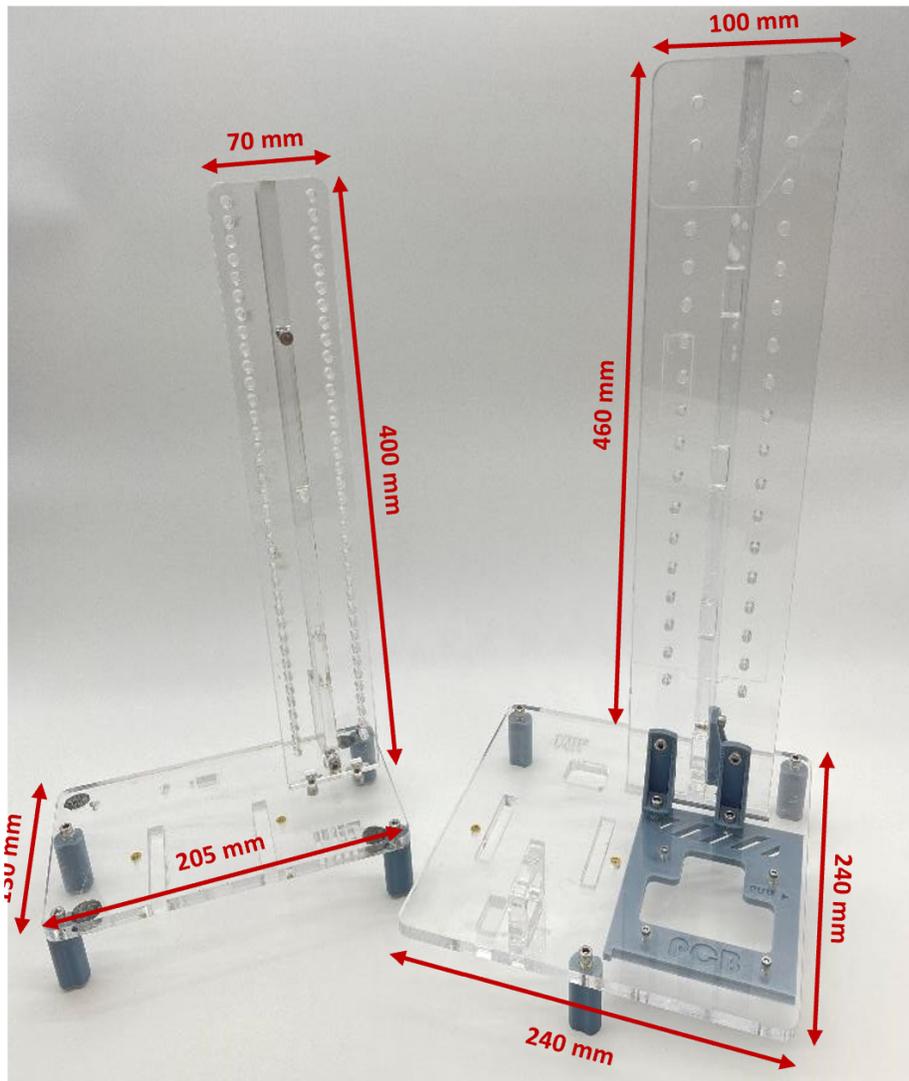


Figure 111. This shows the size of the old version of the FrED frame (right frame) and the new version with reduced size (left frame).

7. Upgrading the Micro processing System:

7.1 Teensy:

A microcontroller constitutes a compact computer system that encompasses a central processing unit (CPU), memory, and programmable input/output capabilities all within a solitary integrated circuit. Various iterations of microcontrollers are readily accessible in the market. In the context of the older version of FrED, the Teensy 4.1 microcontroller was employed. This selection was based on the substantiated capability of the Teensy 4.1 to effectively govern FrED utilizing advanced methodologies such as deep reinforced learning controls.

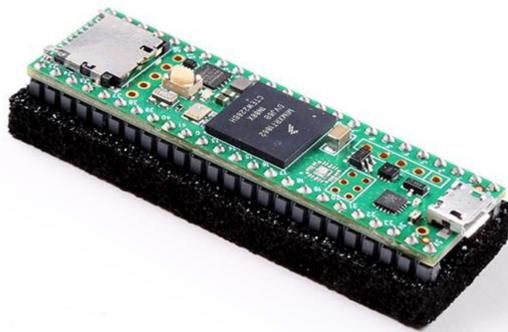


Figure 112. This is the Teensy Arduino that is used in the FrED system for deep reinforce learning control.

However, a notable limitation of the Teensy Arduino pertains to its inability to perform image processing tasks. This constraint arises due to the inherent hardware limitations of the Teensy microcontroller. Consequently, the execution of image processing functions must be relegated to an external device, such as a laptop. This external device would handle the image processing workload and subsequently transmit the processed data back to the Teensy for analysis and to contribute to the PID control of the fiber diameter in FrED [7]. While this approach was an option, Professor Brian Anthony held reservations about its implementation. His preference was for an integrated system where all processing tasks occur within a single framework, rather than being distributed across multiple devices.

It is also important to acknowledge that relying on an external device like a laptop introduces potential vulnerabilities. The operation of FrED could be jeopardized in the event of the external

device, such as the laptop, encountering technical issues or failures during its operation. This underscores the need for a robust and reliable setup to ensure the continuous and uninterrupted performance of FrED.

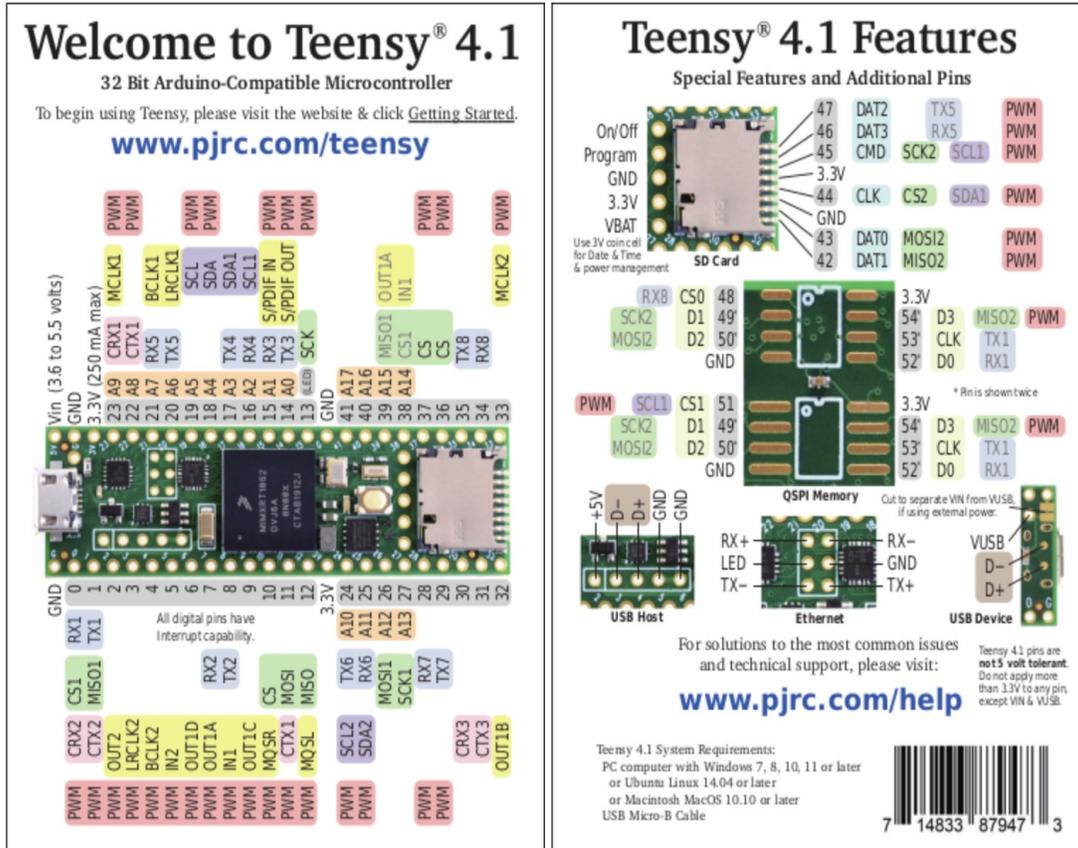


Figure 113. This is the schematic of the Teensy Arduino I/O.

7.2 Raspberry Pi:

In our pursuit of achieving comprehensive system integration and streamlined processing, we opted for a Raspberry Pi microprocessor model, specifically the 4 Model B variant with 8GB of RAM and a 128GB SD card. The primary focus of this choice was not centered on storage capacity but rather on assessing the Raspberry Pi's compatibility and efficacy within the integration framework.

The Raspberry Pi, fundamentally, is a credit-card-sized computer that encompasses a diverse array of components and functionalities on a single board. This board typically comprises a central

processing unit (CPU), memory modules, USB ports, HDMI ports, Ethernet connectivity, GPIO (General Purpose Input/Output) pins, and other peripherals. Such a configuration renders the Raspberry Pi capable of performing a range of tasks akin to those of a conventional computer, albeit on a more compact scale.

