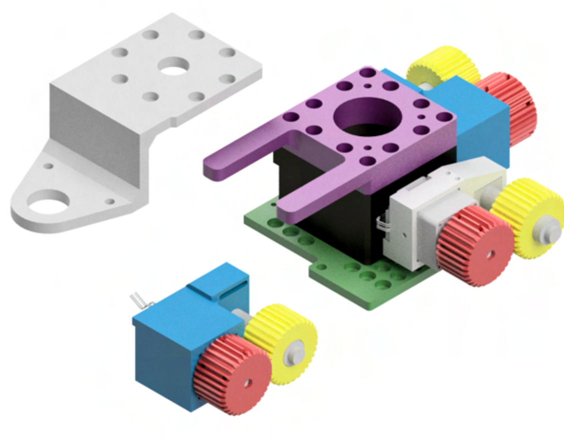


Low-Cost High-Precision XYZ Stage 2022

Project Document

Logan York, Anthony Bryan, Aleksi Hieta

Table of Contents

1. Overview

1.1. Executive Summary

The purpose of this project is to create a motorized XYZ stage that can be used for semiconductor inspection. In essence, this includes a few major pieces. Firstly, the project includes the mechanical stage itself which can move with 10 micron accuracy in the X, Y, and Z directions. Additionally, a microscope/camera setup is included along with the appropriate image processing software which allows for high-resolution mosaic images. Finally, a user interface has been created in order for a user to control the stage and take their desired pictures.

Currently, the project is in the build phase. The team is utilizing existing resources and has become familiar with the materials provided by the project partner. We have been pointed to a few open-source projects which implement solutions with varying levels of documentation. Additionally, many of the mechanical components necessary were provided by the project partner, including manual stages to modify, stepper motors, 3D printed adapters and gears, as well as microcontroller boards and motor drivers. So, the team is currently in the process of working with these mechanical pieces and implementing the necessary software for stage motion, image processing, and user interface solutions.

The majority of the mechanical components have been designed and tested, and the team is currently working on finalizing these pieces. The 3D printed adapters and gears have been fixed to the stage, and their range of motion has been tested. Reworking of the gears in CAD software has taken place and prototyping of new gears/adapters has been done including adding motion in the z-direction. Additionally, the team has gained familiarity with G-code, which allows for specifying stage motion in a way that can be interpreted by GRBL firmware, an open-source firmware which translates these G-code commands into consistent motor motion. In order to properly calibrate this firmware and set the number of motor steps/mm, a precise dial gauge has been purchased and utilized. Limit switches have also been purchased, implemented into the design of the 3D printed adapters, and tested fully. Most of the design of the user interface and image processing has been done and tested. Additionally, an input controls PCB has been designed which allows for the stage to be controlled by endless encoders as well as limit switch integration.

1.2. Team Contacts and Protocols

Below is the contact info for all team members:
Logan York: yorklo@oregonstate.edu, (760) 505-1757
Anthony Bryan: bryanan@oregonstate.edu, (541) 870-2476
Aleksi Hieta: hietaa@oregonstate.edu, (925) 322-3045

Table I
Protocols and Standards

Topic	Protocol	Standard
Open Communication between Team Members	All individuals will respond within 3 days to ensure messages are received in a timely fashion	During weekly meetings communication can be evaluated by the team and if a given individual must improve their communication performance.
Accountability	Individuals will hold themselves accountable for assigned work, issues, or absences.	Team members will state issues encountered in weekly meetings when they arise. A problem thread may be present on the main Discord server for more immediate communication.
Coding	Software created will have clear and descriptive naming conventions with adequate commenting.	Each function will have standardized headers to explain functionality and I/O.
Timeliness	Assignments will be completed by agreed upon deadlines.	Incomplete assignments by an individual will have help from other team members to complete tasks more quickly when applicable.
Management	Management of other team members by an individual will be conducted in a respectful manner.	Formal conversation will be utilized when needed and respectfully listening without interruption will be present.

1.3. Gap Analysis

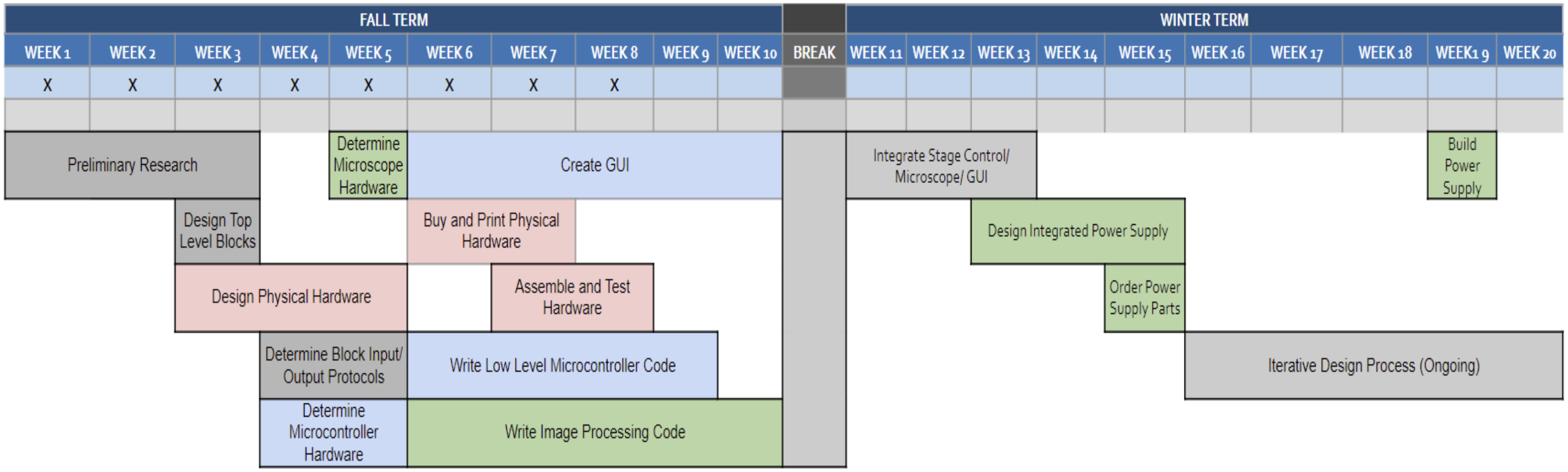
High-precision XYZ stages are currently available for purchase with the desired 1 micron accuracy that this project is shooting for. However, they cost upwards of $30,000. Therefore, the main gap that this project fills with respect to the mechanical stage itself is in the “low-cost” portion of the project. The typical user of this project would likely be a student without the resources or experience to operate such an expensive piece of equipment. Our project would ideally be as mechanically sound and accurate as the high-end counterparts while providing excellent documentation for a cheaper implementation.

The basic idea for how the project will actually be used is for semiconductor inspection. Specifically, the end product could be used to take high-resolution images of a PCB or silicon die. When working with the high magnifications necessary for such inspection, if a surface is tilted even slightly and only a single image is taken, the result is a blurred image. However, by taking a series of images while varying the z-axis of the stage then merging the images to perform depth composition, a sharp image can be created. Thus, through the automated image processing portion of the project, clear and complete images of a chip can be created. These images can then be used for hardware security applications for example. Of course, all of this functionality from cheap, precise physical stage motion to creation of razor sharp images could be performed simply through a robust user interface.

1.4. Timeline/Proposed Timeline

Attached is a simplified timeline of the entire project.

PROJECT TITLE	Low-cost high-precision motorized stage	GROUP #	4
PROJECT MANAGER	Vincent Immler	GROUP MEMBERS	Anthony Bryan, Aleski Hieta, Logan York



Key:

PRIMARY BLOCK OWNER	Group	Aleski	Logan	Anthony
COLOR CODE				

1.5. References and File Links

1.5.1. References (IEEE)

B. Diederich, "OpenUC2/UC2-micronstage," *GitHub*, 24-Jun-2021. [Online]. Available: https://github.com/openUC2/UC2-MicronStage. [Accessed: 18-Nov-2022].

B. Deiderich, "OpenUC2/UC2-motorized-xy-table" *GitHub*, 03-Dec-2021. [Online]. Available: https://github.com/openUC2/UC2-Motorized-XY-Table. [Accessed: 18-Nov-2022].

"Helicon Focus and Focus Stacking," *Helicon Focus - Helicon Soft*. [Online]. Available: https://www.heliconsoft.com/heliconsoft-products/helicon-focus/. [Accessed: 18-Nov-2022].

1.5.2. File Links

Weekly Timeline for Low-cost high-precision motorized stage

1.6. Revision Table

10/13/2022	Aleksi Hieta: Added draft references and team contacts and protocols
11/4/2022	Logan York: Revised executive summary
11/17/2022	Anthony Bryan: Revised Timeline and fixed links to timeline and references
11/17/2022	Logan York: Added latest details to executive summary
11/17/2022	Aleksi Hieta: Revised protocol table formatting

2. Impacts and Risks

2.1. Design Impact Statement

The XYZ stage is a multipurpose viewing platform for microscopic items. It allows for viewing microchips or biomedical supplies with micrometer precision using a high quality camera. The impacts of this stage are vast as its application can be in a variety of industries from manufacturing to chip research. With this in mind, the low-cost aspect and open source nature of the design also allows it to be implemented in far more fields than a high end unaffordable option. The purpose of this report is to take into consideration potential system-level impacts in a variety of areas in order to adequately address any potential concerns that this project may affect. Additionally, this report helps to inform stakeholders of any subtle limitations or benefits of the design. The areas that will be considered in this report include: public health, safety, and welfare impacts; cultural and social impacts; environmental impacts; as well as economic factors.

The open source nature of this project allows anyone to access the files necessary to recreate it in order to increase accessibility to a low cost, automated microscope stage. A Public Health, Safety, and Welfare Impact concern is the fact that this stage, used in tandem with an electromagnetic probe could exploit hardware insecurities.
A growing concern in the cybersecurity community is the use of electromagnetic probes to target the binary states that are processed directly on computer chips. Such exploits directly affect the public health and safety impacts of cyber security. For example high-resolution EM probes can exploit the routing of dual rail logic systems built on FPGAs, reducing the security of such systems and necessitating additional security measures to be used[1]. Cybersecurity attacks that exploit at the hardware level are known as invasive attacks and require high technical skills as well as specialized equipment[2].

Invasive attacks are generally not common due to the high barrier of entry and because they require expensive specialized equipment but the Low-Cost High-Precision XYZ Stage designed for this project could be used to control an EM probe which lowers the barrier of entry for commiting invasive attacks. This project will be open source and publicly available which could lead to an increase of invasive attacks as the prohibitive costs of invasive hacking is lowered.

There has already been a sharp increase in the rise of cyber crimes, but almost all of these attacks use non-invasive means such as malware, or social engineering such as phishing[3]. By increasing the tools available to hackers this project poses the risk of increasing an already growing problem. Increasing availability to technology has positive impacts as well, our group predicts that the Low-Cost High-Precision XYZ Stage will be used almost exclusively by research institutions which will further research into hardware security with applications also possible in the biologic and medical fields.

The Cultural and Social Impacts of this project is concentrated on the supply chain acquisitions of the material components needed to keep the project low cost. Attempting to build at the cheapest prices often leads to using exploitative labor forces for sourcing parts. One of the design requirements for the X, Y, Z stage is for the cost to be kept under $300 causing the project to rely on sourcing directly from Chinese manufacturing to acquire the necessary hardware. To illustrate the price difference, an American company building an XY stage similar to one used in the project costs $901.25[4], compared to the Chinese made stage which costs $56.98[5] which is 15 times cheaper. These savings come at a cost to both the cultural and social welfare of the people living in thes low cost production zones caused by both globalization and industrialization. The book "*China's Workers Under Assault: Exploitation and Abuse in a Globalizing Economy"* evaluates case studies and determined that although factory jobs lead to an increase of GDP, it widens income disparity leading to the systemic violation of labor rights for workers[6].

Cultural impacts of industrialization and globalization is a reduction in cultural practices and job diversity. According to the research article "Impact of Rural Industrialization on Village Life and Economy: A Social Accounting Matrix Approach*",* industrialization directly affects the long-standing cultural practices found in rural areas. The article found that large factories reduce the amount of household industries which include jobs such as basket makers, carpenters, potters and many artisanal jobs[7]. Many of these jobs are multi-generational and steeped with cultural history, but are unsustainable when compared to the relatively high wages of factory workers. Because this project directly increases the demand for industrial manufacturing it causes negative impacts on rural culture in recently globalized markets.

The actual operation of a low-cost high-precision XYZ stage is unlikely to have direct environmental impacts due to low power consumption and its main use case being inspection of semiconductor devices. However, in order to create the stage at reduced cost, our team is relying on the use of 3D printing technology. For the potential of mass production of the system, producing a low-cost alternative to traditional motorized stages must take into account the environmental impacts of 3D printing. Producing parts of the system in this way must consider energy consumption, waste management, and air pollution.

Energy consumption is low for printing with PLA (the most common 3D printing material) compared to actual machining of adapters or gears which must go through manufacturing and shipping elsewhere [8]. This factor makes for significantly lower consumption, especially when rapidly prototyping. Additionally, PLA is often made from biological sources such as plant material, which make it mostly carbon neutral. It is also non-toxic and can be made from recyclable materials, which allows for reduced waste compared to traditional manufacturing processes, significantly reducing environmental impacts [8]. However, 3D printed mechanical components such as gears have a shorter lifespan than their machined counterparts, decreasing the overall benefit of the material.

Finally, since 3D printing often allows for production of components closer to consumers, this leads to reduced transportation if the system is ever mass produced, having a considerable positive impact on air pollution [9]. Specifically, this results in a reduction of common harmful vehicle byproducts like nitrogen dioxide, carbon monoxide, hydrocarbons, benzene, and formaldehyde as well as carbon dioxide, the most common human-caused greenhouse gas. Overall, the use of 3D printing to produce essential components for the XYZ stage allows for substantial positive environmental impacts.

The economic impacts of a low-cost high-precision XYZ stage are directly related to the cumulative final price and its difference from other stages on the market. Many stages that would perform a similar task to the one designed in our project could run upwards of thousands of dollars if not more so. Providing an extremely affordable alternative compared to other options would drastically reduce the price of research and development on microchips and/or objects viewed with a high-precision microscope. The stage can also have economic impacts on the private and public sectors of research. Public sectors with much lower budgets would suddenly be able to have the opportunity to inspect equipment for a lower cost than ever before [10]. Likewise, businesses with a higher budget would suddenly have a cheaper alternative that functions in essentially the same way.

Furthermore, the design of a low-cost XYZ stage can threaten the market for more moderate to high end alternatives. Specifically, stages that are currently considered on the low end in the current market would likely have decreasing sales with the advent of a cheaper alternative [11]. With the goal of a cost effective system at the center of the design, the choices for parts, hardware, and even software are limited to produce such. For example, image processing software licenses, controls hardware, and stage motors all lean towards cheaper, and in turn, less precise and reliable modules. With low-cost modules performing precise tasks, it should still be noted that higher cost add-ons could be used in practical applications.

The overall impact of the Low-Cost High-Precision XYZ stage is expected to be low as it is expected to be used by highly technical people for a low cost. Many research institutions which require high precision stages have enough capital to afford the $15,000+ stages that this project is attempting to emulate. This project hopes to expand the use of precision stages at the graduate research level as well as aid the development of burgeoning research institutions in developing nations. The overall tone of this impact assessment was negative, but this was done with a worst case scenario focus and we expect very little negative impact from this project.

2.2. Risk Assessment

Table II

Risk Assessment and Action Plans

Risk ID	Risk Description	Risk category	Risk prob-ability	Risk impact	Performance indicator	Action Plan

R1	Vendor Delay	Schedule	Low	Mid	Shipping delays	Have backups in place and/or research alternative solutions
R2	Incompatible Blocks	Technical	High	High	Blocks are unable to communicate	Rework interfaces between blocks
R3	Breaking Components	Technical	Low	High	Supplied materials do not function	Have backups and/or order new parts immediately
R4	Motors Injure Member	Safety	Low	Low	Unforeseen movement of gears	Assign work to uninjured team members
R5	Lost Documentation	Organiza tional	Low	Mid	Lost references in documents	Recreate necessary documents and extend timeline

2.3. References and File Links

References

[1]V. Immler, R. Specht, and F. Unterstein, “Your rails cannot hide from localized em: How dual-rail logic fails on fpgas—extended version,” *Journal of Cryptographic Engineering*, vol. 8, no. 2, pp. 125–139, 2018.

[2]S. Hamdioui, J.-L. Danger, G. D. Natale, F. Smailbegovic, G. van Battum, and M. Tehranipoor, “Hacking and protecting IC Hardware,” *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014*, Apr. 2014.

[3]H. S. Lallie, L. A. Shepherd, J. R. C. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, “Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic,” Computers & Security, vol. 105, p. 102248, Jun. 2021.

[4]“LX20 self-contained xy 25 mm translation stage, 1/4’-20 taps,” *Thorlabs*. [Online]. Available: https://www.thorlabs.com/thorproduct.cfm?partnumber=LX20. [Accessed: 04-Nov-2022].

[5]“XY Axis 60*60mm Trimming Station Manual Displacement Platform Linear Stage Sliding Table XY60-C XY60-R,XY60-LM LY60 Cross Rail,” *aliexpress.com*. [Online]. Available: https://www.aliexpress.us/item/2255800615306628.html?gatewayAdapt=glo2usa4itemAdapt&_randl_shipto=US. [Accessed: 04-Nov-2022].

[6]A. Chan, *China's workers under assault: Exploitation and abuse in a globalizing economy*. London: Routledge, 2016.

[7]A. Parikh and E. Thorbecke, “Impact of rural industrialization on village life and economy: A Social Accounting Matrix Approach,” *Economic Development and Cultural Change*, vol. 44, no. 2, pp. 351–377, 1996.