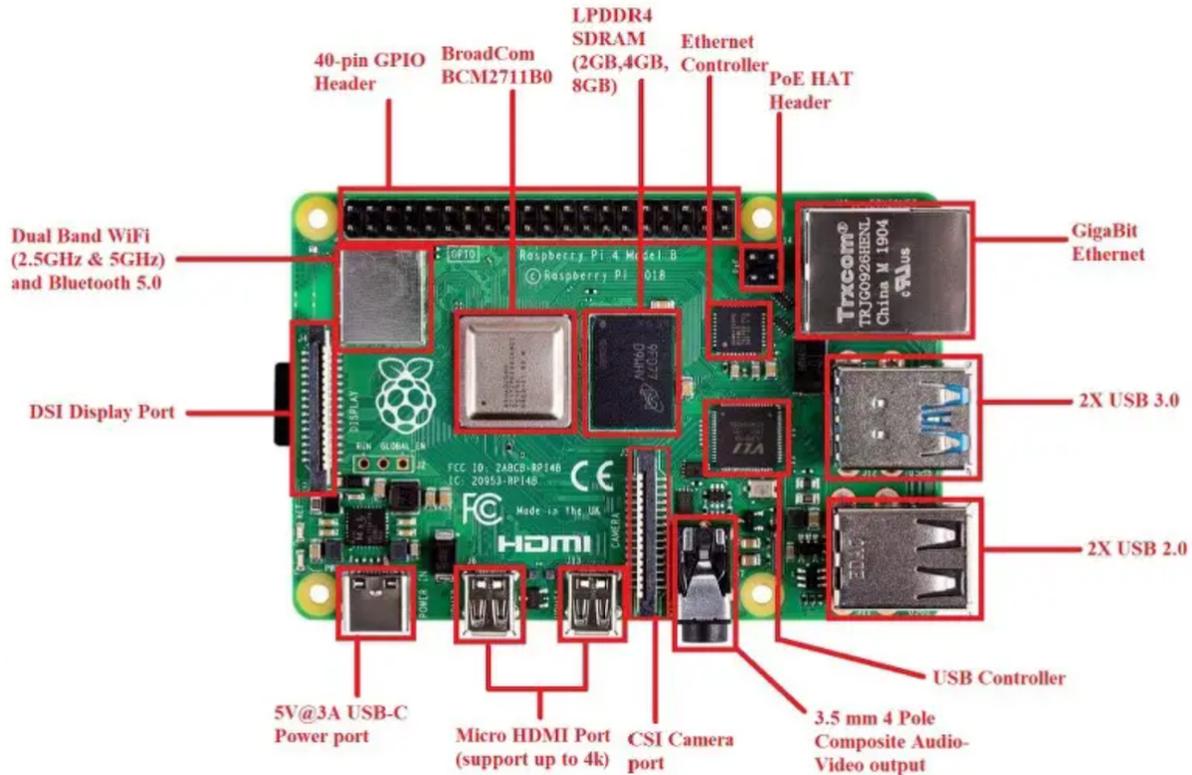


Figure 114. This is the 4 Model B, 8GB of RAM Raspberry Pi with a 128GB SD card for the integration of the running the FrED system and image processing.

There are several advantages to employing a Raspberry Pi microprocessor within the system. Firstly, its compact size and relatively low cost make it a convenient and cost-effective choice for integration into various projects. Additionally, the Raspberry Pi's processing capabilities are robust enough to handle complex tasks, including running the FrED system and performing image processing tasks. This capability is crucial for tasks like analyzing images to aid in controlling the fiber diameter accurately [16].

Furthermore, the Raspberry Pi board incorporates multiple ports and connectors, offering the potential for seamless integration with other components utilized in the FrED system, both in its current configuration and in anticipation of future enhancements [17]. These expansion options empower the system to accommodate additional functionalities or components as the project evolves and requirements evolve.

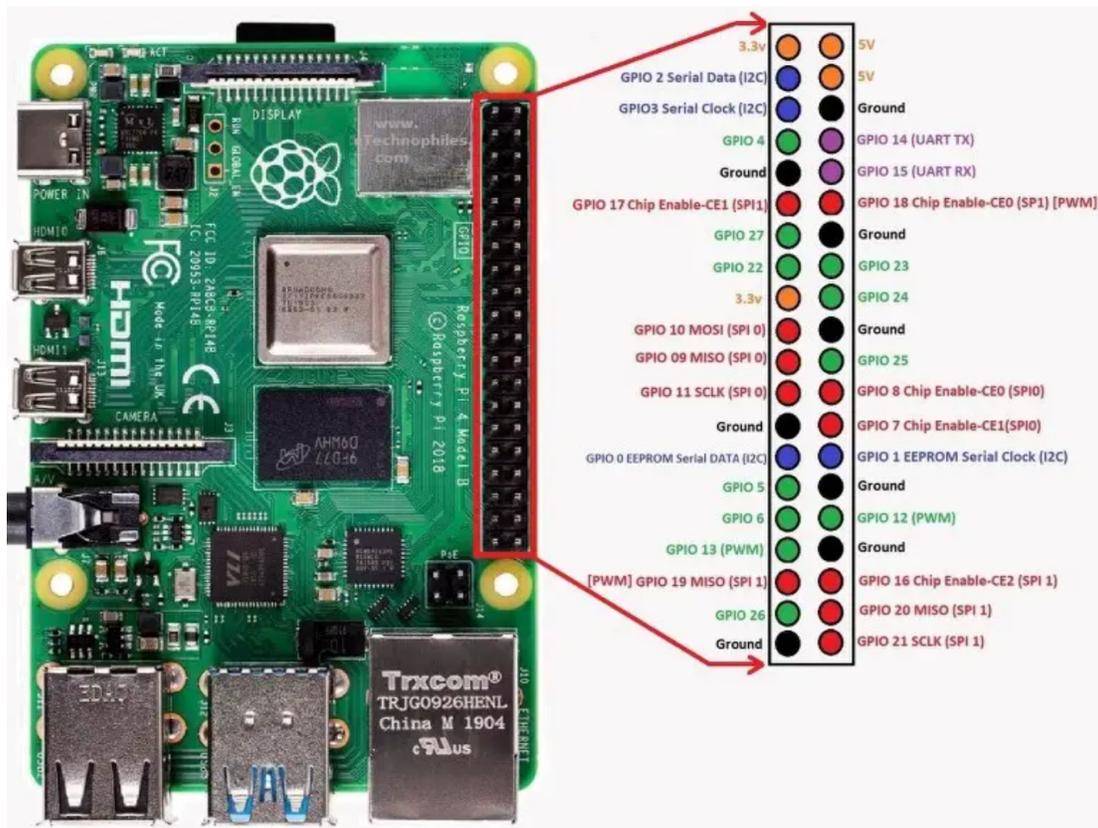


Figure 115. This is the schematic of the GPIO pins of the Raspberry Pi.

7.3 Raspberry Pi Installation:

Before fully committing to the utilization of the Raspberry Pi microprocessor, it was imperative to ascertain its capability to smoothly and seamlessly operate the FrED system without encountering any hindrances. This validation process commenced with a series of steps aimed at ensuring the optimal functioning of the Raspberry Pi within the context of the FrED project.

The initial phase entailed the installation of the requisite operating system onto the Raspberry Pi. To facilitate this installation, a pivotal preliminary measure was to format the micro-SD card

provided within the kit. This formatting procedure aimed to eliminate any extraneous files or potential impediments that might compromise the effective execution of the operating system. Achieving this was accomplished through the using of an SD formatter, which could be found online. By connecting the micro-SD card to a laptop, a swift format was executed, ensuring a clean slate for the forthcoming operating system installation.

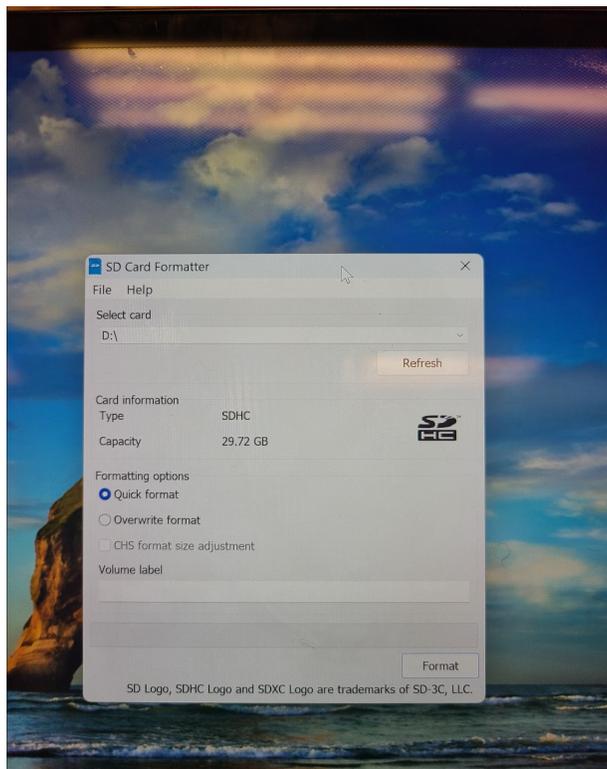


Figure 116. This shows the use of the SD formatter to clean out the SD card.

Subsequently, the process moved forward with the installation of the Raspberry Pi Imager, an essential tool obtained from the official Raspberry Pi website. This tool was instrumental in the successful integration of the operating system required for the Raspberry Pi's functioning. Through the Imager, the specific Raspberry Pi model is selected, along with the designated micro-SD card, thus enabling the seamless transfer of the necessary operating system onto the formatted card.



Figure 117. This shows using the Raspberry Pi Imager to download the requisite operating system on the SD card.

With the operating system downloaded on the SD card, the next step is inserting the SD card into the Raspberry Pi. Powering on the Raspberry Pi was facilitated by connecting the USB-C cable included in the kit. To ensure a comprehensive setup process, the Raspberry Pi was further connected to a monitor, keyboard, and mouse. An important aspect to bear in mind was that the connection between the Raspberry Pi and the monitor necessitated the use of a mini-HDMI cable shown in Figure 114.

As the Raspberry Pi booted up, a guiding resource in the form of a YouTube video titled "The New Method to Setup Raspberry Pi (2023 Update)" was consulted [18]. This video served as a comprehensive tutorial to ensure the accurate configuration of the Raspberry Pi system. For the direct link to the video, it can be accessed in the References section below.

7.4 Raspberry Pi and PCB integration

Subsequent to the setup and initialization process, the Raspberry Pi was integrated into the FrED system's PCB. This integration was executed, taking into consideration the specific components on the PCB that possessed pulse width modulation (PWM) attributes, those requiring analog write capabilities, and those reliant on GPIO ports to achieve full functionality. Ensuring the alignment of the jumper wire schematic connecting the Raspberry Pi and the PCB was of paramount importance, mirroring the configuration utilized with the Teensy microcontroller. This meticulous synchronization guaranteed the seamless operation of all integrated components.

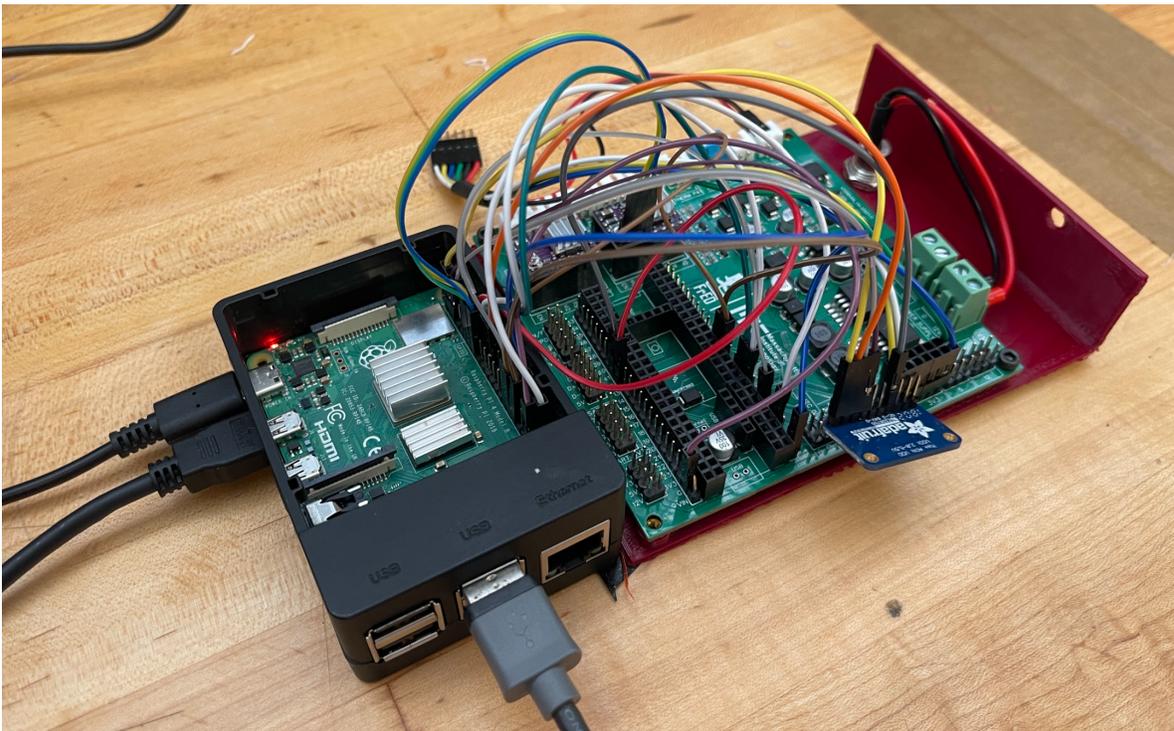


Figure 118. This shows the use of the jumper wire used for the Raspberry Pi and PCB integration.

The schematic detailing the precise ports employed for connecting the Raspberry Pi to the PCB is succinctly outlined in the table provided below for reference.

Table 2. This shows the GPIO ports from Figure 114 that are used to connect to the PCB ports for output for the seamless integration of the FrED component.

Output	GPIO Number	Board Number
	3.3V	1
	2 (I2C SDA)	3
	3 (I2C SCL)	5
	4	7
GND	GND	9
M0 Stepper Controller	17	11
M1 Stepper Controller	27	13
M2 Stepper Controller	22	15
	3.3V	17
MCP3008 Din	10 (SPI MOSI)	19
MCP3008 DOUT	9 (SPI MISO)	21
MCP3008 CLK	11 (SPI SCLK)	23
GND	GND	25
	0 (I2C SDA)	27
DC Motor PWM	5	29
Heater PWM	6	31
Fan PWM	13	33
	19 (PCM FS)	35
	26	37
GND	GND	39

**B
L
A
N
K**

Board Number	GPIO Number	Output
2	5V	Power Input
4	5V	
6	GND	GND
8	14 (UART TXD)	
10	15 (UART RXD)	
12	18 (PCM CLK)	
14	GND	GND
16	23	DC Motor Encoder 1
18	24	DC Motor Encoder 2
20	GND	GND
22	25	
24	8 (SPI CE0)	MCP3008 CS
26	7 (SPI CE1)	
28	1 (I2C SCL)	
30	GND	GND
32	12	Extruder Motor Step
34	GND	GND
36	16	Extruder Motor Dir
38	20 (PCM DIN)	
40	21 (PCM DOUT)	

A notable limitation of the Raspberry Pi is its lack of an analog read capability, which holds significance for the thermistor that measure temperature in an analog manner. To circumvent this limitation, the integration of an ADS1115 Adafruit chip was deemed necessary. This chip serves as an analog-to-digital converter, enabling the translation of analog signals into digital data that can be processed by the Raspberry Pi. The Adafruit ADS1115 chip functions as a bridge between the analog input from the thermistor and the Raspberry Pi's digital processing capabilities. It operates by precisely converting analog voltage values into their corresponding digital representations. This conversion process involves the successive comparison of the input voltage against a reference voltage. The resulting digital output is then transmitted to the Raspberry Pi, allowing it to interpret and process the acquired data effectively [19]. In essence, the integration of the Adafruit ADS1115 chip into the system architecture addresses the absence of analog read capabilities in the Raspberry Pi, facilitating the acquisition and utilization of analog data from various components in the FrED system.

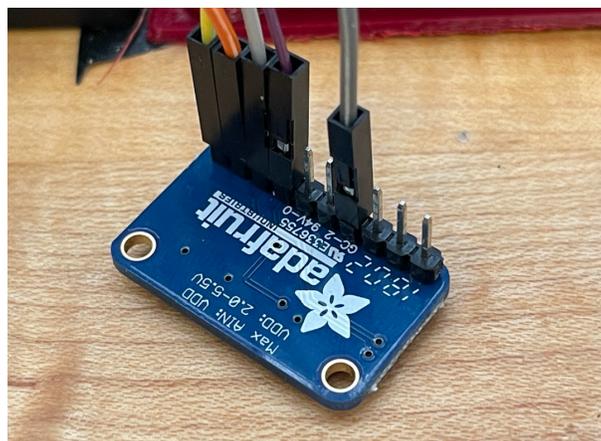


Figure 119. This is the Adafruit ADS1115 chip that is used as the analog to digital converter for the thermistor.

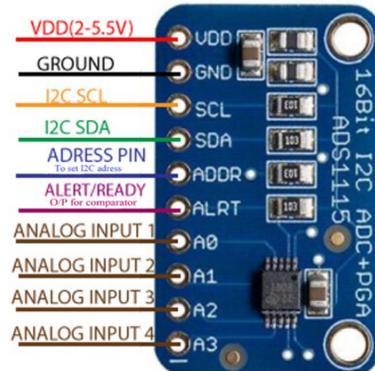


Figure 120. This is the schematic of the Adafruit ADS1115 chip.

7.5 Translation of code to Raspberry Pi:

The Teensy microcontroller was programmed using the Arduino IDE in the C++ programming language. However, a transition was required due to the distinct nature of the Raspberry Pi's IDE, which predominantly relies on Python. This shift necessitated the conversion of the existing C++ code for each individual component to a compatible Python format, while also incorporating the requisite libraries for proper functionality. The adaptation process commenced by addressing each component separately to ensure their accurate operation within the new programming framework.

Key components during this process included the heater, thermistor, DC motor controlling the spool, and the stepper motor governing the extrusion mechanism. These components were addressed to ensure seamless interaction within the Python-based programming environment. A subsequent and crucial addition to the integrated code was the incorporation of the microscope's control for image processing. The details of the code alterations for the following listed above can be found in Wenhao Xu's thesis paper. The integrated code is displayed in the Appendix.

7.6 PCB remodeling:

Throughout the iterative process of prototyping various component functions, as well as testing the functionality of the Raspberry Pi, the existing PCB was used. During these phases, as the necessity for connections arose, some ports and junctions on the PCB were shared among multiple components. As the selection of components for the new iteration of the FrED system reached its conclusion, it became evident that a redesign of the PCB was indispensable.