[8]Khosravani, & Reinicke, T. (2020). On the environmental impacts of 3D printing technology. Applied Materials Today, 20, 100689–. https://doi.org/10.1016/j.apmt.2020.100689

[9]Sharma, Mondal, S., Mondal, A. K., Baksi, S., Patel, R. K., Chu, W.-S., & Pandey, J. K. (2017). 3D printing: It’s microfluidic functions and environmental impacts. International Journal of Precision Engineering and Manufacturing - Green Technology, 4(3), 323–334. https://doi.org/10.1007/s40684-017-0038-6

[10]L. H. Lim and D. Yang, “High-precision XY stage motion control of Industrial Microscope,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 1984–1992, May 2018. http://eprints.gla.ac.uk/161979/13/161979.pdf

[11]X. Chen, Y. Li, Y. Xie, and R. Wang, “Design and analysis of new Ultra compact decoupled XYZ, 2022. https://www-sciencedirect-com.oregonstate.idm.oclc.org/science/article/pii/S0094114X21002792

[12] E. P. T. Magazine (2022, May 3). “Most electronics supply chain challenges are getting worse.” Electronic Products & Technology. Retrieved November 4, 2022, from https://www.ept.ca/2022/04/most-electronics-supply-chain-challenges-are-getting-worse/

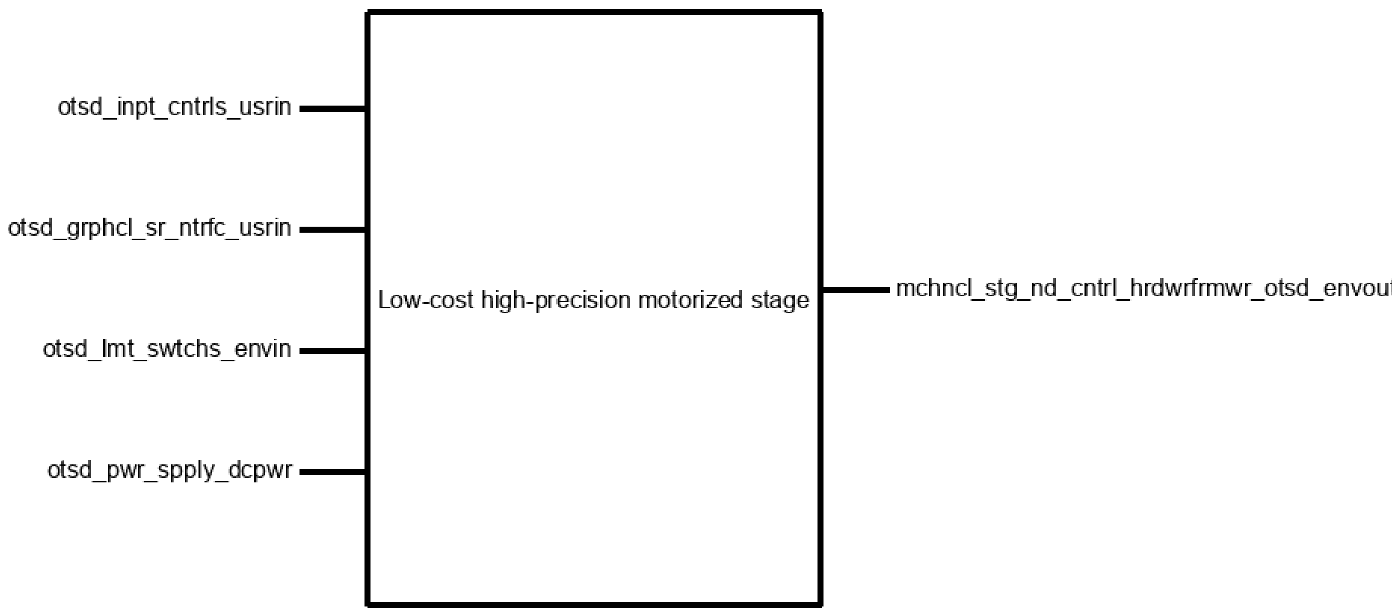
2.4. Revision Table

11/3/2022	Aleksi Hieta: Risk Assessment Table R1-R5
11/4/2022	Logan York: Edited sections, added reference, reworked action plans
4/28/2023	Anthony Bryan: Edited sections, removed headings and numberings to simplify section, changed DIS to better reflect the conclusion of the project.

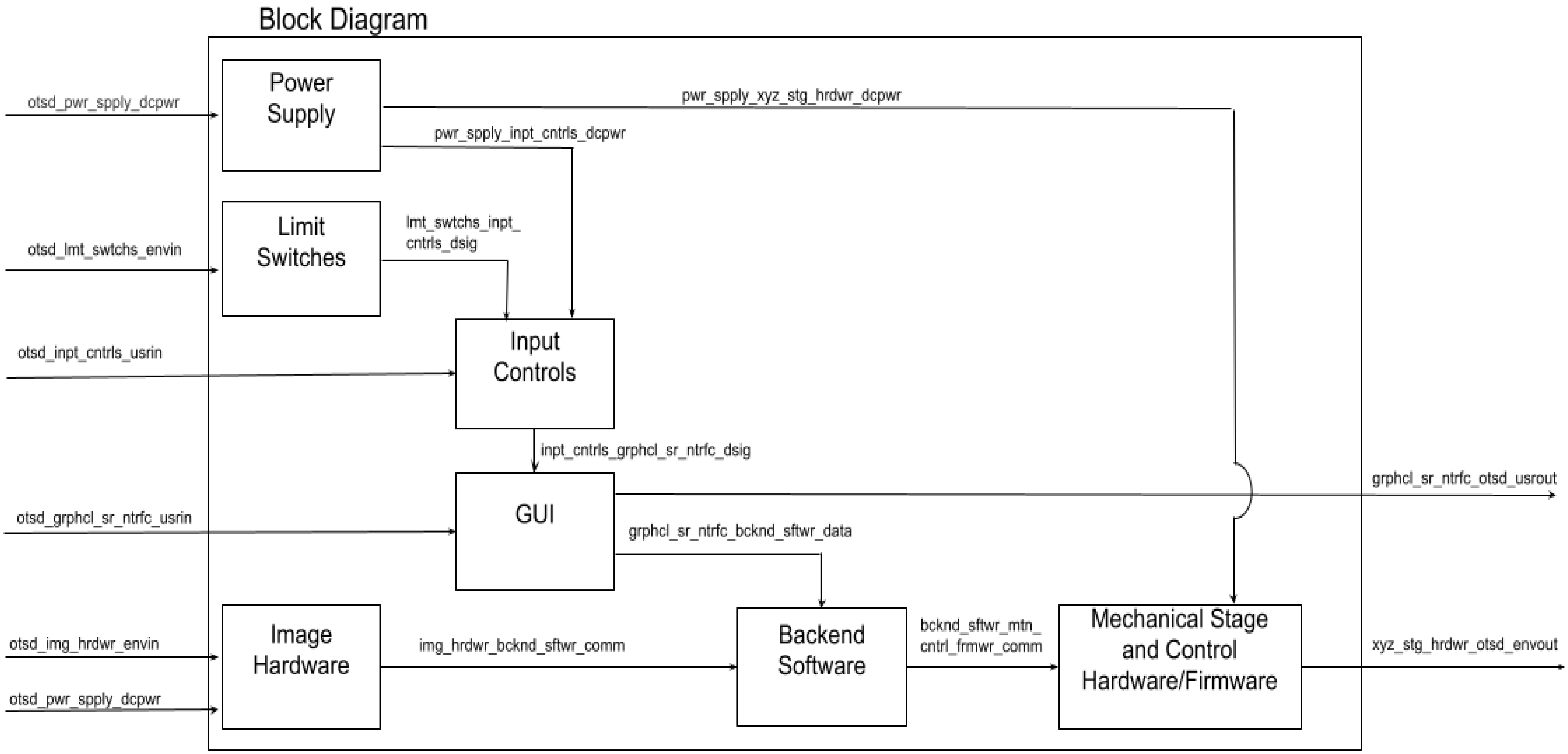
3. Top-Level Architecture

3.1. Block Diagram

3.1.1. Black Box Diagram



3.1.2. Complete Block Diagram



3.2. Block Descriptions

Name	Description
Input Controls Champion: Aleksi Hieta	The input controls will be an array of endless encoders (rotary encoders) to control the X, Y, and Z motion of the stage. Movement of each encoder will be read from an arduino nano and outputted to the GUI for processing.
Graphical User Interface Champion: Aleksi Hieta	GUI will be the central point for taking in inputs from the user through both physical and digital methods. The settings, images, and general controls will all be contained within this block.
Limit Switches Champion: Logan York	Switches that sense when the stage has moved to its limit in any direction in order to stop the motion. The switches will produce a digital signal which will be read in by an arduino nano on the input controls PCB and outputted to the GUI in order to limit the stage motion.
Backend Software Champion: Anthony Bryan	The backend software takes inputs from the GUI to control the image hardware and stepper motors of the X, Y, Z stage so the user can take images of the chip surface. The backend also performs image processing using focus stacking and image stitching to create a clearer image.
Power Supply Champion: Aleksi Hieta	The power supply will power the input controls at 5V to an arduino nano
Mechanical Stage and Control Hardware/Firmware Champion: Logan York	This block includes a manual microscope stage, 3D printed adapters/gears, stepper motors, a control board with motor drivers, and the firmware for this board. It takes in motion commands through a serial interface and produces the physical stage motion corresponding to those commands. In essence, this block includes everything necessary to produce precise stage motion. This is a crucial part of the overall system as the stage must be able to be controlled precisely both manually via the user interface as well as automatically by the imaging process. This block has many parts because it is largely building off of previous work/hardware provided by the project partner and therefore would not be useful to break up into smaller pieces.

Image Hardware Champion: Anthony Bryan	This block consists of the digital microscope used to capture images of the microchip surface on the X, Y, Z stage. Images are captured by a digital sensor which captures IR light through a swappable microscope lens. The sensor is connected to a frame which positions the sensor above the stage.
Image Processing Champion: Anthony Bryan	The image processing block is a code based block that takes multiple images in the form of JPG from the digital microscope and processes them into a single image with improved depth and greater size than a single image from the imaging sensor. It does this through two processes, image stitching and focus stacking. Focus stacking takes multiple images of the same subject with different parts of the image in focus to improve the depth of field which reduces the blurriness of the image. Image stitching takes multiple images from different areas and runs an algorithm to stitch the seams of the images, creating a picture that is multiple times larger than the size of the image sensor could normally produce.

3.3. Interface Definitions

Name	Properties
otsd_inpt_cntrls_usrin	• **Timing:** 50ms pull rate • **Type:** Physical Rotation • **Usability:** Consistent readings more than 95% of the time
otsd_grphcl_sr_ntrfc_usrin	• **Other:** Threaded COM port communication • **Type:** String variable and Buttons • **Usability:** Usable by 90% of students without difficulty
otsd_lmt_swtchs_envin	• **Other:** Max Limit Margin: 1mm (limit switches will activate within 1mm of the physical stage limits) • **Other:** Max Stage Speed: 1mm/sec • **Other:** Max Stage Increments: 0.1mm
otsd_pwr_spply_dcpwr	• **Ipeak:** 2A • **Vmax:** 28V • **Vmin:** 5V
inpt_cntrls_grphcl_sr_ntrfc_comm	• **Datarate:** 9600 • **Messages:** encoded 'utf' • **Other:** Serial Comm
grphcl_sr_ntrfc_bcknd_sftwr_data	• **Messages:** Values • **Other:** Error Handling

	• **Other:** Live Updates
lmt_swtchs_inpt_cntrls_dsig	• **Logic-Level:** Active Low • **Vmax:** 5V (open switch internal voltage) • **Vmin:** 0V (low signal provided from closed switch)
pwr_spply_inpt_cntrls_dcpwr	• **Inominal:** 10mA • **Ipeak:** 100mA • **Vmax:** 5.25V • **Vmin:** 4.35V
pwr_spply_mchncl_stg_nd_cntrl_hrdwrfrmwr_dcpwr	• **Inominal:** 1A • **Ipeak:** 1.5A • **Vmax:** 12.3V • **Vmin:** 11.7V
bcknd_sftwr_img_hrdwr_comm	• **Protocol:** UDP • **Protocol:** Python 3 on Linux • **Protocol:** COM port over USB
bcknd_sftwr_mchncl_stg_nd_cntrl_hrdwrfrmwr_data	• **Datarate:** 115200 baud • **Messages:** G-code commands (Ex: G21 G91 G01 X0.1 F30) • **Other:** Connector: USB Type-A to USB Type-B
img_hrdwr_img_prcssng_data	• **Datarate:** Grayscale • **Datarate:** 1280x1024 pixels • **Messages:** .jpg images

mchncl_stg_nd_cntrl_hrdwrfrmwr_otsd_envout	• **Other:** Max Stage Speed: 1mm/sec (in X, Y, and Z directions) • **Other:** Min Stage Range: 6mm (in X, Y, and Z directions) • **Other:** Motion Resolution: 10um +/- 10% (in X, Y, and Z directions)
img_prcssng_bcknd_sftwr_data	• **Datarate:** Grayscale • **Datarate:** 1160x960 - 3280x2824 pixels • **Messages:** .jpg images • **Other:** 1 image

3.4. References and File Links

3.4.1. References (IEEE)

3.4.2. File Links

3.5. Revision Table

3/12/2023	Logan: Initial section creation. Included content from student portal and updated block diagram.

4. Block Validations

4.1. Mechanical Stage and Control Hardware/Firmware

4.1.1. Description

This block includes a manual microscope stage, 3D printed adapters/gears, stepper motors, a control board with motor drivers, and the firmware for this board. It takes in motion commands through a serial interface and produces the physical stage motion corresponding to those commands. In essence, this block includes everything necessary to produce precise stage motion. This is a crucial part of the overall system as the stage must be able to be controlled precisely both manually via the user interface as well as automatically by the imaging process. This block has many parts because it is largely building off of previous work/hardware provided by the project partner and therefore would not be useful to break up into smaller pieces.

4.1.2. Design

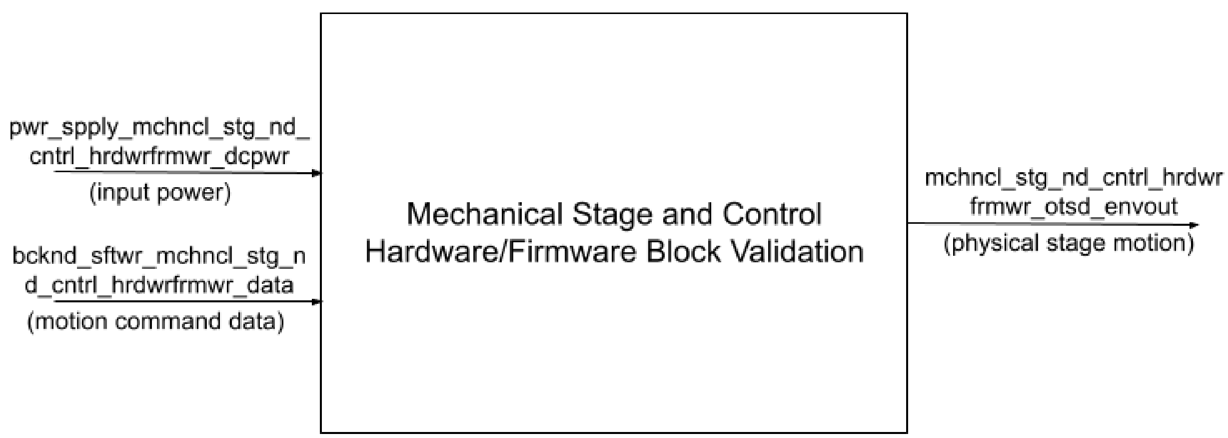


Fig. 1. Black box diagram

The complete design looks like this:

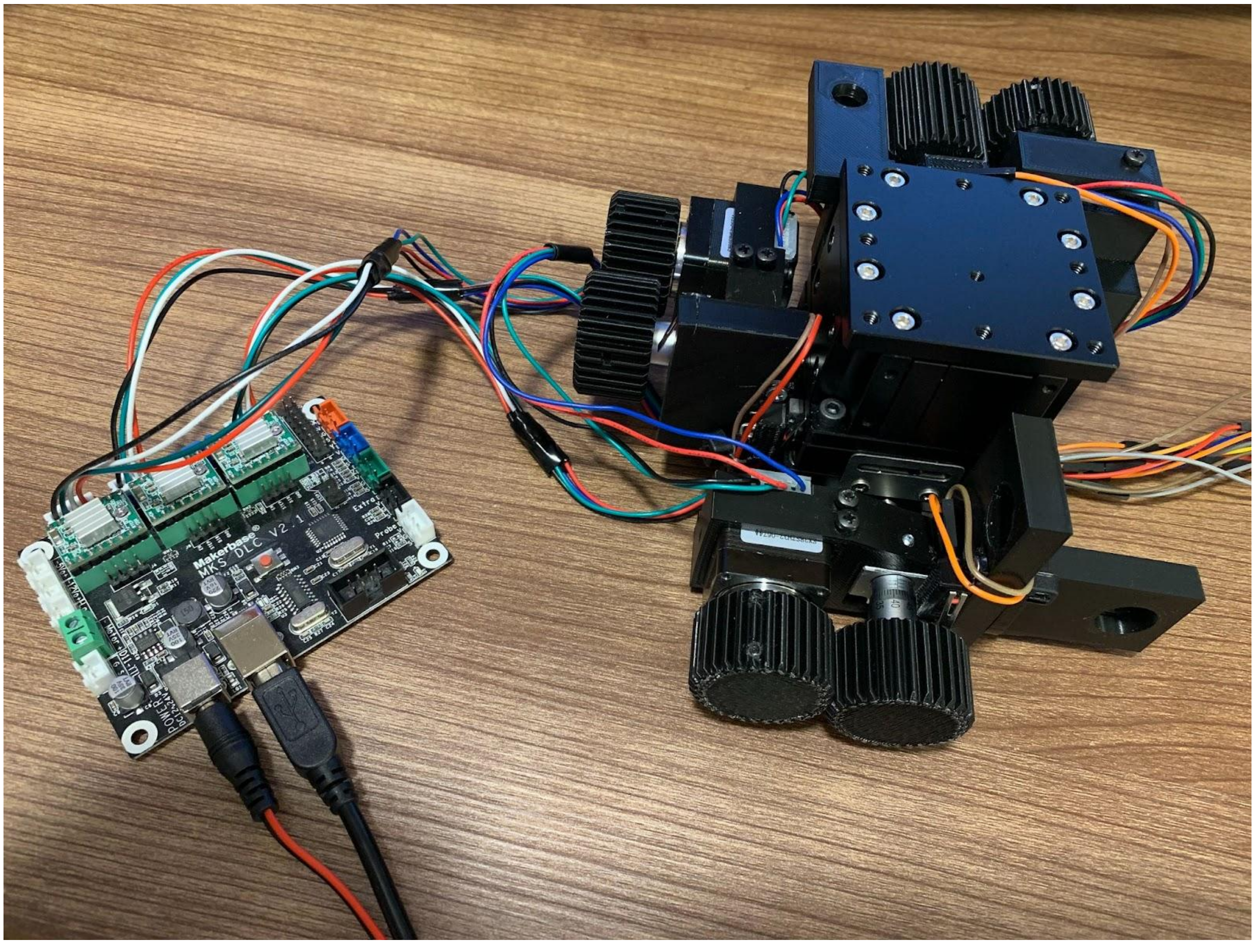


Fig. 2. Complete design

This section will now describe the different parts of the design including: the manual microscope stage, the 3D printed adapters/gears, the stepper motors, the control board with motor drivers, and the firmware for this board.