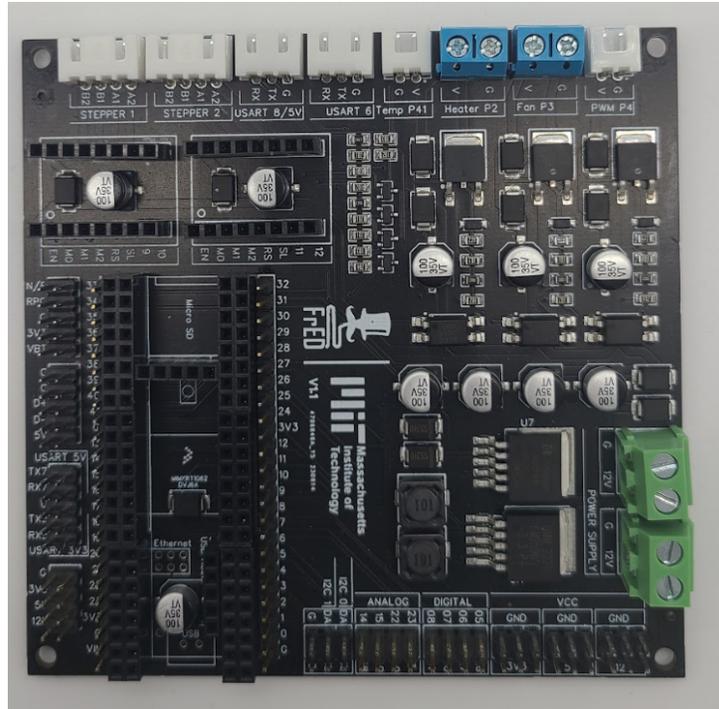


Figure 121. This is the old version of the PCB used for testing of the new component and Raspberry Pi.

The primary objective of this redesign endeavor was multifaceted. First and foremost, the intention was to eliminate any components or ports that no longer served a purpose in the refined FrED system like one of the stepper drivers. Simultaneously, the redesign aimed to introduce provisions for future FrED components that might be integrated like a PWM port, ensuring the PCB's adaptability and scalability. In conjunction with this, a concerted effort was dedicated to reducing the overall size of the PCB, not only for practical space considerations but also as a measure to curtail costs. Notably, the PCB emerged as a notable expense, and its downsizing held the potential for significant cost savings.

Furthermore, the redesigned PCB was made to facilitate the seamless integration of the Raspberry Pi microprocessor. This involved designing specific connectors and pathways that remove the need for jumper wires, thereby streamlining the integration process and enhancing the system's robustness. The new PCB was an embedded chip that fulfilled the role of an analog-to-digital converter. This innovation obviated the need for external converters and enhanced the PCB's self-sufficiency in handling analog signals from various components. This embedded chip played a

pivotal role in translating analog data, such as temperature readings from the thermistor, into digital signals that could be processed by the Raspberry Pi.

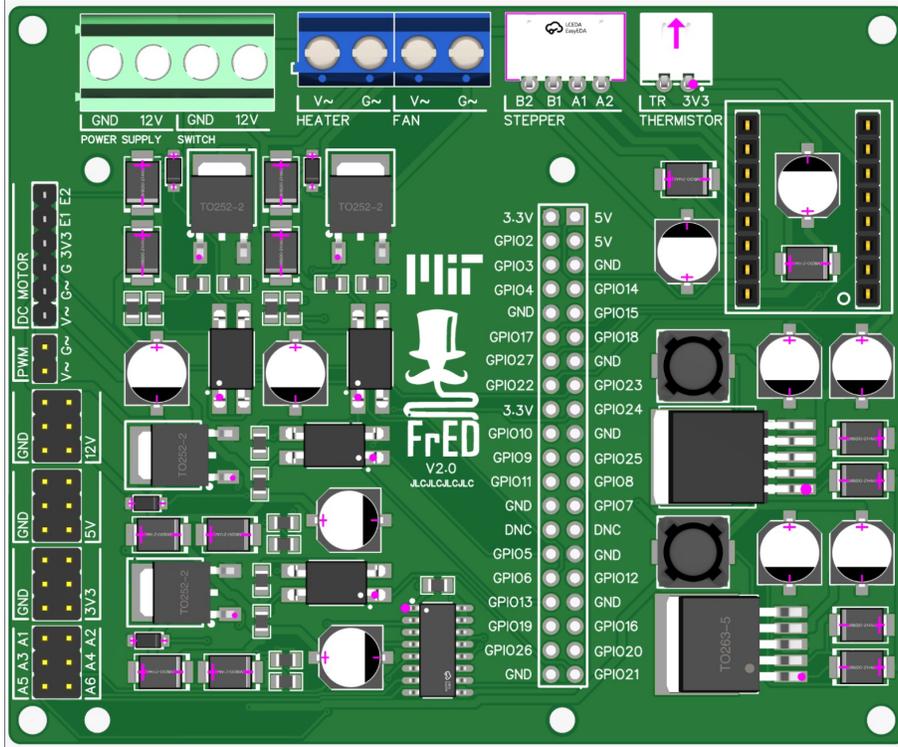


Figure 122. This is the new PCB that will be used for the new low-cost FrED system.

By repositioning some of the existing components and optimizing their arrangement, the dimensions of the PCB were successfully reduced by approximately 35%. This reduction in size not only contributed to a more compact and streamlined system but also resulted in a significant cost reduction of approximately \$8. Further details regarding the changes made to the PCB can be found in Wenhao Xu's thesis paper.

8. Cost Analysis:

8.1 Old Design:

One crucial factor in the transition of FrED from a research-based project to an educational tool for teaching manufacturing and machine learning principles was the constant consideration for cost-effective materials. This was essential to ensure accessibility for both users and learners. During its research-focused phase, FrED incurred a total cost of approximately \$5,428 [6].

In the initial phase of transforming FrED into an affordable teaching tool, the 2022 MEng batch made significant strides by reducing the cost dramatically to \$270 [7]. However, Prof. Brian Anthony aspired for an even more ambitious target – aiming to bring the cost down to \$100. This endeavor posed a significant challenge, given that the materials used for the low-cost FrED were already among the most economical available in the market. Despite this, the team set a goal to redesign the FrED system at a cost of \$150.

The previous iteration of the FrED system consisted of fiber collection, fiber extrusion, diameter measurement, frame, cooling, electronics, and specialized sub-parts. The cost breakdown for each sub-part is presented in the table. Notably, the specialized sub-parts are integral components utilized across all sections of the FrED system.

Table 3. This shows the number of parts used in each sub part and the cost for each of them.

Sub part	Quantity	Cost
Fiber collection	21	36.58
Fiber extrusion	11	11.94
Diameter measurement	6	23.64
Frame	9	20.15
Cooling	2	3.62
Special parts	88	35.49
Electronics	15	138.7
	152	270.12

The manufacturing process primarily involved techniques such as 3D printing for parts fabrication, laser cutting of acrylic, adhesive application, and limited drilling and band sawing for specific components. The remaining materials were primarily sourced externally, mainly from Amazon or McMaster.

8.2 New FrED design:

The cost analysis of the new components incorporated into the redesigned FrED placed a strong emphasis on sourcing the most cost-effective options without compromising functionality. A similar assessment was carried out for the pre-existing components, exploring the potential for cost reduction without sacrificing quality. The following tables provide a breakdown of material costs for each subcomponent. Notably, the tables exclude the special parts, which are detailed within their respective subcomponents for clarity.

As part of the optimization process, labor-intensive manufacturing methods such as band sawing and milling were eliminated. Instead, the focus shifted towards utilizing 3D printing, laser cutting, and outsourcing for production.

8.2.1 Cost analysis of the fiber extrusion subcomponents:

A significant transformation occurred in the extrusion segment. The heat sink and pipe were replaced with a PC alloy 3D-printed block, resulting in a more streamlined and cost-effective solution. Furthermore, enhancements were made to the overall structure and thickness of the 3D-printed parts, leading to a 37% reduction in material usage. This reduction not only made the 3D-printed components more economical but also contributed to the overall cost savings of the project.

Table 4. This is the bill of material for the fiber extrusion subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
Stepper Motor	1	5.998	5.998
M3-10 screw	3	0.06	0.18
M3-25 screw	1	0.0849	0.0849

M5-35 screw	1	0.35	0.35
Spring	1	0.745	0.745
M3-12 screw	1	0.06	0.06
Extrusion Support	1	0.47	0.47
Lever	1	0.15	0.15
PC alloy block	1	0.05	0.05
Heat Block	1	1.798	1.798
Thermistor and Heater	1	2.158	2.158
Thumb screws	2	0.06	0.12
Driver gear	1	0.4995	0.4995
Idler pulley	1	0.025	0.025
M5 lock nut	1	0.052	0.052
M3 inserts	2	0.0899	0.1798
M5 inserts	2	0.1798	0.3596
M3-22 screw	2	0.0929	0.1858
Wire protector	1	0.7	0.7
	25		14.16

8.2.2 Cost analysis of the cooling subcomponents:

The primary alteration within the cooling sub part involved a shift to utilizing centrifugal fans instead of the pricier axial counterparts. While this change initially incurred a slightly higher cost due to the implementation of two fans and the creation of more intricate supports for enhanced maneuverability, the investment proved worthwhile. This adaptation ultimately resulted in an optimal cooling mechanism for the fiber, justifying the additional expense.

Table 5. This is the bill of material for the cooling subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
Fan support	2	0.37	0.74

M3-18 screw	4	0.09	0.36
M3 inserts	6	0.0899	0.5394
M5 inserts	4	0.1798	0.7192
Centrifugal fan	2	4.495	8.99
M3-8 screw	2	0.054	0.109
Thumb screws	4	0.06	0.24
	24		11.69

8.2.3 Cost analysis of the diameter measurement subcomponents:

The key modification within the diameter measurement sub part centered around identifying a more cost-effective USB micrometer compared to the previously employed one. Additionally, a significant alteration involved transitioning to a fully modular 3D-printed system. This change contributed to cost reduction, as meticulous consideration was given to optimizing the thickness to achieve the ideal balance between component strength and weight-bearing capacity within the sub part.

Table 6. This is the bill of material for the diameter measurement subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
Adjustable Side Camera Mount base	1	0.32	0.32
Adjustable Side Camera Pulley Mount base	1	0.22	0.22
Adjustable Side Camera Mount Camera Slider	2	0.15	0.3
Adjustable Side Camera Mount Single Side pulley	1	0.05	0.05
Adjustable Side Camera Mount Camera Holder		0.27	0
Card	1	0.01	0.01
Adjustable Side Camera Mount Card Holder	1	0.27	0.27
Adjustable Side Camera Mount Pulley Slider	2	0.1	0.2

Pulley	3	0.657	1.971
M3 inserts	9	0.0899	0.8091
M5 inserts	2	0.1798	0.3596
M3-5 screws	9	0.0978	0.8802
Thumb screws	2	0.06	0.12
USB Camera	1	16.99	16.99
	35		22.49

8.2.4 Cost analysis of the fiber collection subcomponents:

The primary alterations within the fiber collection sub part involved a switch from a $\frac{3}{4}$ acrylic sheet to a $\frac{1}{4}$ acrylic sheet. This change aimed to achieve two goals: reducing the cost of the sheet used and making the FrED system lighter and more portable. Moreover, cost-effective options were identified for other components such as the belt and tube stock.

Table 7. This is the bill of material for the fiber collection subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
Modular Plate	1	1.51	1.51
Main Axle Plate	2	0.5	1
DC Motor Mount	1	0.22	0.22
Scotch Yolk Wheel	1	0.19	0.19
Rotary shaft_L80	1	0.399	0.3996
Slider	1	0.37	0.37
Round Tube Stock_L185	1	0.62	0.62
Spool	1	0.73	0.73
Spool Cap	1	0.06	0.06
Axle Gear_D25	1	0.07	0.07
Spur Gear Stock_D20x62	1	0.44	0.44
Timing Belt	1	1.088	1.088
Rotary Shaft_D5_L20	1	0.099	0.099

Timing Belt Pulley	2	0.798	1.596
Plastic Miter Gear	1	3.66	3.66
Modified Miter Gear	1	0.09	0.09
Plastic Spur Gear_D28	1	0.08	0.08
Round Tube Stock_L14.5	1	0.05	0.05
Cover	1	0.94	0.94
DC motor	1	16.88	16.88
M5-16 screws	2	0.13	0.26
M3 inserts	1	0.0899	0.0899
M3-8 screw	2	0.05	0.1
	27		30.54

8.2.5 Cost analysis of the frame subcomponents:

The notable alteration in this sub part paralleled that of the fiber collection, with the adoption of a $\frac{1}{4}$ acrylic sheet in lieu of the previous $\frac{3}{4}$ acrylic sheet. Furthermore, a substantial enhancement was achieved by decreasing the dimensions of the frame's back, rib, and base by approximately 20%. This adjustment was undertaken to achieve two key objectives: reducing overall weight and enhancing affordability in the manufacturing process.

Table 8. This is the bill of material for the frame subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
Back	1	2.11	2.11
Rib	1	0.66	0.66
Base	1	2.31	2.31
M5-14 screws	5	0.15	0.75
M5 nuts	5	0.097	0.488
Leg support	2	0.55	1.1
Rubber feet	4	0.045	0.183
M5-12 screw	4	0.094	0.379

M5 inserts	2	0.179	0.359
	25		8.34

8.2.6 Cost analysis of the electronics subcomponents:

The significant transformations within the electronics sub part encompassed a shift towards integrating the FrED system solely onto the Raspberry Pi, as opposed to the previous version that included both the Raspberry Pi and Teensy. This substantial adjustment led to a considerable cost reduction, eliminating the need for the Teensy while also sourcing an economical Raspberry Pi.

Additionally, a noteworthy alteration involved reducing the size of the printed circuit board (PCB) by approximately 35%, coupled with a reduction in the number of components. This optimization resulted in a more cost-effective solution compared to the older version.

Table 9. This is the bill of material for the electronics subsystem for the new version of the low cost FrED system.

Part Name	Quantity	Unit Cost	Total Cost
PCB	1	11.49	11.49
Raspberry pi 4 1 GB	1	35	35
Power Supply	1	17.99	17.99
Switch	1	0.599	0.599
FF Standoff (M2.5x12)	4	0.03	0.12
MF Standoff (M3x10)	4	0.045	0.18
M3-6 philips screw	4	0.06	0.24
M2.5-6 philips screw	8	0.079	0.639
Drawer	1	1.3	1.3
M3 inserts	4	0.089	0.359
Plug connector	1	1.373	1.373
Stepper Motor Driver	1	1.999	1.999

	31	71.30
--	-----------	--------------

8.2.7 Cost analysis of the overall FrED system changes:

Examining the table provided below, the overall cost of the FrED system amounts to approximately \$159, reflecting a notable reduction of about 42%. Although the precise target of \$150 was not quite achieved, considering the challenges of sourcing cost-effective components and the meticulous process of removing or substituting parts to ensure both affordability and optimal performance, significant progress has been made.

Furthermore, a significant achievement is witnessed in the reduction of the FrED system's weight. From its initial mass of 2.27 kg, the system's weight has been successfully brought down to 1.7 kg – a 25% reduction compared to the previous version. This lighter weight not only enhances portability and ease of transport but also contributes positively to overall packing considerations.

Table 10. This shows the number of parts used in each sub part and the cost for each of them.

Sub part	Quantity	Cost
Fiber collection	27	30.54
Fiber extrusion	25	14.16
Diameter measurement	35	22.49
Frame	25	8.34
Cooling	24	11.69
Electronics	31	71.30
	167	158.52

The comprehensive bill of materials will be presented in the Appendix, complete with links directing to the sources where these specific parts can be obtained. This detailed reference will provide easy access to all the necessary information regarding the components used in the FrED system.

9. Future Works

Following the hardware modifications aimed at optimizing fiber cooling through a pulley system to enhance stability and support computer vision via the USB microscope, the final appearance of the FrED system is depicted below.

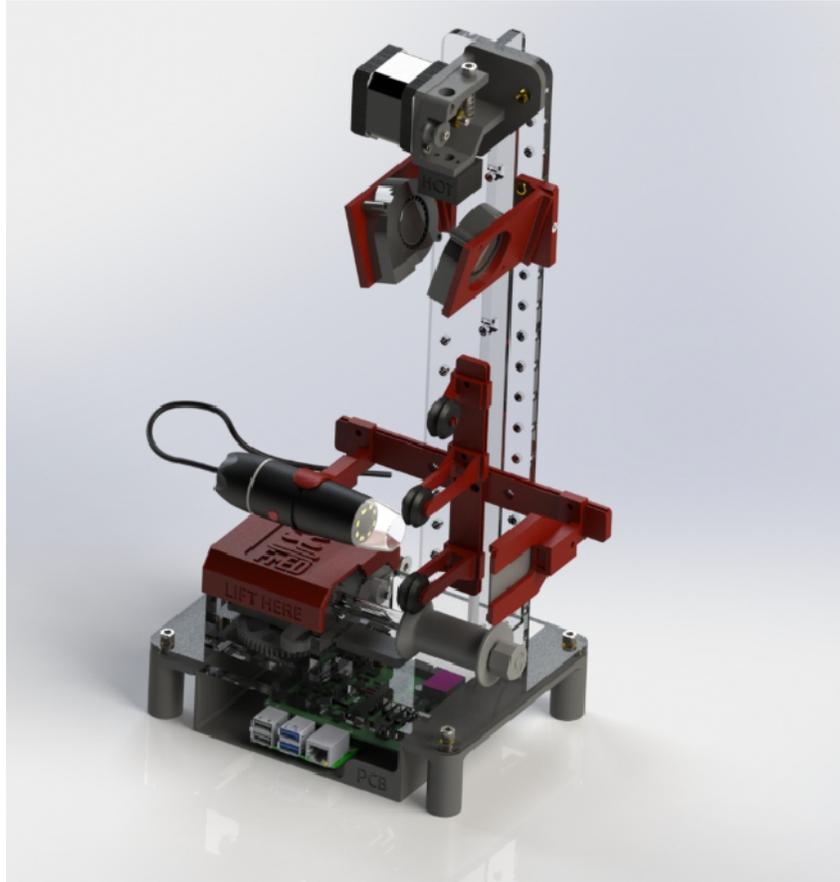


Figure 123. This is a rendered picture of the current low cost FrED system.