4.1.2.1. The manual microscope stage

The microscope stage itself was provided by the project partner. The XY part is called: “XY Axis 60*60mm Trimming Station Manual Displacement Platform Linear Stage Sliding Table XY60-C XY60-R,XY60-LM LY60 Cross Rail”[1] and the Z part: ”Z axis 60*60mm Displacement Lift Stage Manual fine tuning platform Double Cross rail Sliding Table LZ60-2 Z60-2”[2] . The manufacturer provides these mechanical drawings:

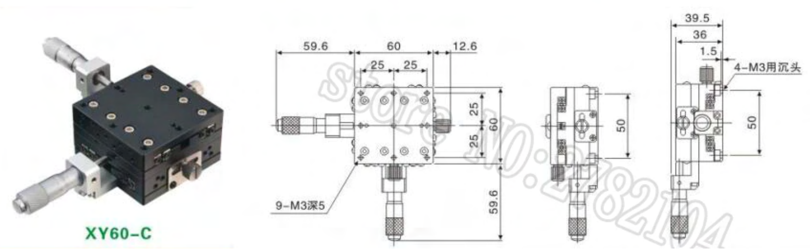


Fig. 3. XY stage mechanical drawings

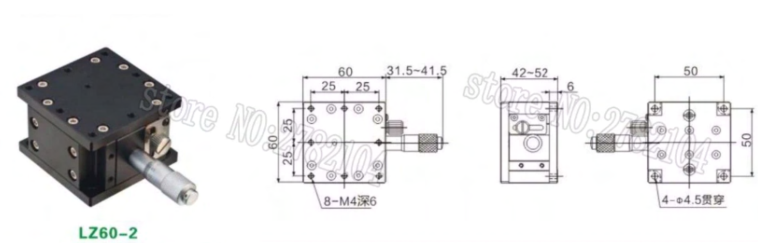


Fig. 4. Z stage mechanical drawings

Additionally, these details are provided in lieu of a datasheet:

Feature:
1. Using mobile cross roller guides, enabling high-precision, smooth
2. The main body made of aluminum alloy, light weight, suitable to be built for other devices

Parameter:
1. Direction of movement: X Y axis direction
2. Body material: Aluminum
3. Drive: differential head
4. The table size: 60x60mm
5. table thickness: 36mm/30mm
6. Stroke: ± 6.5mm (Customizable ±12.5,if you need,contact us)
7. Load: 49N (5kgf)
8. Minimum scale: 0.01mm
9. Accuracy class: 0.01mm
10. Parallelism: 0.06mm
11. Weight: 0.48kg

Fig. 5. Mechanical stage specifications

Most notably, the manufacturer claims 10um precision for the stage.

4.1.2.2. 3D printed adapters/gears

In order to turn the stage knobs with motors, adapters and gears needed to be designed. Initial designs of these parts were provided by the project partner from an open-source project[3] which implemented a motorized XY stage. These parts were then modified in order for the gears to better fit the knobs as well as to allow for the mounting of limit switches and a dial gauge for calibration of the firmware.

X- adapter design:

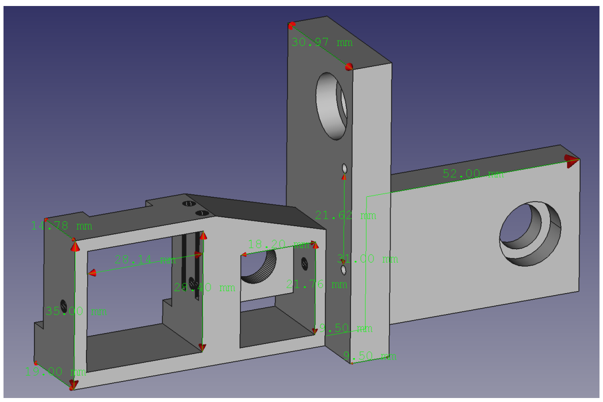


Fig. 6. X-adapter CAD 1

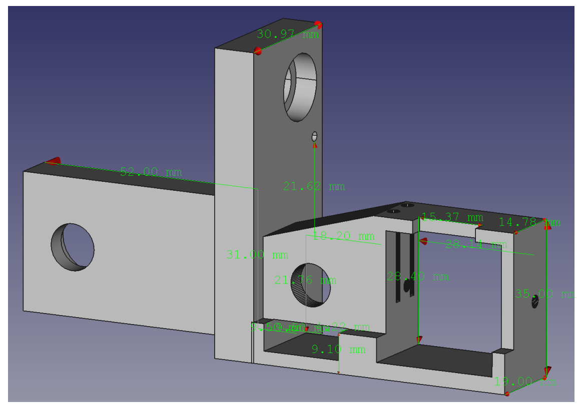


Fig. 7. X-adapter CAD 2

When assembled it looks like this:

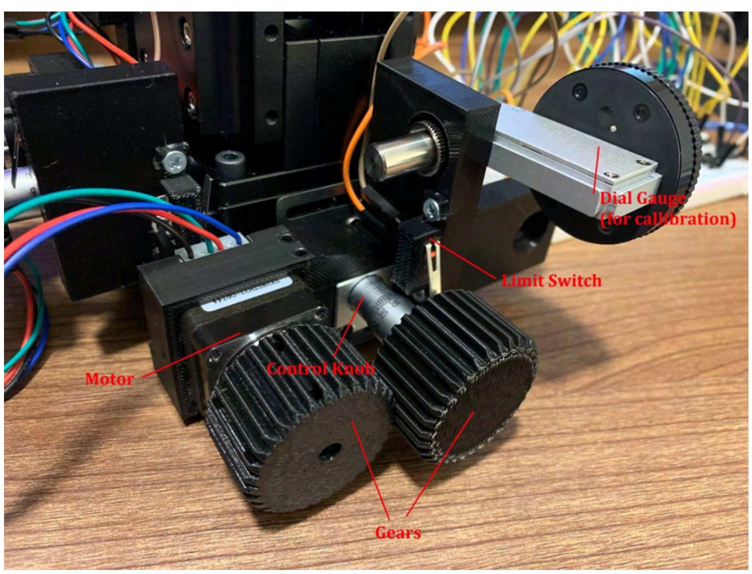


Fig. 8. X-adapter assembled

Y-adapter design: (same as X-adapter without dial gauge mounts)

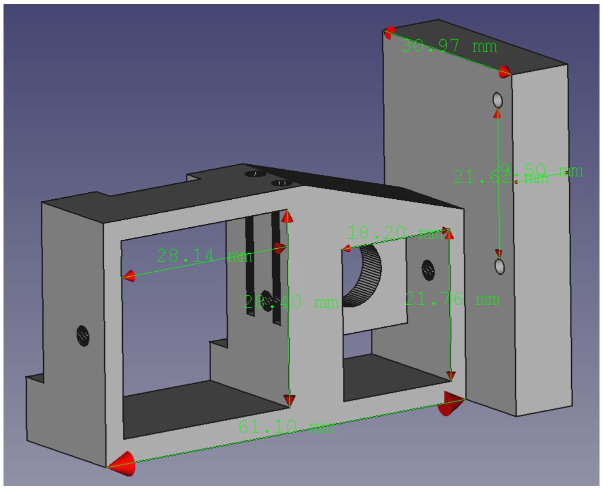


Fig. 9. Y-adapter CAD 1

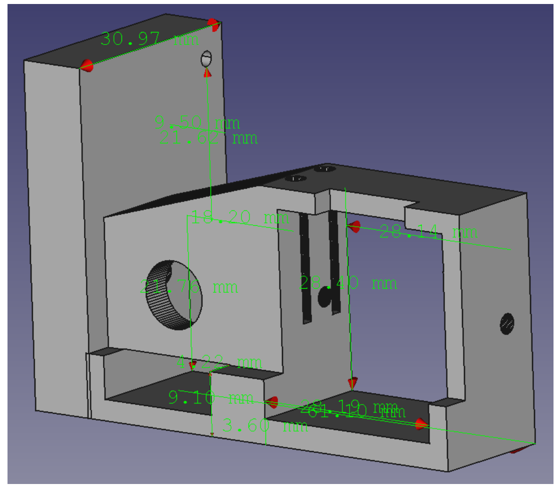


Fig. 10. Y-adapter CAD 2

When assembled it looks like this:

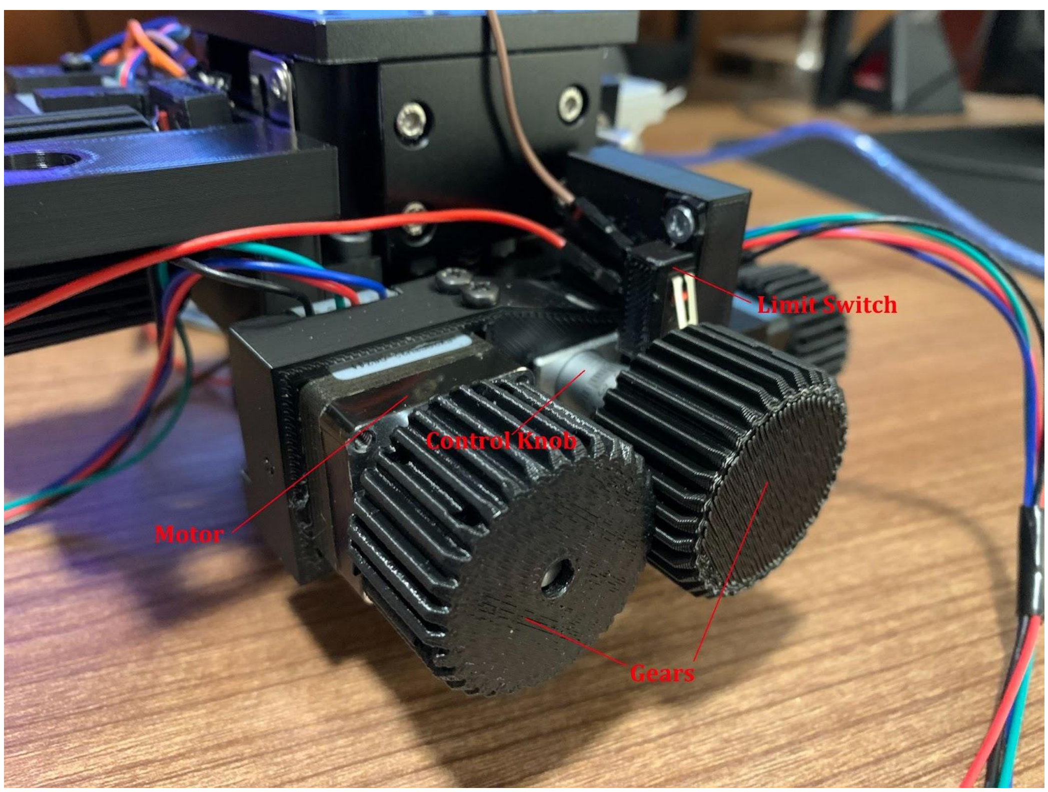


Fig. 11. Y-adapter assembled

Z-adapter design:

The open-source project did not include a Z-component for the stage. So, the Z-adapter required the design of an entirely new mounting mechanism.

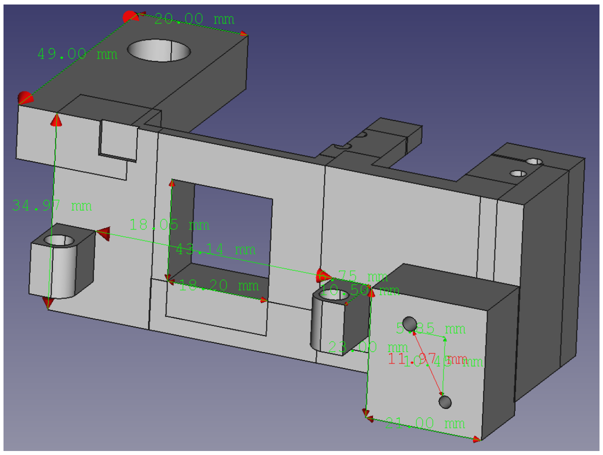


Fig. 12. Z-adapter CAD 1

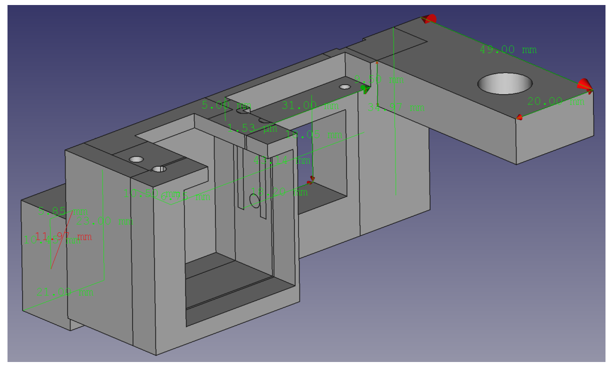


Fig. 13. Z-adapter CAD 2

When assembled it looks like this:

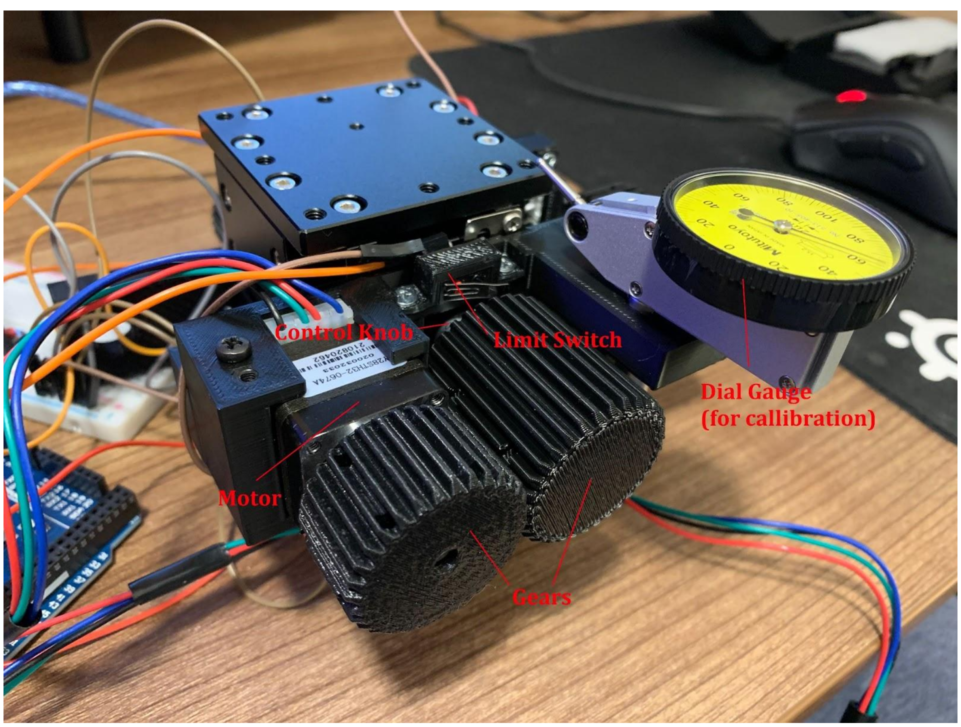


Fig. 14. Z-adapter assembled

Gears:

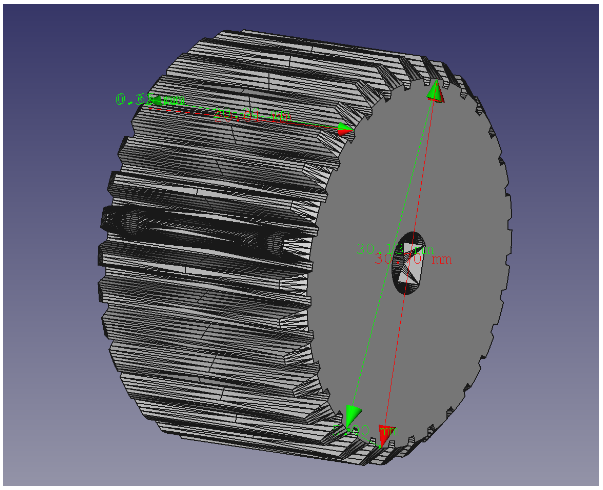


Fig. 15 Motor gear CAD

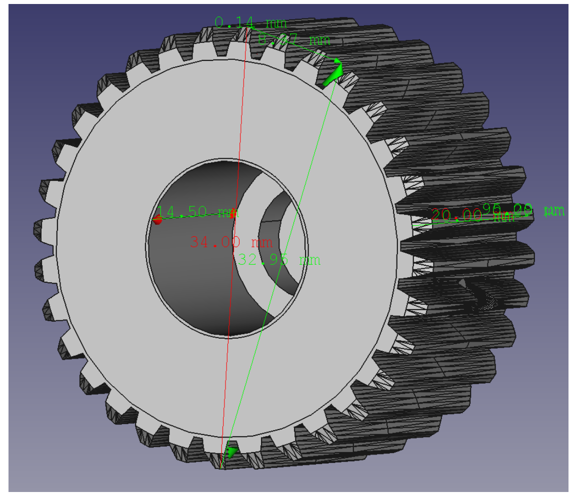


Fig. 16. XY knob gear CAD

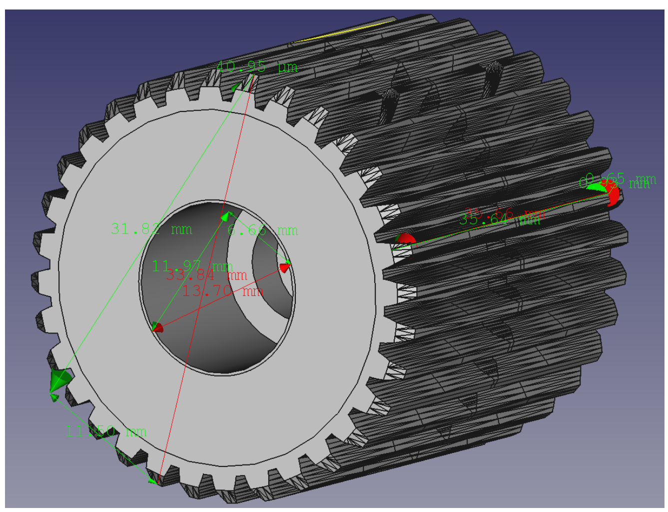


Fig. 17. Z knob gear CAD

4.1.2.3. Motors

In order to actually turn the gears and move the stage, the project partner provided NEMA 11-size hybrid bipolar stepper motors[4].

The manufacturer provides these mechanical drawings:

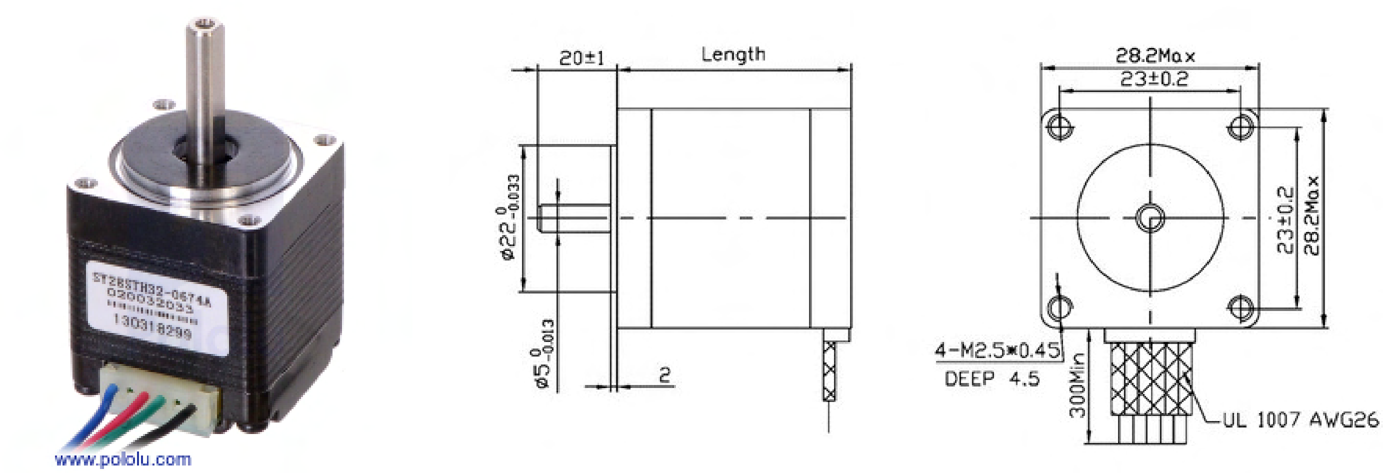


Fig. 18. Motor mechanical drawings

As well as these specifications:

Specifications

- Size: 28 mm square × 32 mm, not including the shaft (NEMA 11)
- Weight: 110 g (4 oz)
- Shaft diameter: 5 mm "D"
- Steps per revolution: 200
- Current rating: 670 mA per coil
- Voltage rating: 3.8 V
- Resistance: 5.6 Ω per coil
- Holding torque: 600 g-cm (8.3 oz-in)
- Inductance: 4.2 mH per coil
- Lead length: 30 cm (12")
- Output shaft supported by two ball bearings

Fig. 19. Motor specifications

Most notably, the motor is capable of 200 steps per revolution (without microstepping).

4.1.2.4. Control Hardware/Firmware

In order to precisely control the stepper motors, the project partner provided a Makerbase MKS DLC V2.1 control board[5]:

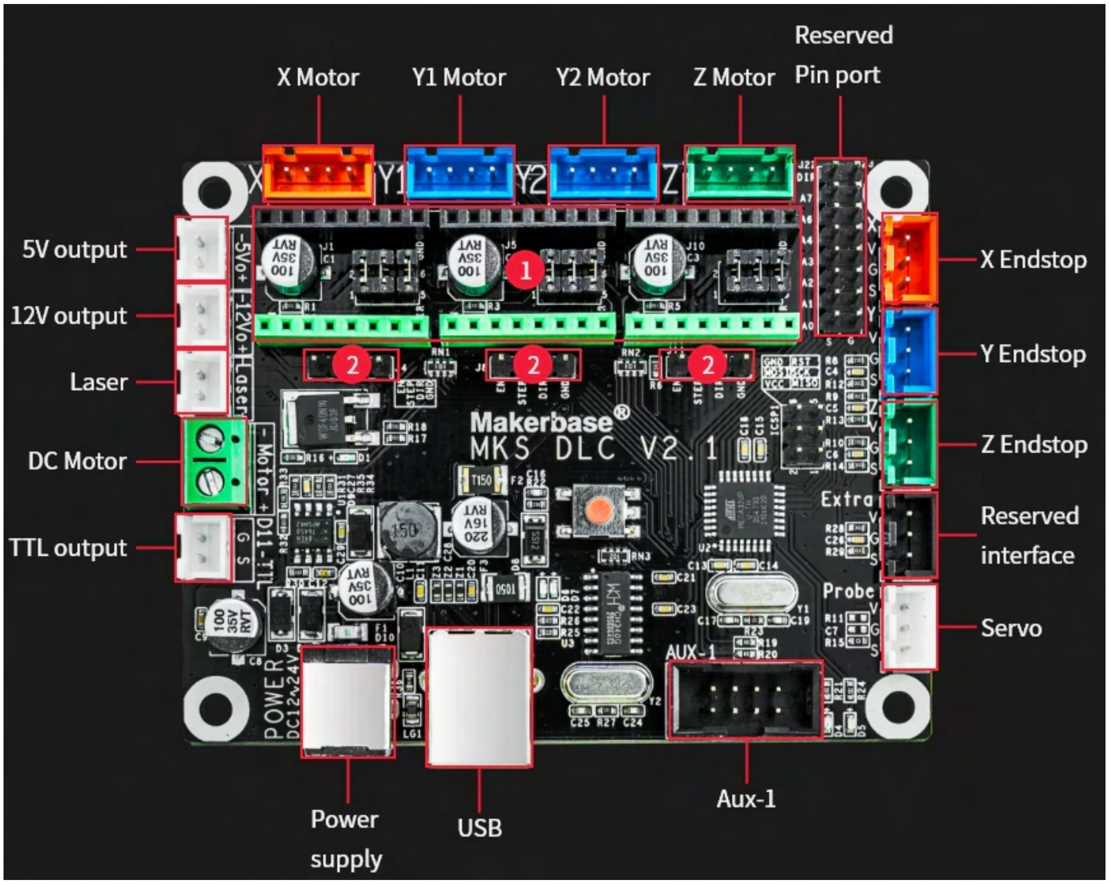


Fig. 20. Control hardware board

On this board, as suggested by the open-source reference project, we are running an open-source motor control firmware called Grbl[6]. The firmware can be calibrated with various settings (most importantly the steps/mm for each axis) and takes G-code commands through a USB interface. G-code commands are a specific format for motion control. For example: “G21 G91 G01 X0.1 F30” tells the stage to move linearly 0.1mm in the X direction with respect to its current location at a speed of 30mm/min. The serial interface functions at 115200 baud and uses the UART protocol.

4.1.3. General Validation

The basic goal of this block is to produce precise stage motion based on specific commands. Each part of the design allows for this to happen while meeting the overall project requirements.

The mechanical stage itself claims a precision of 10um with a total range of 13mm[1]. This makes the project requirements of 10um accuracy with a total range of at least 8mm possible. Additionally, 1 full rotation of each control knob corresponds to 500um of stage motion. The NEMA 11 motors are capable of 200 steps/revolution before microstepping[4]. The MKS DLS control board is capable of microstepping as much as 1/16 of a step[5]. So, by combining these hardware pieces, we can get 3200 steps/revolution of the motors. This leads to a ratio of 64 steps per 10um of stage motion, allowing for plenty of precision to achieve our desired 10um accuracy with a tolerance of +/- 10%.

The other hardware pieces allow for these goals to be met as well. The motor and control knob gears are the same diameter, leading to a 1:1 gear ratio which is all that is necessary due to the sufficient motor steps described above. The gears fit completely snuggly on the knobs, allowing for consistent, sturdy mounting. The use of PLA is plenty durable for this low torque application of the motors[7].

The control firmware, Grbl, is described as a “no-compromise, high performance, low cost alternative to parallel-port-based motion control for CNC milling“ that is ”written in highly optimized C utilizing every clever feature of the AVR-chips to achieve precise timing and asynchronous operation. It is able to maintain up to 30kHz of stable, jitter free control pulses.”[6] The wide use of this firmware for CNC milling leads to confidence that it is fully capable of delivering the precision necessary for the system. Additionally, it takes standard G-code commands, leading to a simple interface with the backend software of the system.

A dial gauge was chosen to calibrate the system due to its low-cost compared to alternative precise measurements devices like laser interferometers or linear scales.[8]

4.1.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

pwr_spply_mchncl_stg_nd_cntrl_hrdwrfrmwr_dcpwr : Input

Inominal: 1A	The NEMA 11 motors have a current rating of 670mA per coil[4]. Each motor has two coils. So, when a single motor is powered on it draws about 1A, and only a single motor will be used at a time.	Since the motors make up the vast majority of the current draw, the nominal current will be about 1A as calculated using the datasheet values[4].
Ipeak: 1.5A	The current draw of a single motor being powered[4] as well as the max current draw of the microcontroller on the control board were added[5], and a surplus of 500mA was added on top of that to ensure the peak is never reached.	When a single motor is powered it draws 1A[4] and the microcontroller on the control board will draw a max of 500mA[5], so the current should never go over a peak of 1.5A.
Vmax: 12.3V	The control board takes a 12V input, so engineering judgment was used to set a reasonable max voltage 300mV above that.[5]	The control board takes a 12V input, and the voltage regulator on the board is capable of handling over 12.3V.[5]
Vmin: 11.7V	The control board takes a 12V input, so engineering judgment was used to set a reasonable min voltage 300mV below that.[5]	The control board takes a 12V input, and the voltage regulator on the board is capable of handling below 11.7V.[5]

bcknd_sftwr_mchncl_stg_nd_cntrl_hrdwrfrmwr_data : Input

Datarate: 115200 baud	The Grbl firmware which runs on the MKS DLC control board functions at this data rate.[6]	The Grbl firmware which runs on the MKS DLC control board functions at this data rate.[6]
Messages: G-code commands (Ex: G21 G91 G01 X0.1 F30)	The Grbl firmware which runs on the MKS DLC control board takes this command format as input.[6]	The Grbl firmware which runs on the MKS DLC control board takes this command format as input.[6]
Protocol: UART	The Grbl firmware which runs on the MKS DLC control board utilizes this protocol.[6]	The Grbl firmware which runs on the MKS DLC control board utilizes this protocol.[6]

mchncl_stg_nd_cntrl_hrdwrfrmwr_otsd_envout : Output

Other: Max Stage Speed: 1mm/sec (in X, Y, and Z directions)	Engineering judgment was used to determine that this speed would be a reasonable max in order to allow for precise stage motion.	The Grbl firmware is capable of a lowest tick rate of about 30 steps/sec. This corresponds to about 5 um per second for the system. Thus the system is fully capable of operating at speeds slower than 1mm/sec and can do so by simply only specifying G-code commands with a feed rate at or below this max value.[6]
Other: Min Stage Range: 10mm (in X, Y, and Z directions)	The project partner stated this is an acceptable range for his applications of the system.	The stage itself claims a total range of 13mm in each direction.[1]
Other: Motion Resolution: 10um +/- 10% (in X, Y, and Z directions)	The project partner stated that this was the level of precision he thought would be acceptable.	1 full rotation of each stage control knob corresponds to 500um of stage motion. The NEMA 11 motors are capable of 200 steps/revolution before

		microstepping. The MKS DLS control board is capable of microstepping as much as 1/16 of a step. So, by combining these hardware pieces, we can get 3200 steps/revolution of the motors. This leads to a ratio of 64 steps per 10um of stage motion, allowing for plenty of precision to achieve our desired 10um accuracy with a tolerance of +/-10%.[4]

4.1.5. Verification Process

4.1.5.1. Input data verification

The input data interface properties specify a data rate, protocol, and message format. To confirm that these interfaces are met, the serial monitor in the Arduino IDE can be used.

1) Connect the control board to a computer via USB.
2) Select the correct COM port.
3) Open the serial monitor and set the baud rate to 115200.
4) Send the following command: "G21 G91 G01 X1 F30".
5) Ensure the stage moves 1mm in the X direction.

Since the Arduino serial monitor makes use of UART, the "Protocol: UART" property is verified. The baud rate is specifically set in the serial monitor, so the "Datarate: 115200 baud" property is verified. And if the stage moves corresponding to the command, the "Messages: G-code commands (Ex: G21 G91 G01 X0.1 F30)" property is verified.

4.1.5.2. Input power verification

The input power interface properties specify a nominal and peak current as well as a voltage min and max. To verify these interface properties:

1) Connect the board to a power supply set to 11.7V.
2) Turn on all 3 motors at the same time and ensure that the stage can move in all directions while the current is roughly 4A and never goes over 5A.

3) Set the power supply to 12.3V.
4) Turn on all 3 motors at the same time and ensure that the stage can move in all directions while the current is roughly 4A and never goes over 5A.

4.1.5.3. Output verification

To verify the "Other: Min Stage Range: 8mm" property:

1) Move the stage in the positive X direction completely to one limit switch.
2) Measure the stage position.
3) Move the stage in the other direction completely to the other limit switch.
4) Measure the stage position and ensure it was able to move at least 8mm.
5) Repeat for the Y and Z directions.

To verify the "Other: Motion Resolution: 10um +/- 10%" property:

1) Measure stage position with a dial gauge.
2) Use the GUI to instruct the stage to move 10um in the X direction.
3) Measure the new position and ensure the stage really moved 10um +/- 10%.
4) Repeat for the Y and Z directions.

To verify the "Other: Max Stage Speed: 1mm/sec" property:

1) Measure stage position.
2) Use the GUI to instruct the stage to move 10mm in the X direction.
3) Time how long it takes for the stage to stop moving and ensure this time was 10 seconds or less.