Figure 124. This is the actual image of the new low cost FrED system.

Nonetheless, there remains ongoing work to be undertaken. There exists an ongoing scope for hardware enhancement to further optimize functionality and strive towards achieving the targeted cost of \$100. An area of focus involves the incorporation of a tension sensor into the system. This sensor plays a pivotal role in providing feedback and facilitating PID control within the FrED system. Its integration holds educational significance, particularly in teaching principles of deep reinforcement learning (DRL) and machine learning.

To accomplish this, integration with the PID control algorithm developed by Somesh Jaiswal over the summer is paramount. Additionally, efforts will be directed towards manufacturing multiple FrED systems and establishing an assembly line for their production. This step serves dual purposes: educating individuals in manufacturing principles and systems and validating the repeatability and reliability of the FrED system in consistently producing accurate data and fibers.

This successful validation will subsequently pave the way for a pilot program, allowing interested individuals and past alumni to engage with the FrED system. This trial phase aims to ensure its proper functioning across diverse users, further enhancing its practical applicability.

10. Reference

- [1] D. D. Kim, "Design and Development of Desktop Fiber and Fabric Manufacturing System for Advanced Materials," May 2020.
- [2] J. D. Cuiffi, H. Wang, J. Heim, and S. Kim, "Factory 4.0 Toolkit for Smart Manufacturing Training," Jul. 2021.
- [3] D. D. Kim and B. Anthony, "Design and fabrication of desktop fiber manufacturing kit for Education," Volume 3: Vibration in Mechanical Systems; Modeling and Validation; Dynamic Systems and Control Education; Vibrations and Control of Systems; Modeling and Estimation for Vehicle Safety and Integrity; Modeling and Control of IC Engines and Aftertreatment Systems; Unmanned Aerial Vehicles (UAVs) and Their Applications; Dynamics and Control of Renewable Energy Systems; Energy Harvesting; Control of Smart Buildings and Microgrids; Energy Systems, 2017.
- [4] S. Kim, D. D. Kim, and B. W. Anthony, "Dynamic Control of a Fiber Manufacturing Process using Deep Reinforcement Learning," 2019.
- [5] S. S. Tallury, B. Pourdeyhimi, M. A. Pasquinelli, and R. J. Spontak, "Physical microfabrication of shape-memory polymer systems via bicomponent fiber spinning," *Macromolecular Rapid Communications*, 2016.
- [6] Aviva Jesse Levi, "Design and Manufacturing of the Extrusion Assembly for an Advanced Process Control Educational Device" September 2022
- [7] Russel Bradley, "Design and Manufacturing of Educational Fiber Extrusion Device and Smart Factory" Septmeber 2022
- [8] Rui Li, "Design and Manufacturing of the Filament Collection and Diameter Measurement Systems of Fiber Extrusion Device for Educational Purposes", September 2022

- [9] Tanach Rojrungsasithorn, "Factory and Material Flow Design for Mass Production of an Advanced Process Control Educational Device", September 2022
- [10] "Choosing and Using Advanced Peltier Modules for Thermoelectric Cooling," Digi-Key Electronics. [Online]. Available: <https://www.digikey.com/en/articles/choosing-using-advanced-peltier-modules-thermoelectric-cooling>.
- [11] "Centrifugal Fans Guide," RS Components. [Online]. Available: <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/centrifugal-fans-guide>.
- [12] "Download Confirmation," Keyence. [Online]. Available: https://www.keyence.com/download/download/confirmation/?dlAssetId=AS_16773&dlSeriesId=WS_SR47840&dlModelId=&dlLangId=&dlLangType=en-GB.
- [13] "Camera Module V3," Raspberry Pi. [Online]. Available: <https://www.raspberrypi.com/products/camera-module-3/>.
- [14] "Nylon Insert Lock Nuts," Albany County Fasteners. [Online]. Available: https://www.albanycountyfasteners.com/All-Categories/Nylon_Insert_Lock_Nuts#:~:text=A%20nylon%20insert%20lock%20nut,not%20cut%20into%20the%20nylon.
- [15] "Aluminum Structural Framing Systems - T-Slot," AngleLock. [Online]. Available: <https://anglelock.com/blog/aluminum-structural-framing-systems-t-slot>.
- [16] "What is a Raspberry Pi?" Raspberry Pi. [Online]. Available: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>.
- [17] "Raspberry Pi 4 GPIO Pinout, Specifications, and Schematic," eTechnophiles. [Online]. Available: <https://www.etechnophiles.com/raspberry-pi-4-gpio-pinout-specifications-and-schematic/>.
- [18] "DataSlayer," YouTube. [Online]. Available: https://www.youtube.com/watch?v=jRKgEXiMtns&t=692s&ab_channel=DataSlayer.

[19] "ADS1115 Module with Programmable Gain Amplifier," Components101. [Online]. Available: <https://components101.com/modules/ads1115-module-with-programmable-gain-amplifier>.

[20] "Design for Manufacturing and Assembly (DFMA)." Fractory. [Online]. Available: <https://fractory.com/design-for-manufacturing-and-assembly-dfma/>

11. Appendix

CODE FOR PICTURES

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# Read image from Google Drive
img = cv2.imread('') #copy file path of the picture
cv2_imshow(img)

# Define the rotation angle in degrees (uncomment if needed)
#angle = 22

# Calculate the image center (uncomment if needed)
#height, width = img.shape[:2]
#center = (width // 2, height // 2)

# Perform the rotation (uncomment if needed)
#rotation_matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
#img = cv2.warpAffine(img, rotation_matrix, (width, height))
cv2_imshow(img)

# Convert image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Apply Canny edge detection
edges = cv2.Canny(gray, 20, 200, apertureSize=3)

# Apply Hough Line Transform
lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)
print(lines)

line1 = []

# Draw detected lines on the original image
for line in lines:
    rho, theta = line[0]
    a = np.cos(theta)
    b = np.sin(theta)
```

```

x0 = a * rho
y0 = b * rho
x1 = int(x0 + 1000 * (-b))
y1 = int(y0 + 1000 * (a))
x2 = int(x0 - 1000 * (-b))
y2 = int(y0 - 1000 * (a))
cv2.line(img, (x1, y1), (x2, y2), (0, 0, 255), 2)
cv2.imshow(img)

#(uncomment if needed)
# if str(round(theta, 5)) != (str(round((angle*np.pi)/180, 5))):
#     if str(round(theta, 5)) != str(round(np.pi/2 - ((angle*np.pi)/180),
5)):
#         if str(round(theta, 5)) != str(round((angle*np.pi)/180 + np.pi/2,
5)):
#             if str(round(theta, 5)) != str(round(np.pi -
((angle*np.pi)/180), 5)):
#                 if str(round(theta, 5)) != (str(round(np.pi +
((angle*np.pi)/180), 5))):
#                     if str(round(theta, 5)) != str(round(np.pi/2*3 -
((angle*np.pi)/180), 5)):
#                         if str(round(theta, 5)) != str(round(np.pi/2*3 +
((angle*np.pi)/180), 5)):
#                             if str(round(theta, 5)) != str(round(np.pi*2 -
((angle*np.pi)/180), 5)):
#                                 line1.append(rho)
#                                 print(line1)

# Calculate the distance between the two detected lines
print (line)
distance = abs(line1[0] - line1[1])

# Display the result
cv2.imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()

print("Width of wire:", distance, "pixels")
print("Width of wire:", distance*0.00732142857, "mm")

```

CODE FOR RECORDED VIDEO

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# Read video from Google Drive
video_path = '' #copy file path of the video
cap = cv2.VideoCapture(video_path)

# Get video properties
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)
duration = frame_count / fps

# Print video properties
print("Total Frames:", frame_count)
print("FPS:", fps)
print("Duration (seconds):", duration)

# Define the rotation angle in degrees (uncomment if needed)
#angle = 113

# Define a list to store line distances
line_distances = []
line_og = []
line_final = []

while True:
    # Read each frame of the video
    ret, frame = cap.read()

    if not ret:
        break # Break the loop if there are no more frames

    #cv2_imshow(frame)

    # Convert frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply histogram equalization to enhance contrast
```

```

#frame_equalized = cv2.equalizeHist(gray)

# Apply Canny edge detection
edges = cv2.Canny(frame, 50, 150, apertureSize=3)
#print(edges)

# Apply Hough Line Transform
lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)
#print ('line =', lines)

if lines is None or len(lines) < 2:
    continue

line_distances = []

for line in lines:
    rho, theta = line[0]
    #print (rho, theta, line)
    a = np.cos(theta)
    b = np.sin(theta)
    x0 = a * rho
    y0 = b * rho
    x1 = int(x0 + 1000 * (-b))
    y1 = int(y0 + 1000 * (a))
    x2 = int(x0 - 1000 * (-b))
    y2 = int(y0 - 1000 * (a))
    cv2.line(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)

    line_distances.append(rho)
    # print ("line distance = ", line_distances)
line_final.append(line_distances)

# Display the frame with detected lines
cv2.imshow(frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break # Break the loop if 'q' is pressed

cap.release()
cv2.destroyAllWindows()

# Calculate the distance between the two detected lines
distance_final = []
for j in line_final:
    #print(j)

```

```
#print (line_final)
distance = abs(j[0] - j[1])
#print (line_distances[0], " ", line_distances[1])
distance_final.append(distance)

for index, distance in enumerate(distance_final):
    print("Distance", index+1, ":", distance, "pixels")
    print("Width of wire", index+1, ":", distance * 0.00732142857, "mm")
    print() # Print an empty line to create space between each index
```

CODE FOR LIVE VIDEO FOR MICROSCOPE USING TEENSY ARDUINO

```
!pip install pyserial

import cv2
import numpy as np
import serial
from google.colab.patches import cv2_imshow
from google.colab import drive

# Connect to the Teensy Arduino
ser = serial.Serial("COM6", 9600) # Replace <serial_port> and <baud_rate>
with your Arduino's port and baud rate

# Capture video from USB camera
cap = cv2.VideoCapture(0) # Use the appropriate camera index (0 or 1) if
you have multiple cameras connected

while True:
    # Read each frame of the video
    ret, frame = cap.read()

    if not ret:
        break # Break the loop if there are no more frames

    # Convert frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply Canny edge detection
    edges = cv2.Canny(gray, 50, 150, apertureSize=3)

    # Apply Hough Line Transform
    lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)

    if lines is None or len(lines) < 2:
        continue

    line_distances = []
    for line in lines:
        rho, theta = line[0]
        line_distances.append(rho)

    # Calculate the distance between the two detected lines
    distance = abs(line_distances[0] - line_distances[1])
    width_of_wire_mm = distance * 0.00732142857
```

```

print("Width of wire :", width_of_wire_mm, "mm")
print() # Print an empty line to create space between each index

# Send the width of the wire to the Teensy Arduino
ser.write(str(width_of_wire_mm).encode())

# Display the frame with detected lines
cv2.imshow('Frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break # Break the loop if 'q' is pressed

cap.release()
cv2.destroyAllWindows()

```

CODE FOR THE RASPBERRY PI CAMERA USING RASPBERRY PI

```

import picamera2
from libcamera import controls
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math
from sklearn.cluster import KMeans
import time

# Create a Picamera2 instance
cam = picamera2.Picamera2()
cam.start(show_preview=True)
cam.set_controls({"AfMode": controls.AfModeEnum.Manual, "LensPosition":
100})

# Create a named window
cv2.namedWindow("Lines", cv2.WINDOW_AUTOSIZE)
# Move the window to a specific position
cv2.moveWindow("Lines", 100, 100)

while True:
    # Capture an image into the stream
    image = cam.capture_array("main")

    # Apply Canny edge detection
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Canny edge detection

```

```

low_threshold = -20
high_threshold = 40
edges = cv2.Canny(gray, low_threshold, high_threshold)

# Apply Hough Line Transform
rho = 1 # Distance resolution in pixels of the Hough grid
theta = 0.2* np.pi / 180 # Angular resolution in radians of the Hough
grid
threshold = 120 # Minimum number of votes (intersections in Hough
grid cell)

# Run Hough on edge detected image
# Output "lines" is an array containing endpoints of detected line
segments
lines = cv2.HoughLines(edges, rho, theta, threshold)

if lines is not None:
    for i in range(0, len(lines)):
        rho = lines[i][0][0]
        theta = lines[i][0][1]
        a = math.cos(theta)
        b = math.sin(theta)
        x0 = a * rho
        y0 = b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
        cv2.line(image, pt1, pt2, (0,0,255), 3, cv2.LINE_AA)

    lines_array = []
    for i in range(len(lines)):
        lines_array += [[lines[i][0][0],lines[i][0][1]]]

    # K-mean cluster algorithm depending on rho to end up with two
lines
    kmeans = KMeans(n_clusters=2, random_state=0,
n_init="auto").fit(lines_array)

    # Display the image
    cv2.imshow("Lines", image)

    # Compute distance between the two lines
    distance =
abs(math.sin(kmeans.cluster_centers_[0][1])*kmeans.cluster_centers_[0][0]
-

```

```
        math.sin(kmeans.cluster_centers_[1
][1])*kmeans.cluster_centers_[1][0])
        width_of_wire_mm = distance * 0.0230659188788095

        # Perform further processing or actions with the width_of_wire_mm
value
        print("Diameter : ", width_of_wire_mm,"mm")

    else:
        print("No fiber is seen")

# Close the camera when done
camera.close()
```

CODE FOR THE INTEGRATION OF THE FRED SYSTEM

```
#Import Libraries
import time
import math
import board
import busio
import RPi.GPIO as GPIO
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
from gpiozero import Motor
from gpiozero import RotaryEncoder
from time import sleep
import atexit

#####
#####
##Start of user changeable section

# Spooling (Stepper) Motor Settings
direction = 1 #Clockwise = 1 or Anticlockwise = 0
microstepping = '1'      # Enter '1' , '1/2', '1/4', '1/8', '1/16' or
'1/32', with colons
delay = 95              #Delay between steps, in micro second

# Heater Settings
targetTemp = 100        #degree celcius

# Extruder (DC) Motor Settings
dcduty = 20             #duty cycle from 0 to 100%

##End of user changeable section
#####
#####
# DO NOT CHANGE THE FOLLOWING

# Function to cleanup GPIO and set all outputs to low
def cleanup_gpio():
    GPIO.output(extruderStep, GPIO.LOW)
    GPIO.output(heaterPin, GPIO.LOW)
    p.ChangeDutyCycle(0)
    print('Resetting Pins')
    # Add more lines for other GPIO pins as needed
    GPIO.cleanup()
```

```

#GPIO Pin Definitions
extruderDirection = 13    # Extruder Direction Pin
extruderStep = 6         # Extruder Step Pin
encoderPin1 = 23         # DC Motor Encoder Pin 1
encoderPin2 = 24         # DC Motor Encoder Pin 2
dcmotorPin = 5           # DC Motor PWM Pin
heaterPin = 21           # Heater Pin
M0 = 17                 # Microstepping Pin 1
M1 = 27                 # Microstepping Pin 2
M2 = 22                 # Microstepping Pin 3

#Initialise GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(heaterPin, GPIO.OUT)
GPIO.setwarnings(False)
GPIO.setup(extruderDirection, GPIO.OUT)
GPIO.setwarnings(False)
GPIO.setup(extruderStep, GPIO.OUT)
GPIO.setwarnings(False)
GPIO.setup(dcmotorPin, GPIO.OUT)
GPIO.setwarnings(False)
GPIO.setup((M0, M1, M2), GPIO.OUT)
GPIO.output(extruderDirection, direction)

# Initialise DC motor encoder
encoder = RotaryEncoder(encoderPin1, encoderPin2, max_steps=0)

#Analogue to Digital (ADC) initialisation for thermistor
i2c = busio.I2C(board.SCL, board.SDA)    # Create the I2C bus
ads = ADS.ADS1115(i2c)                  # Create the ADC object using the
I2C bus
chan = AnalogIn(ads, ADS.P0)            # Create single-ended input on
channel 0

# Thermistor Constants
RT0 = 100000    # Ω
T0 = 298.15     # K
B = 3977        # K
VCC = 3.3       # Supply voltage
R = 10000       # R=10KΩ

#Stepper motor initialisation
SPR = 200                                # Steps per Revolution, from
Stepper Data Sheet

```

```

RESOLUTION = {'1': (0, 0, 0), # (M0, M1, M2)
              '1/2': (1, 0, 0),
              '1/4': (0, 1, 0),
              '1/8': (1, 1, 0),
              '1/16': (0, 0, 1),
              '1/32': (1, 0, 1)} #microstepping settings
FACTOR = {'1': 1,
          '1/2': 2,
          '1/4': 4,
          '1/8': 8,
          '1/16': 6,
          '1/32': 32} #microstepping settings
#*****
#standarddelay = 1/SPR/FACTOR[microstepping] #standard delay is the
amount of time it takes to rotate by one
delay = delay/SPR/FACTOR[microstepping] #convert input from seconds
to microseconds
#delay = delay/1000
#delay = 0.00001/32
GPIO.output((M0, M1, M2), RESOLUTION[microstepping])

# DC Motor initialisation
ppr = 300.8 # Pulses Per Revolution of the encoder
dcfreq = 1000 #DC motor PWM frequency
tsample = 0.1 # Sampling period for code execution (s)

# Initializing previous values and starting main clock
tstart = time.perf_counter() #start internal clock
oldtime = 0 #old DC time
oldpos = 0 #old DC position
lasttime = 0 #old stepper time

#Start DC motor
p = GPIO.PWM(dcmotorPin, dcfreq)
p.start(dcduty)

#Capture video from the camera
cap = cv2.VideoCapture(0) # Use the appropriate camera index (0 or 1) if
you have multiple cameras connected

#Loop Execution
while True:
# print (time.perf_counter())

```

```

#Simple heater controls
VR = chan.voltage
RT = (VCC - VR) * R / VR
ln = math.log(RT / RT0)
TX = (1 / ((ln / B) + (1 / T0)))
TX = TX - 273.15

if GPIO.input(heaterPin) == GPIO.LOW and TX < 1.007 * targetTemp:
    GPIO.output(heaterPin, GPIO.HIGH)
    pass
elif GPIO.input(heaterPin) == GPIO.HIGH and TX > 0.795 * targetTemp:
    GPIO.output(heaterPin, GPIO.LOW)
print("Temp = {:>5.2f} C".format(TX))

# Pausing for `tsample` to give CPU time to process encoder signal
#time.sleep(tsample)

#Stepper motor controls
GPIO.output(extruderStep, GPIO.LOW)
if (time.perf_counter()-lasttime) > delay and TX > targetTemp - 10:
    GPIO.output(extruderStep, GPIO.HIGH)
    lasttime = time.perf_counter()
    #print (time.perf_counter())

#Calculation of DC motor speed
dt = time.perf_counter()-oldtime
ds = encoder.steps - oldpos
rpm = ds/ppr/dt*60
print("DC Speed = {:0.2f} rpm".format(rpm))

# Read each frame of the video
ret, frame = cap.read()

if not ret:
    break # Break the loop if there are no more frames

# Convert frame to grayscale
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Apply Canny edge detection
edges = cv2.Canny(gray, 50, 150, apertureSize=3)

# Apply Hough Line Transform
lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)