4.1.6. References and File Links

4.1.6.1. References (IEEE)

[1] AliExpress. (2023, Jan. 19). *XY Axis 60*60mm Trimming Station Manual Displacement Platform Linear Stage Sliding Table XY60-C XY60-R,XY60-LM LY60 Cross Rail* [Online]. Available: https://www.aliexpress.us/item/2255800615306628.html?spm=a2g0o.productlist.0.0.303b301dvo4AzM&algo_pvid=ca5e3889-75d2-4a53-97d2-df5810ccfbcc&algo_expid=ca5e3889-75d2-4a53-97d2-df5810ccfbcc-2&btsid=2100bb47161190122535265554e7664&ws_ab_test=searchweb0_0%2Csearchweb201602_%2Csearchweb201603&gatewayAdapt=deu2usa4itemAdapt&_randl_shipto=US

[2] AliExpress. (2023, Jan. 19). *Z axis 60*60mm Displacement Lift Stage Manual fine tuning platform Double Cross rail Sliding Table LZ60-2 Z60-2* [Online]. Available: https://www.aliexpress.us/item/2251832663789031.html?spm=a2g0o.detail.1000014.29.3a3c40ectnU3gf&gps-id=pcDetailBottomMoreOtherSeller&scm=1007.40050.281175.0&scm_id=1007.40050.281175.0&scm-url=1007.40050.281175.0&pvid=914bbda3-83a9-48f2-abe6-e3d96a6563dc&_t=gps-id:pcDetailBottomMoreOtherSeller,scm-url:1007.40050.281175.0,pvid:914bbda3-83a9-48f2-abe6-e3d96a6563dc,tpp_buckets:668%232846%238114%231999&pdp_ext_f=%7B%22sku_id%22%3A%2265215981414%22%2C%22sceneId%22%3A%2230050%22%7D&pdp_npi=2%40dis%21USD%2152.22%2146.48%21%21%21%21%21%402101d1ad16742802434645347e72ab%2165215981414%21rec

[3] Github. (2023, Jan. 19). *openUC2 MicronStage* [Online]. Available: https://github.com/openUC2/UC2-MicronStage

[4] Pololu. (2023, Jan. 19). *Stepper Motor: Bipolar, 200 Steps/Rev, 28×32mm, 3.8V, 0.67 A/Phase* [Online]. Available: https://www.pololu.com/product-info-merged/1205

[5] Github. (2023, Jan. 19). *MKS DLC* [Online]. Available: https://github.com/makerbase-mks/MKS-DLC?spm=a2g0o.detail.1000023.17.6fac69a0FVHUp9

[6] Github. (2023, Jan. 19). *Grbl* [Online]. Available: https://github.com/gnea/grbl

[7] 3DPrinterly. (2023, Jan. 19) *How Long Do PLA 3D Printed Parts Last?* [Online]. Available: https://3dprinterly.com/how-long-do-3d-printed-parts-last/

[8] Mitutoyo. (2023, Jan. 19). *Dial Test Indicator, Horizontal Type 0,2mm, 0,002mm, 8mm Stem* [Online]. Available: https://shop.mitutoyo.eu/web/mitutoyo/en/all/all/Dial%20Test%20Indicator,%20Horizontal%20Type/PR/513-405-10E/index.xhtml

4.1.7. Revision Table

1/19/2023	Logan York: Initial document creation.
2/11/2023	Logan York: Revised introduction section, reformatted design section, added references to validation section, added DC power interface and references to validation, reformatted references.

4.2. Limit Switches

4.2.1. Description

Switches that sense when the stage has moved to its limit in any direction in order to stop the motion. There are 6 switches in total, two for each axis. The switches will produce a digital signal which will be read in by an arduino nano on the input controls PCB and outputted to the GUI in order to limit the stage motion.

4.2.2. Design

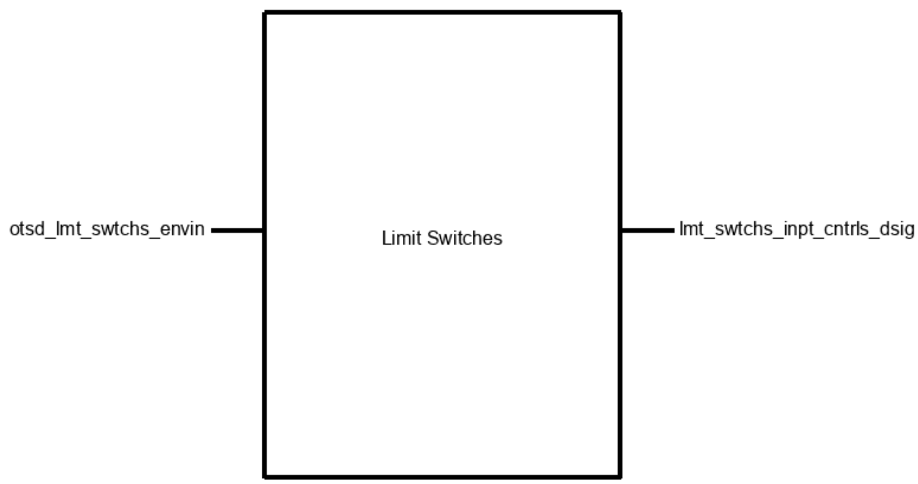


Fig. 1. Black box diagram

The limit switches used for the system are called “Cylewet AC 1A 125V 3Pin SPDT Limit Micro Switch Long Hinge Lever CYT1073”. The manufacturer provides these images and mechanical drawings:

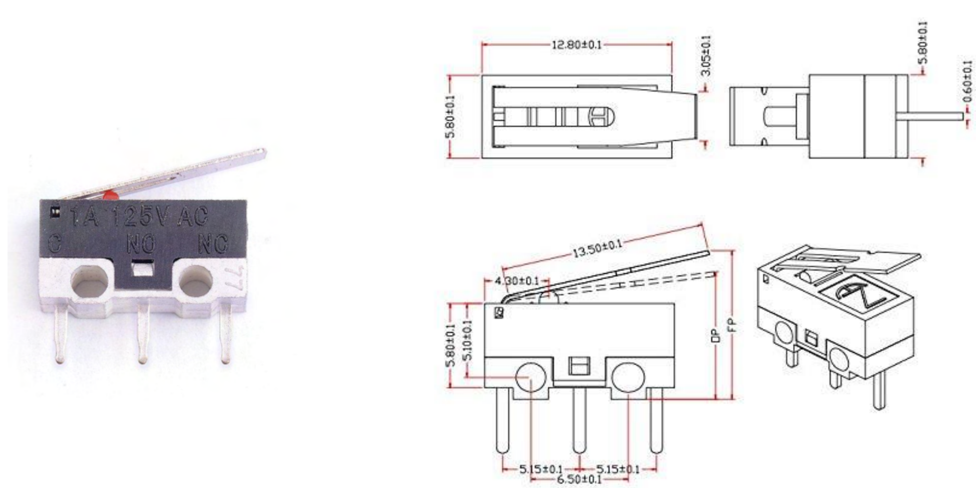


Fig. 2. Limit switches

In order to fix the switches to the stage, screw holes are included in the design of each adapter. These adapters were detailed in the previous section. Additionally, 3D printed adapters for each individual limit switch were designed to assist in fixing the switches to the stage.

Limit Switch Adapters Design:

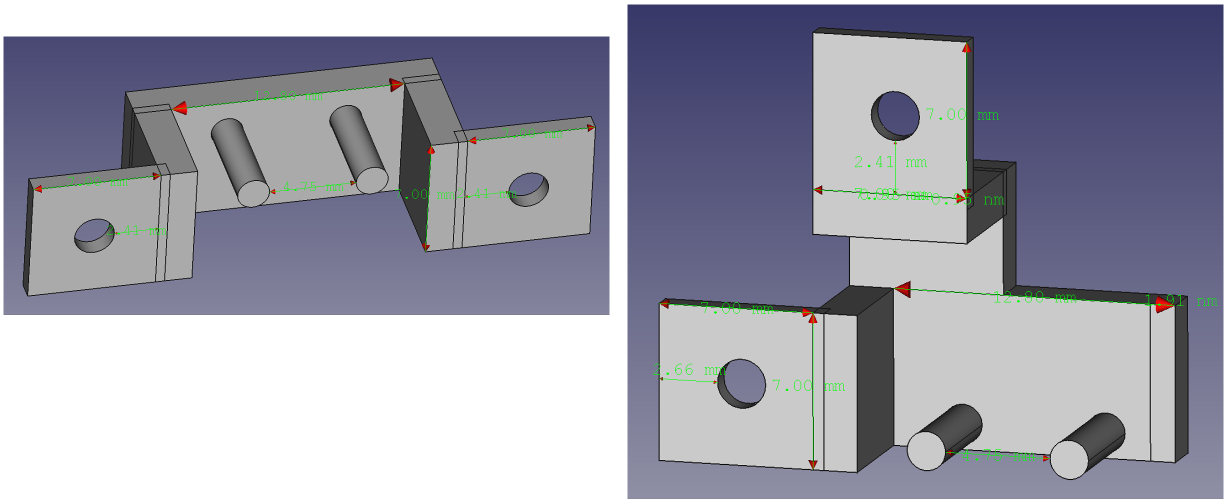


Fig. 3. Limit switch adapters design

When fixed to the stage, the limit switches and adapters look like this:

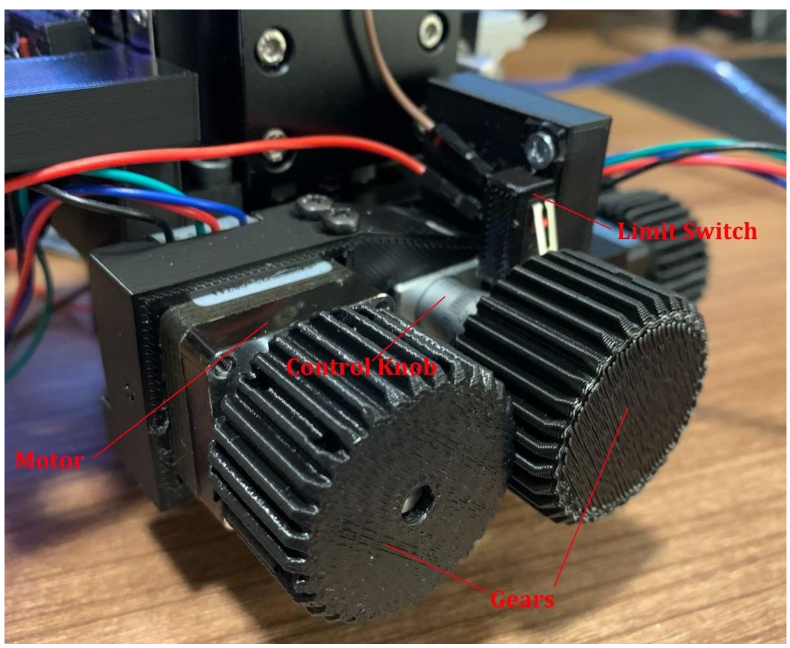


Fig. 4. Limit switch assembled

4.2.3. General Validation

The goal of this block is to allow the system to know when the stage has moved to its limit in any direction. This design accomplishes that goal by having a designated limit switch for each limit of every axis. The limit switches[1] themselves are rated for 1A and 125V which is plenty for our system as they interface with an Arduino Nano, which operates at 5V. Additionally, these limit switches are often used with 3D printers, which is very similar to our application, meaning they will certainly hold up for this specific use case. Fixing them to the stage using the same 3D printed adapters for the motors/gears allows for the limit switches to be seamlessly incorporated into the system.

4.2.4. Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

otsd_lmt_swtchs_envin : Input

Other: Max Stage Increments: 0.1mm	The system needs to read in the limit switch values after moving the stage in order to know whether a limit has been reached. In order for the stage to not move past its physical limits, it must move in small increments. Thus, engineering judgment was used to make the max stage increments 0.1mm.	The limit switches will be fixed no closer than 0.1mm from the physical limits of the stage, therefore in the worst case scenario, the stage will not move past the physical limits.
Other: Max Limit Margin: 1mm (limit switches will activate within 1mm of the physical stage limits)	In order to not waste too much of the stage's range by putting the limit switches too far from the physical limits, engineering judgment was used to set the max limit margin to 1 mm.	The limit switches were fixed and adjusted so as to not waste unnecessary amounts of stage range, positioning them within 1mm of the physical stage limits.
Other: Max Stage Speed: 1mm/sec	Engineering judgment was used to determine that this speed would be a reasonable max in	The Grbl firmware is capable of a lowest tick rate of about 30 steps/sec. This corresponds to about 5 um per second for the

	order to allow for precise stage motion.	system. Thus the system is fully capable of operating at speeds slower than 1 mm/sec and can do so by simply only specifying G-code commands with a feed rate at or below this max value.[6]

lmt_swtchs_inpt_cntrls_dsig : Output

Logic-Level: Active Low	The limit switches are not pressed until the stage reaches its limit. Therefore, the normally open pin of the switches is used. Using the internal pull-up resistors of the Arduino Nano makes the switches active low.	The limit switches are not pressed until the stage reaches its limit. Therefore, the normally open pin of the switches is used. Using the internal pull-up resistors of the Arduino Nano makes the switches active low.
Vmax: 5V	The Arduino Nano functions at 5V, so when unpressed the voltage reading on the input pin will be 5V.	The Arduino Nano functions at 5V, so when unpressed the voltage reading on the input pin will be 5V.
Vmin: 0V	The input pin on the Arduino Nano will be tied to ground when the switch is pressed, resulting in a voltage reading of 0V.	The input pin on the Arduino Nano will be tied to ground when the switch is pressed, resulting in a voltage reading of 0V.

4.2.5. Verification Plan

Output Verification

1) Connect the switches to the Arduino Nano. 1 pin to the input and 1 pin to ground.
2) With the switch unpressed, measure the voltage on the input pin and ensure it is 5V.
3) Press the limit switch, measure the voltage on the input pin and ensure it is 0V. This means the switch is active low.

Input Verification

To verify the “Other: Max Stage Speed: 1mm/sec” property:

1) Measure stage position.
2) Use the GUI to instruct the stage to move 5mm in the X direction.
3) Time how long it takes for the stage to stop moving and ensure this time was 5 seconds or less, meaning the stage is capable of moving at 1mm/sec.

To verify the “Other: Max Stage Increments: 0.1mm” property:

1) Measure stage position with a dial gauge.
2) Use the GUI to instruct the stage to move 0.1mm in the X direction.
3) Measure the new position and ensure the stage really moved 0.1mm.
4) Repeat for the Y and Z directions.

To verify the “Other: Max Limit Margin: 1mm (limit switches will activate within 1mm of the physical stage limits)” property:

1) Move the stage in the positive X direction completely until the limit switch activates.
2) Manually twist the control knob until the stage reaches its physical limit, and ensure it takes less than 2 full rotations to do so. (2 full rotations of the knob correspond to 1mm of stage motion, so if it takes less than 2 full rotations to reach the physical limit, then the switch stopped the stage within 1mm of this limit.)
3) Repeat for the negative X direction as well as the Y and Z directions.

4.2.6. References and File Links

[1] Amazon. (2023, March 12). *Cylewet 25Pcs AC 1A 125V 3Pin SPDT Limit Micro Switch Long Hinge Lever for Arduino (Pack of 25) CYT1073* [Online]. Available: https://www.amazon.com/dp/B073TYWX86?psc=1&ref=ppx_yo2ov_dt_b_product_details

4.2.7. Revision Table

3/12/2023	Logan York: Initial section creation.

4.3. Input Controls

4.3.1. Description

The project consists of a stage that moves precisely in three axes in order to view the details of an object on the stage with a microscope. The X and Y axes will be able to move with 1 μm precision and the Z axis will move with 0.1 μm precision. Additionally, a requirement of the project will be to attempt to keep the cost low at a price of three hundred dollars or less. The complete project's purpose is to provide alternative equipment for researchers to view and study the layout and architecture of microchips and other electronics. Pictures and live video will be taken of the subject matter on the stage and stitched together for detailed viewing. Movement of the stage allows for focused pictures along with rapid adjustments for following traces or observing other parts of the microchip.

The Input Controls block will serve as a method to communicate with the stage through physical controls. Limit switches are also processed by this block to streamline components needed to control the stage. The input controls will be an array of three endless encoders (rotary encoders) to control the X, Y, and Z motion of the stage. Movement of each encoder will be read from an Arduino Nano and outputted to the GUI for processing. Indicator LEDs on the block will be able to briefly display settings and relevant parameters. Integrated button presses from the encoders allow for these display LED parameters to be toggled on or off.

The block will take in several parameters from each of the interfaces. The primary interface that is taken into the block is the outside user inputs, listed as 'otsd_inpt_cntrls_usrin' on the black box diagram below. This is done by the rotary encoders through their rotation and built-in button presses. The encoders chosen consist of two offset rings that send varying inputs based on the direction of rotation. Each click of the knob changes the signal sent to the Arduino where it is processed to determine movement. From here, the Arduino will output to the serial monitor to be polled by the GUI via a USB connection. Additionally, button presses from the encoders will output in a similar manner to the GUI with an additional internal block output of LEDs. The secondary interface will consist of the limit switch digital signal input. The active low limit switches will send either a zero or five volt signal to six of the pins on the Arduino. This allows for the minimum and maximum stage distances to be tracked and avoids damage to the mechanical components.

4.3.2. Design

The three main input interfaces for the block come from the power supply, the limit switches block, and outside input from the user. An output interface connects the block to the graphical user interface (GUI) where data and limit values are to be displayed.

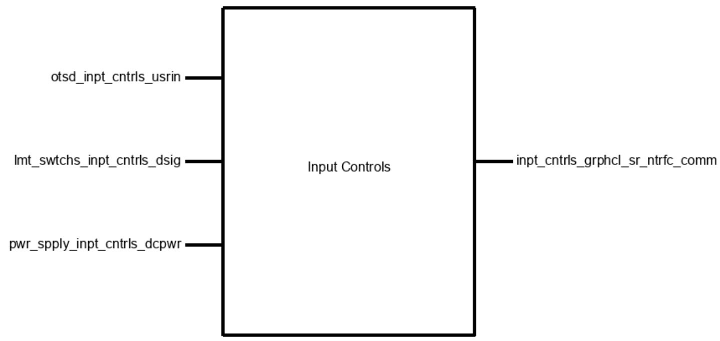


Fig. Input Controls Black Box Diagram

The schematic below depicts the full schematic for the encoders, LEDs, and connection for limit switches to the Arduino Nano model. Note that two RGB LEDs were chosen for the relative first and second encoders as the parameters they represent will have multiple states (i.e. disabled, idle, connected, etc.).