```

```

if lines is None or len(lines) < 2:
    continue

line_distances = []
for line in lines:
    rho, theta = line[0]
    line_distances.append(rho)

# Calculate the distance between the two detected lines
distance = abs(line_distances[0] - line_distances[1])
width_of_wire_mm = distance * 0.00732142857

# Perform further processing or actions with the width_of_wire_mm
value
print("Diameter : ", width_of_wire_mm, "mm")

if cv2.waitKey(1) & 0xFF == ord('q'):
    break # Break the loop if 'q' is pressed

# Display the frame with detected lines
cv2.imshow('Live Video', frame)

oldtime = time.perf_counter()
oldpos = encoder.steps

cap.release()
cv2.destroyAllWindows()
GPIO.cleanup()

# Register the cleanup_gpio function to be called on script exit
#atexit.register(cleanup_gpio)

```

BILL OF MATERIALS FOR 2023 FRED SYSTEM

Part Name	Quantity (#)	Unit Cost (\$)	Total Cost (\$)	Link
Fiber Extrusion				
Stepper Motor	1	5.998	5.998	https://a.co/d/9AJc2ON
M3-10 screw	3	0.06	0.18	https://www.mcmaster.com/91292A113/
M3-25 screw	1	0.0849	0.0849	https://a.co/d/iScHn65
M5-35 screw	1	0.35	0.35	https://www.amazon.com/M3-0-5-Socket-Screw-18-8-Stainless-Quantity/dp/B082GFM6NP?th=1
Spring	1	0.745	0.745	https://a.co/d/7ghEXid
M3-12 screw	1	0.06	0.06	https://www.mcmaster.com/91292A114/
Extrusion Support	1	0.47	0.47	
Lever	1	0.15	0.15	
PC alloy block	1	0.05	0.05	
Heat Block	1	1.798	1.798	https://a.co/d/4M23ZK6
Thermistor and Heater	1	2.158	2.158	https://a.co/d/i3ZMrWw
Thumb screws (new M3-12)	2	0.06	0.12	
Driver gear	1	0.4995	0.4995	https://a.co/d/j8I842w
Idler pulley	1	0.025	0.025	
M5 lock nut	1	0.052	0.052	https://www.mcmaster.com/90631A113/
M3 inserts	2	0.0899	0.1798	https://a.co/d/cQlhOpe
M5 inserts	2	0.1798	0.3596	https://a.co/d/6IINI1D
M3-22 screw	2	0.0929	0.1858	https://a.co/d/diN2YGr
Wire protector	1	0.7	0.7	https://a.co/d/gm09Zzx
Cooling				
Fan support	2	0.37	0.74	
M3-18 screw	4	0.09	0.36	https://www.amazon.com/M3-0-5-Socket-Screws-Stainless-Quantity/dp/B084R9BFF8?th=1
M3 inserts	6	0.0899	0.5394	https://a.co/d/cQlhOpe

M5 inserts	4	0.1798	0.7192	https://a.co/d/6IINI1D
Centrifugal fan	2	4.495	8.99	https://a.co/d/dCT0rvs
M3-8 screw	2	0.0545	0.109	https://www.mcmaster.com/91292A112/
Thumb screws (new M3-12)	4	0.06	0.24	https://www.mcmaster.com/91292A114/
Diameter measurement				
Adjustable Side Camera Mount base	1	0.32	0.32	
Adjustable Side Camera Pulley Mount base	1	0.22	0.22	
Adjustable Side Camera Mount Camera Slider	2	0.15	0.3	
Adjustable Side Camera Mount Single Side pulley	1	0.05	0.05	
Adjustable Side Camera Mount Camera Holder		0.27	0	
Card	1	0.01	0.01	https://a.co/d/dv2HPKu
Adjustable Side Camera Mount Card Holder	1	0.27	0.27	
Adjustable Side Camera Mount Pulley Slider	2	0.1	0.2	
Pulley	3	0.657	1.971	https://a.co/d/i0kWKFB
M3 inserts	9	0.0899	0.8091	https://a.co/d/cQlhOpe
M5 inserts	2	0.1798	0.3596	https://a.co/d/6IINI1D
M3-5 screws	9	0.0978	0.8802	https://a.co/d/dAwrWr5
Thumb screws (new M3-12)	2	0.06	0.12	https://www.mcmaster.com/91292A114/
USB Camera	1	16.99	16.99	https://a.co/d/exQ7VjY
Fiber Collection				
Modular Plate	1	1.51	1.51	https://www.mcmaster.com/8589K925/
Main Axle Plate	2	0.5	1	https://www.mcmaster.com/8589K83/
DC Motor Mount	1	0.22	0.22	
Scotch Yolk Wheel	1	0.19	0.19	

Rotary shaft_L80	1	0.399667	0.399667	https://a.co/d/c2MCukr
Slider	1	0.37	0.37	
Round Tube Stock_L185	1	0.62	0.62	https://www.mcmaster.com/8585K828/
Spool	1	0.73	0.73	
Spool Cap	1	0.06	0.06	
Axle Gear_D25	1	0.07	0.07	
Spur Gear Stock_D20x62	1	0.44	0.44	
Timing Belt	1	1.088	1.088	https://a.co/d/cFZD4mD
Rotary Shaft_D5_L20	1	0.099917	0.099917	https://a.co/d/j99P2Ra
Timing Belt Pulley	2	0.798	1.596	https://a.co/d/019p4rD
Plastic Miter Gear	1	3.66	3.66	https://www.mcmaster.com/2810N2/
Modified Miter Gear	1	0.09	0.09	
Plastic Spur Gear_D28	1	0.08	0.08	
Round Tube Stock_L14.5	1	0.05	0.05	https://www.mcmaster.com/8585K828/
Cover	1	0.94	0.94	
DC motor	1	16.88	16.88	https://a.co/d/ax5uzKI
M5-16 screws	2	0.13	0.26	https://www.mcmaster.com/91292A126/
M3 inserts	1	0.0899	0.0899	https://a.co/d/cQlhOpe
M3-8 screw	2	0.05	0.1	https://www.mcmaster.com/91292A112/
Frame				
Back	1	2.11	2.11	https://www.amazon.com/dp/B0BRY5JF5B?ref_cm_sw_r_cp_ud_dp_VX0RRWV5DS6M0NMKY4EE&th=1
Rib	1	0.66	0.66	https://www.amazon.com/dp/B0BRY5JF5B?ref_cm_sw_r_cp_ud_dp_VX0RRWV5DS6M0NMKY4EE&th=1
Base	1	2.31	2.31	https://www.amazon.com/dp/B0BRY5JF5B?ref_cm_sw_r_cp_ud_dp_VX0RRWV5DS6M0NMKY4EE&th=1
M5-14 screws	5	0.15	0.75	https://www.mcmaster.com/91292A058/
M5 nuts	5	0.0977	0.4885	https://www.mcmaster.com/91828A241/
Leg support	2	0.55	1.1	

Rubber feet	4	0.0459	0.1836	https://a.co/d/1aoFxsB
M5-12 screw	4	0.0949	0.3796	https://a.co/d/gW5Qmrl
M5 inserts	2	0.1798	0.3596	https://a.co/d/6IINI1D
Electronics				
PCB	1	11.49	11.49	https://a.co/d/7YdRR4f
Raspberry pi 4 1 GB	1	35	35	https://www.canakit.com/raspberry-pi-4.html
Power Supply	1	17.998	17.998	https://a.co/d/apv614G
Switch	1	0.599	0.599	https://a.co/d/32k0aLN
FF Standoff (M2.5x12)	4	0.03	0.12	https://www.amazon.com/Standoff-Plastic-Spacer-Pillar-Female/dp/B093W8QXBN/ref=sr_1_5?crid=2TPZ5X80BFZVT&keywords=Female%2Bto%2BFemale%2Bhex%2Bplastic%2Bstand%2Boff%2B(M2.5x12)&qid=1691719540&s=industrial&srefix=female%2Bto%2Bfemale%2Bhex%2Bplastic%2Bstand%2Boff%2Bm2.5x12%2B%2Cindustrial%2C111&sr=1-5&th=1
MF Standoff (M3x10)	4	0.045	0.18	https://a.co/d/1wYYDVk
M3-6 philips screw	4	0.06	0.24	https://a.co/d/7HHktcf
M2.5-6 philips screw	8	0.0799	0.6392	https://a.co/d/8Xu1IoO
Drawer	1	1.3	1.3	
M3 inserts	4	0.0899	0.3596	https://a.co/d/cQlhOpe
Plug connector	1	1.37375	1.37375	https://a.co/d/bP0M2OO
Stepper Motor Driver	1	1.999	1.999	https://a.co/d/ehswdr4
	167		158.5464	