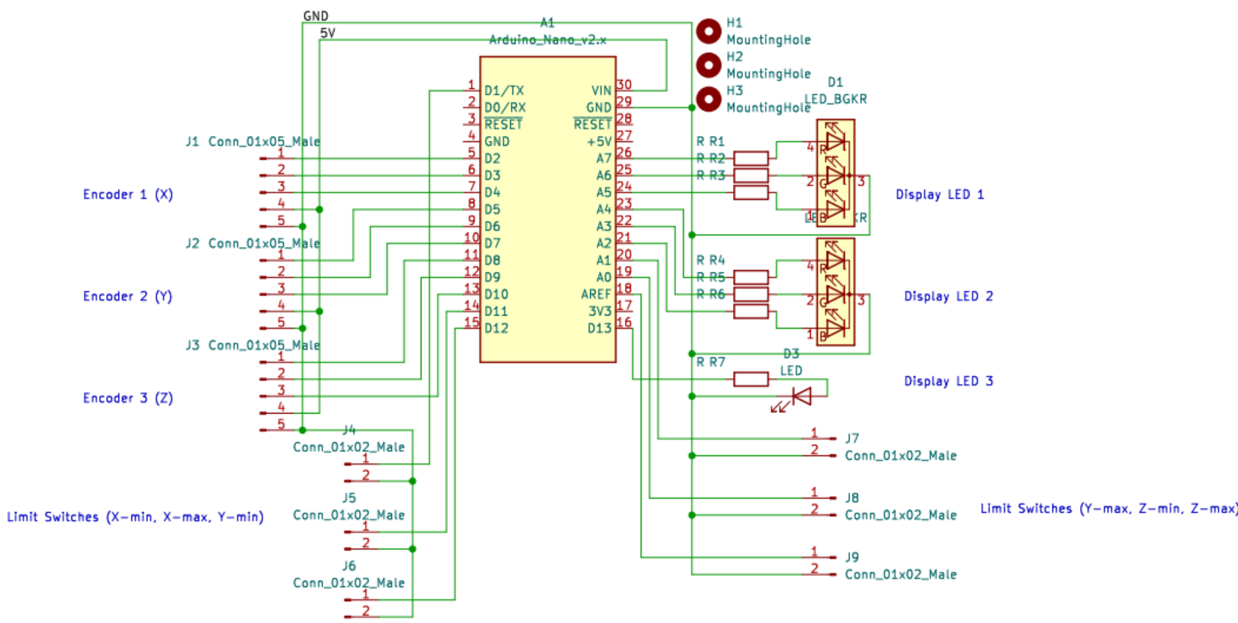


Fig. Input Controls Schematic

The PCB footprints chosen will allow for resistors, LEDs, five and two pin connectors, and the Arduino Nano to be connected directly to the board. Minimal components are needed to accommodate the encoders as direct digital signals can be taken into the Arduino Nano with the 5V supply connected to the board's voltage pin. A generated view of the PCB is also shown below to illustrate what the final product may look like.

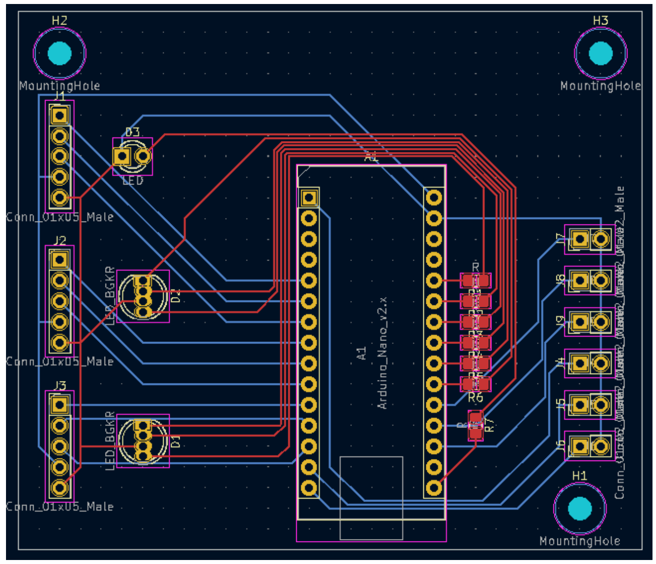


Fig. Input Controls PCB Layout

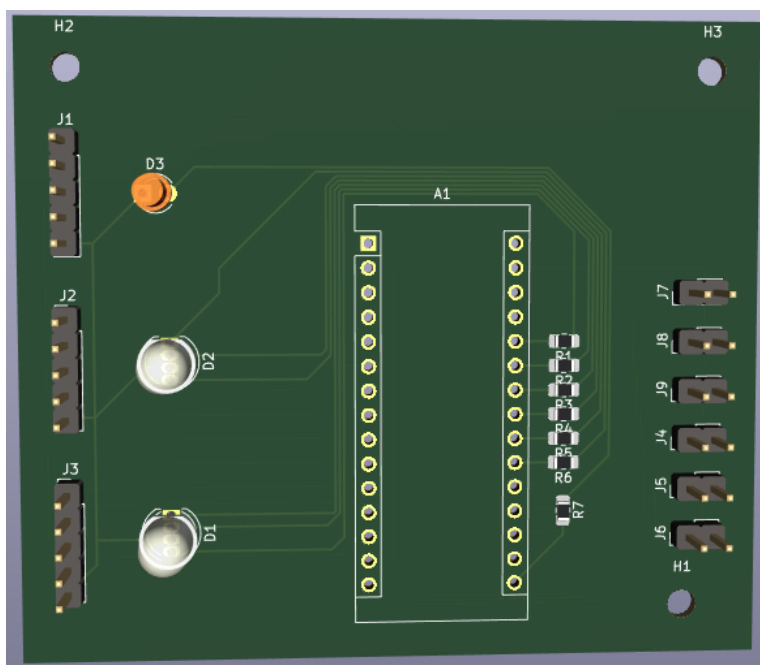


Fig. Generated View of Input Controls PCB

A short excerpt of the code to be used in the design is listed below and demonstrates the method of comparing the clock and dt values to determine the direction of rotation. The button press of the encoder is also shown with a minimum depress of 50 ms. This ideally will prevent switch bouncing. This section of code is repeated for the other two encoders to gain a full picture of all inputs.

```
void loop() {
  unsigned long currentTime = millis();
  //**********************************************
  // Read the current state of CLK
  currentStateCLK1 = digitalRead(CLK1);

  // If last and current state of CLK are different, then pulse occurred
  // React to only 1 state change to avoid double count
  if (currentStateCLK1 != lastStateCLK1  && currentStateCLK1 == 1) {

    // If the DT state is different than the CLK state then
    // the encoder is rotating CCW so decrement
    if (digitalRead(DT1) != currentStateCLK1) {
      counter1 ++;
      //currentDir1 = "CW";
    } else {
      // Encoder is rotating CW so increment
      counter1 --;
      //currentDir1 = "CCW";
    }
  }

  // Remember last CLK state
  lastStateCLK1 = currentStateCLK1;

  // Read the button state
  int btnState1 = digitalRead(SW1);

  //If we detect LOW signal, button is pressed
  if (btnState1 == LOW) {
    //if 50ms have passed since last LOW pulse, it means that the
    //button has been pressed, released and pressed again
    if (currentTime - lastButtonPress1 > 50) {
      Serial.println("B1");
    }

    // Remember last button press event
    lastButtonPress1 = currentTime;
  }
  delay(5);
```

Fig. Section of Arduino Code

4.3.3. General Validation

The control methods of the project are set as being able to have physical knobs, arrow keys on the computer, and G-code inputs to move the XYZ-stage. The physical controls are able to provide the best precision through wired communication to the GUI. This eliminates the need to set up the block for wireless communication or parse analog signals. Additionally, the outside input from the user is intuitively handled through a physical control unit as opposed to software. Furthermore, the limit switches that connect to the block were chosen to be a digital signal in order to be directly read by the block to disable a knob for its respective axis. The rotary encoders provide continuous movement as opposed to a potentiometer and will better suit the nature of axis control for the project. The serial communication is planned to be slower at 9600 Hz in order to avoid data loss or misinterpretation. Lastly, active-low limit switches were chosen to reduce switch bouncing.

4.3.4. Interface Validation

otsd_inpt_cntrls_usrin : Input

Timing: 50 ms poll rate	50 ms is enough time for the button press to be detected without having phantom inputs.	Hard coded parameter in .ino file for the button polling.
Type: Physical Rotation	Physical rotation as opposed to a joystick is prone to less drift	Encoder knobs have been purchased and rotate as

	and clearer input values.	expected. Measurement of rotation will be observed by varying values in the serial monitor.
Usability: Consistent readings more than 95% of the time	Considerable precision as opposed to absolute precision is relevant for the size of steps that the stage will move.	Tactile rotation positions can be felt on knobs.

inpt_cntrls_grphcl_sr_ntrfc_comm : Output

Datarate: 9600 Hz	The baudrate for serial communication is fast, but reliable.	Hard coded into .ino file. GUI will also initialize connection with the port via the serial monitor. Valid data on the receiving end is verified by having 9600 Hz as a hard coded baudrate.
Other: Serial Comm	Serial communication is well documented as opposed to other methods.	Hard coded into .ino file. Connection is established by choosing a COM port listed on the computer.
Protocol: UART	The Arduino Nano is set to UART.	Hard coded into .ino file. Not relevant for the method of polling used in the GUI.

lmt_swtchs_inpt_cntrls_dsig : Input

Logic-Level: Active Low	Fewer errors with an active low signal.	Will be determined by Limit Switches block. Different button presses will simulate connection to Input Controls block.
Vmax: 5V	5V for a high signal to be processed by the Arduino Nano.	Will be determined by Limit Switches block. Different button press voltages will simulate connection to Input Controls block.
Vmin: 3.3V	Minimum threshold for a digital signal to be detected.	Will be determined by Limit Switches block. Different button press voltages will simulate connection to Input Controls

		block.
Vnominal: 4.5V	Optimal signal to be detected by nano.	Will be determined by Limit Switches block. Different button press voltages will simulate connection to Input Controls block.

pwr_spply_inpt_cntrls_dcpwr : Input

Inominal: 10mA	Maximum current tolerated to run the Arduino according to the datasheet [2].	Will be determined by the Power Supply block. The current will aim to supply the Arduino Nano, yet not exceed its power limit.
Ipeak: 100mA	Current needed to run the Arduino according to the datasheet [2].	Will be determined by the Power Supply block. The current will aim to supply the Arduino Nano, yet not exceed its power limit.
Vmax: 5.25V	Voltage needed to run the Arduino according to the datasheet [2].	Will be determined by the Power Supply block. Voltage must not exceed the tolerances of the Arduino Nano.
Vmin: 4.35V	Voltage needed to run the Arduino according to the datasheet [2].	Will be determined by the Power Supply block. Voltage must not exceed the tolerances of the Arduino Nano.

4.3.5. Verification Plan

The below verification plan aims to verify functionality of the block through establishing valid connections through each of its interfaces. Partial functionality of the limit switches and GUI are needed for verification. An external power supply is provided and assumed to be within spec to avoid damage to the block.

4.3.5.1. **Voltage and GND**

4.3.5.1.1. Ensure Arduino Nano GND pin connection is continuous with all ground connections of elements.

4.3.5.1.2. Connect to the provided power supply via the Arduino Nano's mini-B USB connector.

4.3.5.1.3. Ensure voltage is consistent (Vmax, Vmin, Vnominal) with power supply via 5V and GND pins on the Arduino Nano by using a multimeter or oscilloscope.

4.3.5.2. Arduino Code and Serial Monitor

4.3.5.2.1. Once the Arduino is connected, verify that all LEDs are off and limit switches are in the off position.

4.3.5.2.2. Reset the microcontroller with the white button in the center of the nano and upload the Serial_Const.ino file through the Arduino IDE.

4.3.5.2.3. Open the serial monitor in the Arduino IDE for testing.

4.3.5.3. Encoders and Button presses

4.3.5.3.1. Verify inputs in for all encoders and buttons using the table provided.

4.3.5.3.2. Note: The ‘#’ character for verifying encoder rotation represents a decimal value that starts at zero. The value shall increase with clockwise rotation and decrease with counterclockwise rotation.

Table II
Encoder References

Encoder 1 (X axis)		Encoder 2 (Y axis)		Encoder 3 (Z axis)	
Button Press	Encoder Rotation	Button Press	Encoder Rotation	Button Press	Encoder Rotation
‘B1’	1 #	‘B2’	2 #	‘B3’	2 #

4.3.5.4. Test button timing through a range of time pressing buttons, i.e. as fast as humanly possible to a lower frequency of 3 seconds.

4.3.5.4.1. Turn individual encoders at minimum rotation and repeated full rotations and monitor outputs accordingly. Verify results in the serial monitor.

4.3.5.4.2. Note: encoders should rotate at a reasonable rate to emulate fine movement by an individual.

4.3.5.4.3. Turn multiple encoders simultaneously to ensure multiple operations can be performed. Verify results in the serial monitor.

4.3.5.5. Wired Communication with the GUI

4.3.5.5.1. Connect the Input Control block to a computer via USB if not done already.

4.3.5.5.2. Run the ‘BestGUI.py’ file to open the GUI window.

4.3.5.5.3. Establish a connection between the Input Controls block and the GUI by selecting the COM port that the Arduino is connected to.

4.3.5.5.4. Press initialize and view the string displayed in the corner of the GUI. This string represents the data sent via the serial port from the Arduino Nano.

4.3.5.5.5. Repeat steps 3.1 through 3.4 to verify functionality.

4.3.5.6. Limit Switches

4.3.5.6.1. With established connections of the serial monitor and GUI running, check that the connected limit switches are functional.

4.3.5.6.2. In the event of no connected limit switches, replicate using active-low buttons.

4.3.5.6.3. Start by pressing the 'X-minimum' limit switch or respective button and check the GUI for an alert for the respective axis.

4.3.5.6.4. Verify rotation is disabled for the respective encoder by twisting the knob beyond the limit and checking that the value for that axis does not change.

4.3.5.6.5. Turn the encoder in the opposite direction to return to a valid range.

4.3.5.6.6. i.e. increase the value for testing the min and decrease for testing the max.

4.3.5.6.7. Repeat steps 5.2 to 5.4 for the X-maximum, Y-minimum, Y-maximum, Z-minimum, and Z-maximum switches.

4.3.5.6.8. Test the limit switches on a second axis with one limit switch already active.

4.3.5.6.9. Test a third limit switch using the same method above with two already active.

4.3.5.6.10. Note: This aims to test the behavior in all 6 corners of the prism of movement.

4.3.6. References and File Links

[1] B. Diederich, "OpenUC2/UC2-micronstage," *GitHub*, 24-Jun-2021. [Online]. Available: https://github.com/openUC2/UC2-MicronStage. [Accessed: 20-Jan-2023].

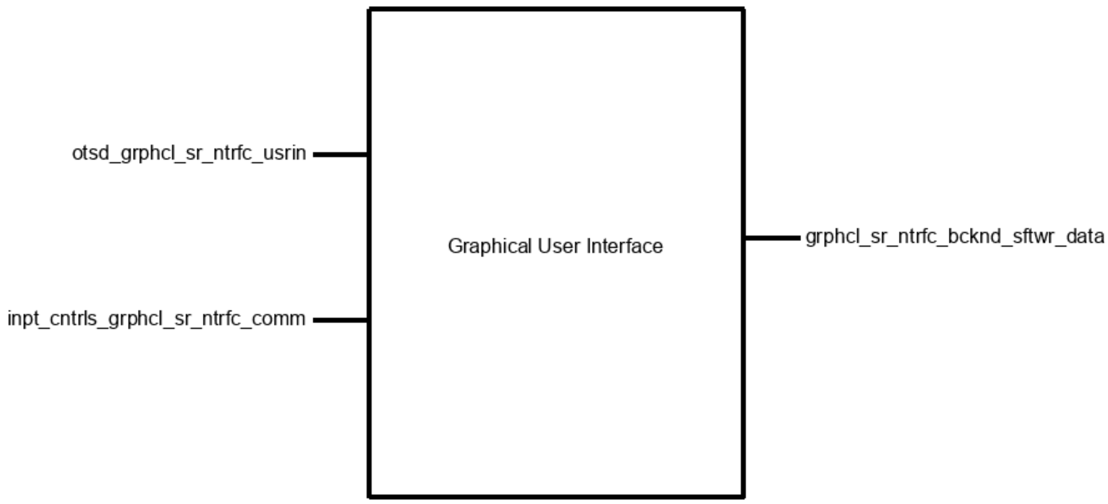
[2] "Arduino Nano2 3." [Online]. Available: https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf. [Accessed: 20-Jan-2023].

4.3.7. Revision Table

1/20/2023	Aleksi Hieta: Filled in info for sections 2, 4, 5, and 6 from the generated template.
2/11/2023	Aleksi Hieta: Revisions to sections 1, 2, 3, 4, and 5.

4.4. Graphical User Interface

4.4.1. Description



The Graphical user interface allows for the given user to communicate with the stage. This is an interface running in python 3.0 and utilizes tkinter to create a tab system to access initialization, control, and video settings independently. Threaded comm port communication allows for multiple serial ports to be active for receiving and sending data.

4.4.2. Design

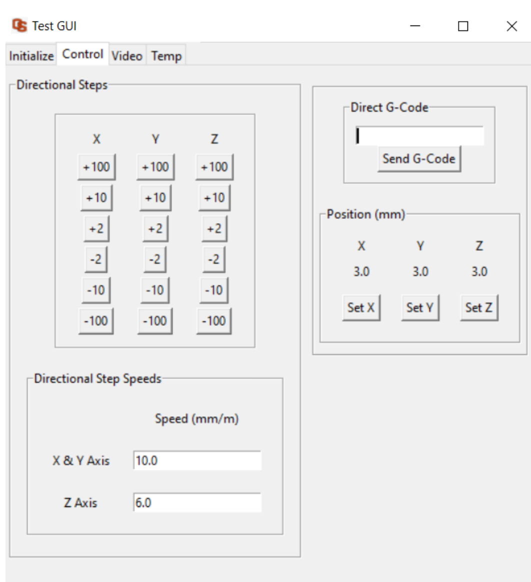


Fig. GUI Control Tab

The design is a three tab system that has a fourth temporary tab for error message checking. This can be added or withdrawn from the GUI depending on the user's preferences. The main portion of the GUI is the control tab, which runs the step commands and direct G-code options for moving the stage. The step commands consistently allow the stage to move either, 2, 10, or 100 mm steps at a variable speed. Additionally, the direct G-code allows for there to be complete control of the stage given a certain command.

4.4.3. General Validation

The control methods of the project are set as being able to have physical knobs, fixed step buttons, and G-code inputs to move the XYZ-stage. The physical controls are able to provide the best precision through wired communication to the GUI. This eliminates the need to set up the block for wireless communication or parse analog signals. However, the outside input from the user is intuitively handled through the input controls to directly receive limit switches in the GUI software. Furthermore, the GUI notifies the user of a limit being hit and restricts access to direction beyond an axis' limit. Additionally, the centering function on the GUI brings the stage to a central position that is far enough away from any of the edges.

4.4.4. Interface Validation

otsd_grphcl_sr_ntrfc_usrin

Other	Threaded COM port communication	Show that multiple serial port objects can be initialized and controlled
Type	String variable and Buttons	Methods of control are proven through copying valid strings and verifying buttons with measurement
Usability	Usable by 90% of students without difficulty	A survey will be performed with students testing and operating the stage with the GUI and asking their feedback and overall opinion at the end.

inpt_cntrls_grphcl_sr_ntrfc_comm

Datarate	9600	A baud of 9600 can be tested with a test aid arduino to show proper communication.
Messages	encoded 'utf'	UTF is the format used for all encoded serial messages.
Other	Serial Comm	The serial ports will be the primary form of

		communication between the GUI and all physical components.

grphcl_sr_ntrfc_bcknd_sftwr_data

Messages	Values	Different values for the speed in X, Y, and Z axes can be handled.
Other	Error Handling	Errors in direct G-Code or speed will be handled and restrict access.
Other	Live Updates	All limit switches will trigger in the GUI within 0.5 seconds.

4.4.5. Verification Process

A survey of ten random students at Oregon State University will be taken with the goal of each defining the Graphical User Interface as "Usable" or "Not Usable". This will be determined by a combination of characteristics such as navigation, layout, formatting, and control inputs.

1. Users will navigate controls without guidance and be asked to show where they believe initialization, movement, and video controls are located.
2. User will demonstrate functionality of GUI
3. Users will take a survey and answer the question "Does the system have a GUI that is usable?"

4.4.6. References and File Links

[1] “Tkinter course - create graphic user interfaces in python tutorial,” *YouTube*, 19-Nov-2019. [Online]. Available: https://www.youtube.com/watch?v=YXPyB4XeYLA. [Accessed: 28-Apr-2023].

4.4.7. Revision Table

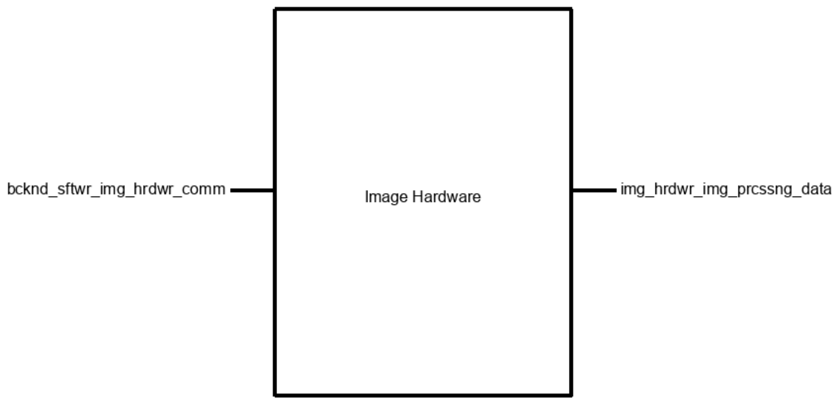
4/28/2023	Aleksi Hieta: Filled in info for all sections
5/14/2023	Aleksi: Revised all sections

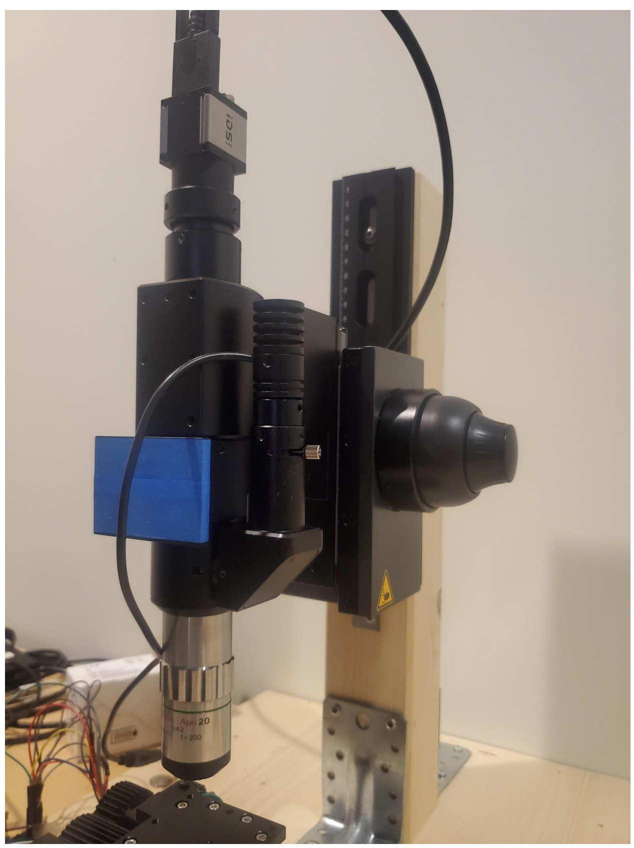
4.5. Image Hardware

4.5.1. Description

This block contains all hardware components required to utilize the digital imaging sensor. The main components are the UI-3240CP-NIR-GL IDS Camera which is an infrared digital sensor which connects to via USB. This sensor requires a stand, a focus unit, a microscope light, and microscope lenses. The focus unit acts as a coupler between the sensor, microscope lens, stand, and microscope light.

4.5.2. Design





All components other than the microscope light were provided by the project partner due the large expense that high precision microscopes generate. The frame is made attaching a ThorLabs XT66P2/M rail to a 2x2 birch post. Attached to that is a manual Z axis adjustment unit and focus unit, the Mitutoyo FOCUSINGUNITA 378-705 and the VMU-V 378-505 for the digital sensor to lens adapter. The lens is a Mitutoyo M Plan Apo 20 which provides 20x magnification. The microscope light is a YEGREN 8mm White LED Microscope light. The digital sensor is the IDS UI-3240CP-NIR-GL and connects to a laptop via USB.

Description

Important information:

The objectives are optional. The microscopes also work with other objectives.

Features

Reflected illumination:	- Telecentric system with aperture stop system - Fibre-optic illuminator (optional) is required
Light source:	Halogen bulb (21V, 150W) (optional)
Applicable Wavelength:	Near-infrared and visible radiation
Fibre optic illumination unit mount:	Yes
Mass:	570 g

4.5.3. General Validation

This block's purpose is to create a single focused monochrome image from the digital sensor when sent a control signal from a linux based computer.

The hardware pieces were chosen to be generally Mitutoyo due to their reliability in the microscope field. By trying to limit hardware to only one or two companies the compatibility between parts is generally better, which is why specifically the focus unit, adapter, and lens are all from Mitutoyo.

4.5.4. Interface Validation

bcknd_sftwr_img_hrdwr_comm

Protocol	UDP	Analyzing the UDP data protocol requires use of a serial port monitor between the USB port in on the laptop and USB port out on the digital image sensor
Protocol	Python 3 on Linux	The project partner has stated that the project must run on a system running linux

		coded in Python. IDS includes drivers and example code for the digital sensor
Protocol	COM port over USB	Using the Linux command lsusb -v the COMM port (serial port in linux) the connection of the sensor can be verified.

img_hrdwr_img_prcssng_data

Datarate	Grayscale	The subject will be displayed in black and white for both video and saved images.
Datarate	1280x1024 pixels	When inspecting saved images from the stage, a property of 1280x1024 pixels will be observed.
Messages	.jpg images	All images will have the .jpg extension.

4.5.5. Verification Process

To test the input and output properties of the image hardware download the correct drivers from the IDS link [1] and the example code provided should allow for images to get captured. OpenCV can be used to verify the properties of the images captured.

1) Download IDS drivers for digital image sensor onto computer
2) Turn on microscope light and adjust Z axis until microscope light is completely focused
3) Create folder and download python code for image taking into folder
4) Run included python code on the terminal of choice
5) Check folder for image and verify data type of photo by inspecting photo. This can be done by right clicking, or running included python code which requires the OpenCV library.

For the backend to image hardware

1) Download IDS drivers for digital image sensor onto computer
2) Open the terminal and type lsusb -v
3) Check the results of the terminal for IDS Imaging Development Systems on the USB bus
4) Check the bit length and protocol given by the terminal.

4.5.6. References and File Links

[1] Incony, “Microscope Unit VMU-V,” Mitutoyo, Product: Microscope Unit, https://shop.mitutoyo.eu/web/mitutoyo/en/mitutoyo/05.03.09c/Microscope%20Unit/$catalogue/mitutoyoData/PR/378-505/index.xhtml (accessed May 14, 2023).

4.6. Image Processing

4.6.1. Description

The image processing block is a code based block that takes multiple images in the form of JPG from the digital microscope and processes them into a single image with improved depth and greater size than a single image from the imaging sensor. It does this through two processes, image stitching and focus stacking. Focus stacking takes multiple images of the same subject with different parts of the image in focus to improve the depth of field which reduces the blurriness of the image. Image stitching takes multiple images from different areas and runs an algorithm to stitch the seams of the images, creating a picture that is multiple times larger than the size of the image sensor could normally produce.

4.6.2. Design

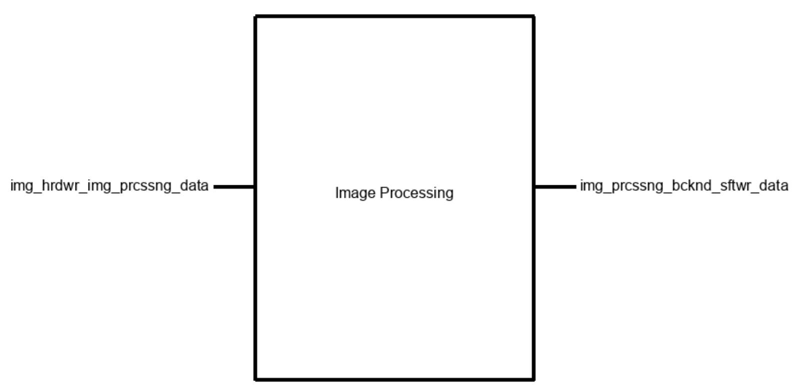


Image processing block uses FIJI and ImageJ, an open source image processing software used mainly for medical and biological microscope specimens. The backend interfaces with this software using user written macros. When images are captured the relative position of the image is recorded in the file name. The macro uses this to pass it into the correct image stitching and focus stacking algorithm. ImageJ exports the final processed image as a .tiff file which is a large file size and is converted back to .jpg using OpenCV.

4.6.3. General Validation

ImageJ is a large-scale open source project with plenty of documentation and proof of its abilities on large scale organic specimens. The computer chip surfaces that this project intends to use for testing are much easier to process due the rigidity and high area of contrasts of the transistors used in computer chips. Image stitching is an area of image processing that is matured and since the camera movements are precise there should be very little overlap necessary to process the images. There is currently no set orientation for the stage so before image taking begins the camera sensor and stage must be aligned so that image stitching can be done to the correct images as defined in the backend code.

4.6.4. Interface Validation

img_hrdwr_img_prcssng_data

Datarate	Grayscale	The grayscale data rate is chosen due to the black and white nature of the infrared sensor. Color images require 3 color channels while monochrome (grayscale) requires one, saving space and processing power
Datarate	1280x1024 pixels	When inspecting saved images from the stage, a property of 1280x1024 pixels will be observed.
Messages	.jpg images	All images will have the .jpg extension.

img_prcssng_bcknd_sftwr_data

Datarate	Grayscale	The grayscale data rate is chosen due to the black and white nature of the infrared sensor. Color images require 3 color channels while monochrome (grayscale) requires one, saving space and processing power
Datarate	1160x960 - 3280x2824 pixels	Stitched images have a varying size due to requiring overlap. In general overlap is around 15% of the image but this percentage is subject to change based off the precision of the stage. Testing has shown that the provided pixel size ranges are average for a 3x3 image where

		overlap is sparse. (10%)
Messages	.jpg images	When inspecting saved images from the stage, a property of .jpg will be observed
Other	1 image	Only one process image will be saved after image processing which can be inspected by checking the processed image file.

4.6.5. Verification Process

Verification is split into two parts as the Engineering Requirements for Focus Stacking and Image Stitching

Image Stitching

1) Download microscope image samples provided by ImageJ or use images taken by Image Hardware block in either 2x2 or 3x3 grid format into a file
2) Copy the Image Stitching macro into the file
3) Change the image stitching macro to reflect the X and Y number of images being stitched ie X = 3, Y = 3 for a 3x3 image grid
4) Run the macro on the terminal of choice
5) Check folder for image and verify data type of photo by inspecting photo. This can be done by right clicking, or running included python code which requires the OpenCV library. Additionally visually inspect the stitch points of the photo for any artifacts

Focus Stacking

1) Download microscope image samples provided by ImageJ or use images taken by Image Hardware block with Z axis movement of 20um 4 times for 4 seperate .jpg images into a file
2) Copy the focus stacking macro into the file
4) Run the macro on the terminal of choice
5) Check folder for image and verify data type of photo by inspecting photo. This can be done by right clicking, or running included python code which requires the OpenCV library. Additionally visually inspect the photo for artifacts caused by image processing

4.6.6. References and File Links

[1] “Fiji downloads,” ImageJ Wiki, https://imagej.net/software/fiji/downloads (accessed May 14, 2023).

[2] “Python,” ImageJ Wiki, https://imagej.net/scripting/python (accessed May 14, 2023).

4.6.7. Revision Table

4/28/2023	Anthony Bryan: Filled in info for all sections
5/14/2023	Aleksi: Edited Interface Validation

4.7. Backend

4.7.1. Description

The backend is required to run in a linux environment and to be coded in the python language due to project partners familiarity as well as concerns for open source development. Additionally any code not written must be free and open source so that this project can remain open source.

The main function of the backend is to handle the serial port communication between the input controls, digital sensor, and motor control hardware as well as run the necessary code to move the stage precisely and capture images defined on user input through the GUI.

4.7.2. Design

The linux kernel is 5.19.0-41 generic Ubuntu. Anaconda an installation handler was installed in order to handle environment management. The main packages required were OpenCV, NumPy, Tkinter, and ImageJ. OpenCV is necessary for general image manipulation in Python such as cropping images and saving images. The latest version 4.5.4 for Ubuntu was installed. Tkinter version 8.6 was used for the GUI. ImageJ version 1.53t was installed using Anaconda to handle the environment, additionally PyImageJ is necessary for the macros used for interfacing with ImageJ. PiP was generally used to handle package installation.

4.7.3. General Validation

The validation consists of checking the serial interfaces of the stage microcontroller, input control microcontroller, and the digital sensor as well as ensuring that the packages and environments installed to run the image taking and processing parts of the code are stable and do not cause the system to crash,

Since the project is running on linux, serial ports are extremely easy to check using the terminal, and give a much more detailed look into the details of the serial communication. Arduino IDE is a tool that can be used to check the BAUD rates and communication protocols of both the stage microcontroller and the input controls.

Anaconda is used to run virtual environments and package management to ensure the programs run into less dependency errors. Many of the libraries used are coded either in C for the ueye drivers used for the imaging sensor, and java for the ImageJ library. This does cause some memory issues in the system, but these are fixed with a restart of the GUI

4.7.4. Interface Validation

grphcl_sr_ntrfc_bcknd_sftwr_data

Messages	Values	Different values for the speed in X, Y, and Z axes can be handled.
Other	Live Updates	All limit switches will trigger in the GUI within 0.5 seconds.
Other	Error Handling	Errors in direct G-Code or speed will be handled and restrict access.

bcknd_sftwr_img_hrdwr_comm

Protocol	COM port over USB	Analyzing the UDP data protocol requires use of a serial port monitor between the USB port in on the laptop and USB port out on the digital image sensor
Protocol	UDP	The project partner has stated that the project must run on a system running linux coded in Python. IDS includes drivers and example code for the digital sensor
Protocol	Python 3 on Linux	Using the Linux command lsusb -v the COMM port (serial port in linux) the connection of the sensor can be verified.

bcknd_sftwr_mchncl_stg_nd_cntrl_hrdwrfrmwr_data

Datarate	115200 baud	The Grbl firmware which runs on the MKS DLC control board functions at this data rate.

Messages	G-code commands (Ex: G21 G91 G01 X0.1 F30)	The Grbl firmware which runs on the MKS DLC control board takes this command format as input.
Other	Connector: USB Type-A to USB Type-B	The Grbl firmware which runs on the MKS DLC control board utilizes this protocol.

img_prcssng_bcknd_sftwr_data

Datarate	Grayscale	The grayscale data rate is chosen due to the black and white nature of the infrared sensor. Color images require 3 color channels while monochrome (grayscale) requires one, saving space and processing power
Datarate	1160x960 - 3280x2824 pixels	Stitched images have a varying size due to requiring overlap. In general overlap is around 15% of the image but this percentage is subject to change based off the precision of the stage. Testing has shown that the provided pixel size ranges are average for a 3x3 image where overlap is sparse. (10%)
Messages	.jpg images	When inspecting saved images from the stage, a property of .jpg will be observed
Other	1 image	Only one process image will be saved after image processing which can be inspected by checking the processed image file

4.7.5. Verification Process

For the backend to GUI

1) Open Linux Terminal and run backend code
2) Go to the Control Tab in the GUI

3) Check click every option under Directional Steps and verify that the correct value is displayed in the Position setting of the GUI

4) Change input of the Directional Step Speeds and check terminal for message showing values have changed

5) Enter characters into Driect G-Code, hit Send G-Code and verify in terminal that message is passed through

6) Verify input entering in the Video tab are working by entering new values and checking the terminal after hitting Take Image button

For the backend to image hardware

1) Download IDS drivers for digital image sensor onto computer

2) Open the terminal and type lsusb -v

3) Check the results of the terminal for IDS Imaging Development Systems on the USB bus

4) Check the bit length and protocol given by the terminal.

Image Stitching

1) Download microscope image samples provided by ImageJ or use images taken by Image Hardware block in either 2x2 or 3x3 grid format into a file

2) Copy the Image Stitching macro into the file

3) Change the image stitching macro to reflect the X and Y number of images being stitched ie X = 3, Y = 3 for a 3x3 image grid

4) Run the macro on the terminal of choice

5) Check folder for image and verify data type of photo by inspecting photo. This can be done by right clicking, or running included python code which requires the OpenCV library. Additionally visually inspect the stitch points of the photo for any artifacts

Focus Stacking

1) Download microscope image samples provided by ImageJ or use images taken by Image Hardware block with Z axis movement of 20um 4 times for 4 seperate .jpg images into a file

2) Copy the focus stacking macro into the file

4) Run the macro on the terminal of choice

5) Check folder for image and verify data type of photo by inspecting photo. This can be done by right clicking, or running included python code which requires the OpenCV library. Additionally visually inspect the photo for artifacts caused by image processing

4.7.6. References and File Links

[1] “Installing on Linux,” Installing on Linux - Anaconda documentation, https://docs.anaconda.com/free/anaconda/install/linux/ (accessed May 14, 2023).

[2] “Install ubuntu desktop,” Ubuntu, https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview (accessed May 14, 2023).

[3] “Install opencv-python in ubuntu,” OpenCV, https://docs.opencv.org/3.4/d2/de6/tutorial_py_setup_in_ubuntu.html (accessed May 14, 2023).

[4] “Installing NumPy,” NumPy, https://numpy.org/install/ (accessed May 14, 2023).

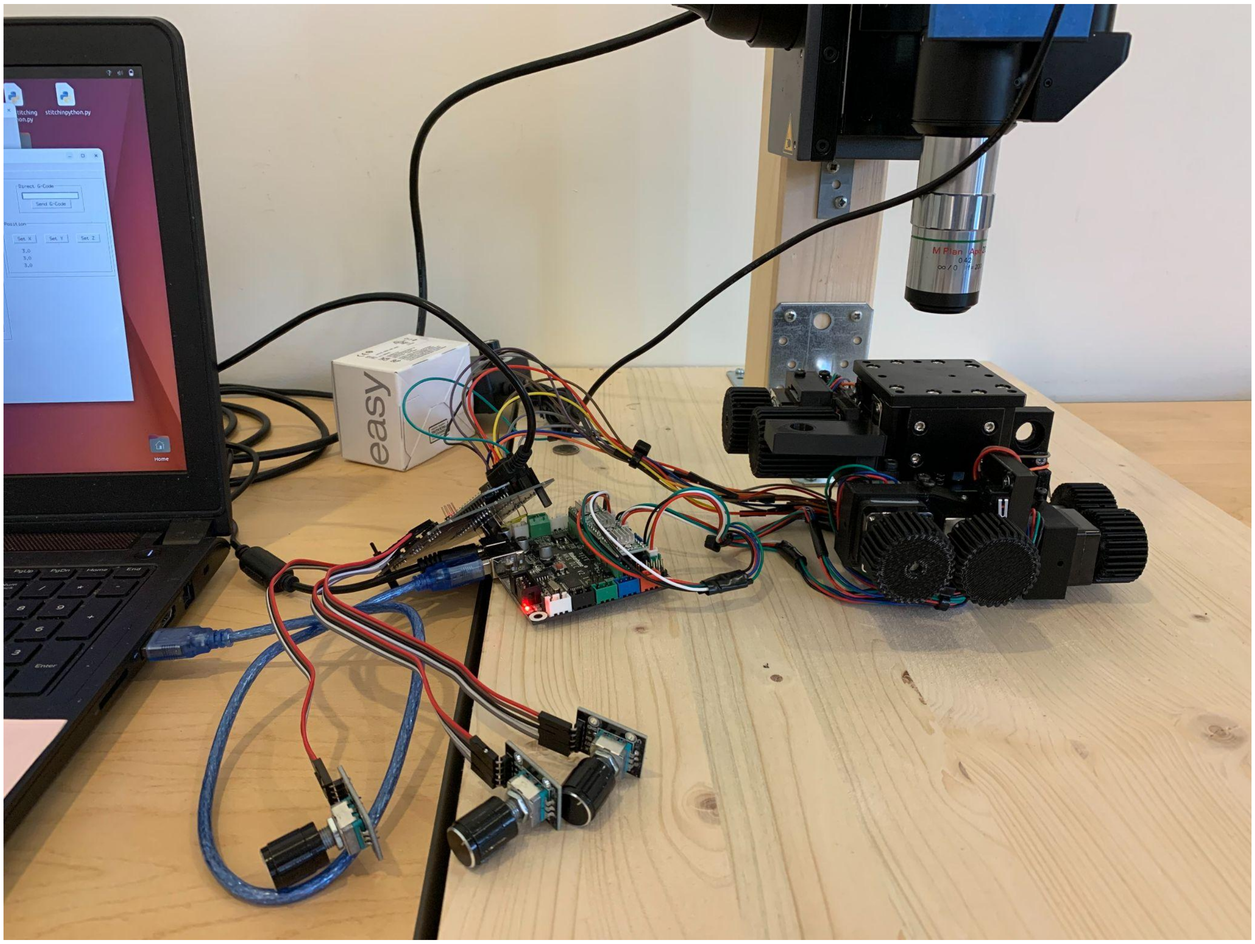
4.7.7. Revision Table

4/28/2023	Anthony Bryan: Filled in info for all sections

5. System Verification Evidence

5.1. Universal Constraints.

5.1.1. The system may not include a breadboard.
Final system:



The final system does not include a breadboard as the only breadboard used in development of the system was replaced by the input controls PCB.

5.1.2. The final system must contain a student designed PCB.
An input controls PCB has been designed for the system (Section 4.3).

5.1.3. All connections to PCBs must use connectors.
The input controls PCB uses connectors for the encoders and limit switches (Section 4.3).

5.1.4. All power supplies in the system must be at least 65% efficient.

1. The power supply used to power the stage control board is rated as VI, meaning a power efficiency of at least 87%.
Control board power supply:

Reference to performance requirements for power supply level standards: https://slpower.com/data/collateral/PW174KB_DS.pdf

5.1.5. The system may be no more than 50% built from purchased 'modules.' The system includes the following blocks:

1. Mechanical Stage and Control Hardware/Firmware: Custom-made
2. Limit Switches: Custom-made
3. Input Controls: Custom-made
4. GUI: Custom-made
5. Backend Software: Custom-made
6. Image Hardware: Purchased
7. Image Processing: Purchased

So, the system has 6/8 custom-made blocks.

5.2. Requirements

5.2.1. 10um Accuracy (X and Y directions)

5.2.1.1. Project Partner Requirement: Project is able to replicate the performance of high end XYZ stages already established.

5.2.1.2. Engineering Requirement: The system will move with a resolution of 10 um in the X and Y directions with a 10% tolerance.

5.2.1.3. Verification Process:
1) Measure stage position with a dial gauge. 2) Use the GUI to instruct the stage to move 10 um in the X direction. 3) Measure

the new position and ensure the stage really moved 10 um +/- 10%. 4) Repeat for the Y direction.
Pass Condition: The stage can move in 10um increments with +/-10% tolerance.

5.2.1.4. Testing Evidence: Video: https://drive.google.com/file/d/1txrZqJ_5FmuRFkGT76KondXMjLUsJGXn/view?usp=share_link

5.2.2. Complete Stage Movement

5.2.2.1. Project Partner Requirement: Stage can move in XYZ directions.

5.2.2.2. Engineering Requirement: The system will move the full distance of at least 6mm in X, Y, and Z directions.

5.2.2.3. Verification Process:
1) Move the stage in the positive X direction completely to one limit switch. 2) Measure the stage position. 3) Move the stage in the other direction completely to the other limit switch. 4) Measure the stage position and ensure it was able to move at least 6mm. 5) Repeat for the Y and Z directions.
Pass Condition: The stage moves with a full range of 6mm in all directions.

5.2.2.4. Testing Evidence: Video: https://drive.google.com/file/d/1Ghw1IAK-dGCLxkie1IC6WYYImXCbMbNX/view?usp=share_link

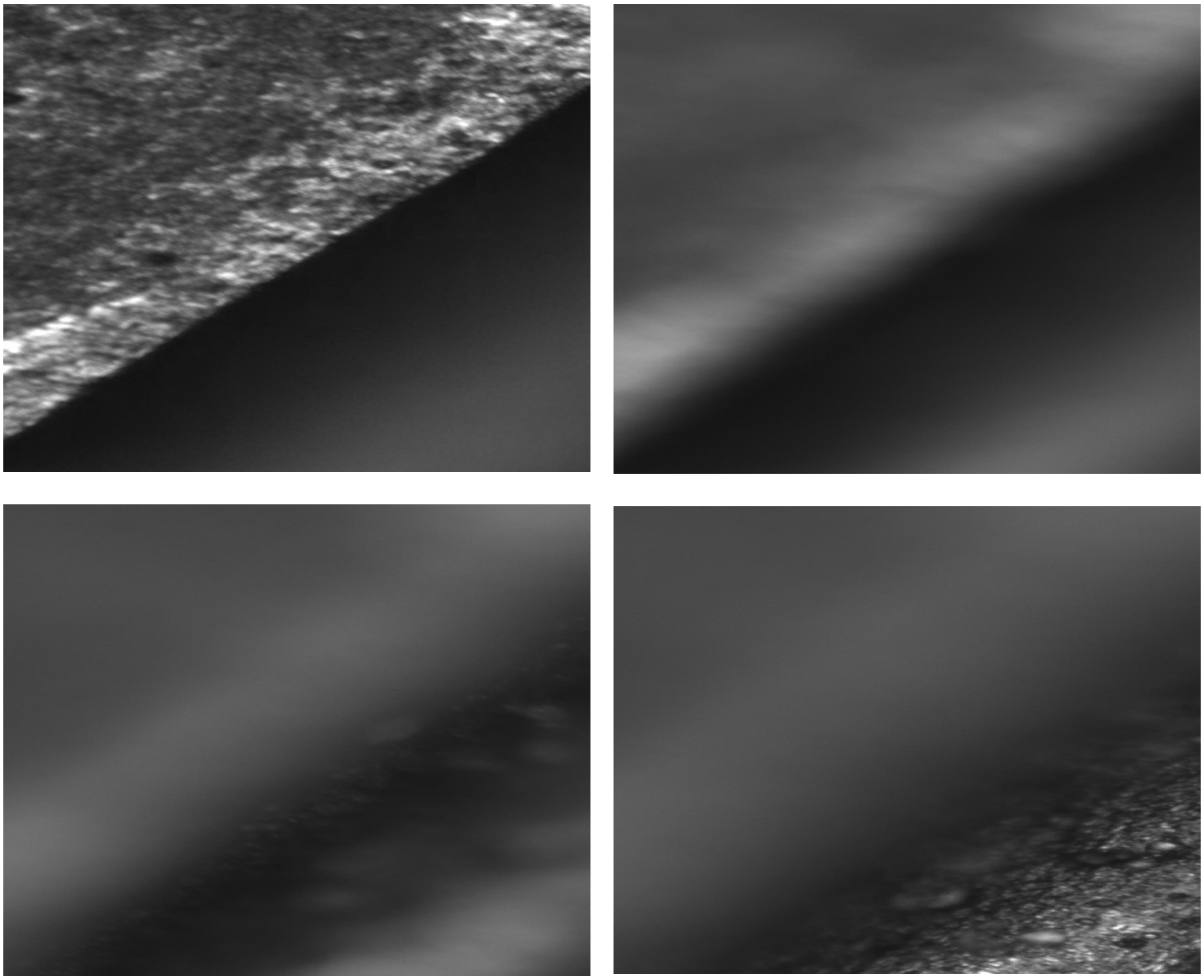
5.2.3. Depth of Field

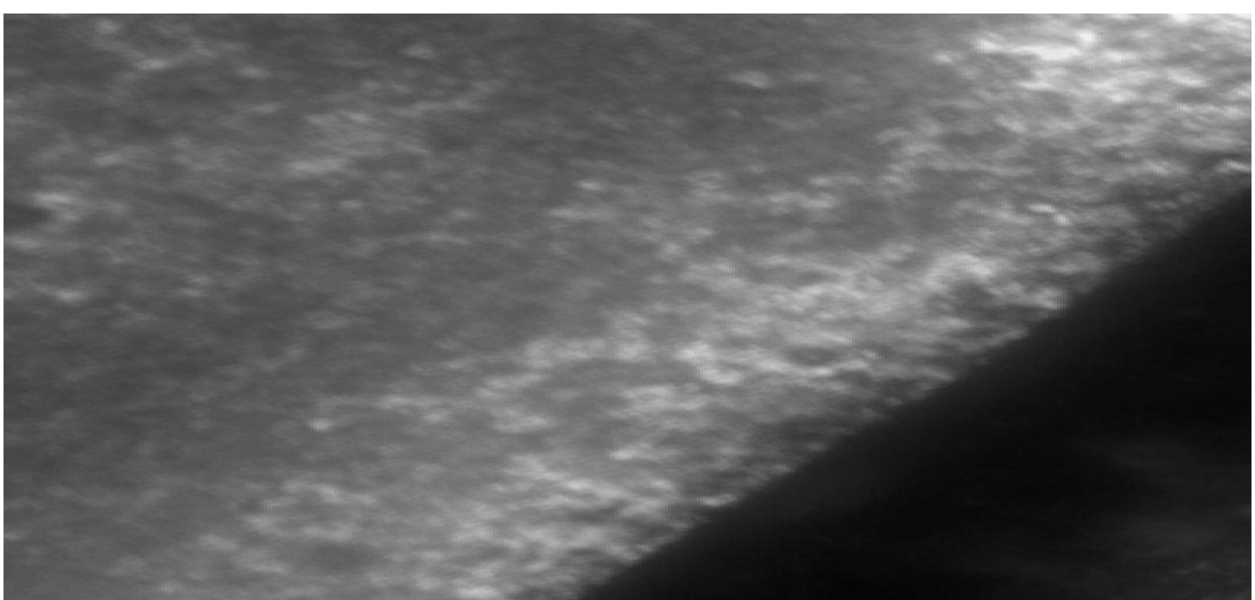
5.2.3.1. Focus stacking increases the depth of field of the images taken of the subject.

5.2.3.2. The system will produce at least 4 different focus depth images per sampling action and a composite image based on these.

5.2.3.3. Take an image of a non flat area of the chip surface and select the option to increase image focus. Check the file location of the .jpg images and determine if 4 unprocessed images taken at different Z axis exist and that 1 processed image exists. Estimate depth of field of an unprocessed image and a processed image using a depth of field calculation. * (this is an estimate because it is difficult to determine the actual focus area since the change is gradual it is difficult to determine the actual focus plane, additionally the height changes of the chips surface are difficult to determine)

5.2.3.4. Testing Evidence: Video and Images





VideoLink:
https://drive.google.com/file/d/1RT6w_JRJRuDWN84ryyfcKOcgPlWcHAev/view?usp=sharing

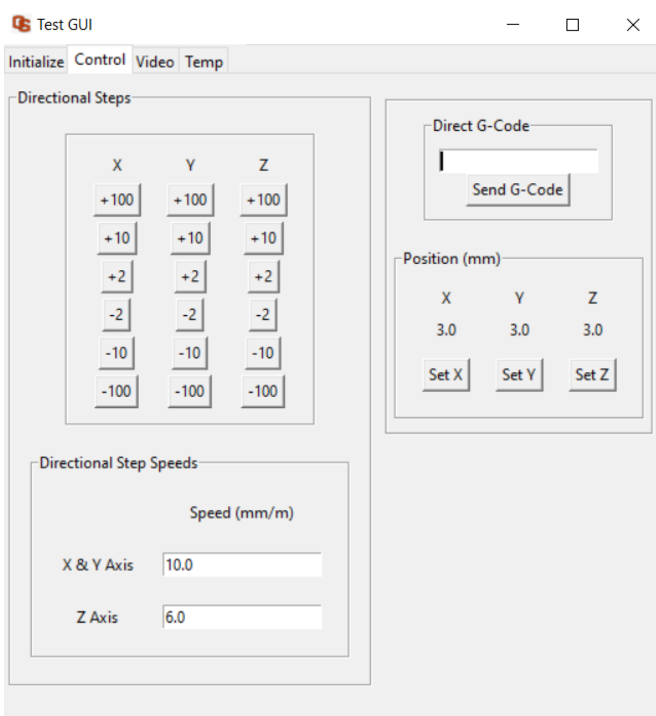
5.2.4. GUI Usability

5.2.4.1. Stage must be usable by graduate level students.

5.2.4.2. The system will have a GUI indicated as usable by 9 out of 10 university students.

5.2.4.3. A survey of ten random students at Oregon State University will be taken with the goal of each defining the Graphical User Interface as "Usable" or "Not Usable". This will be determined by a combination of characteristics such as navigation, layout, formatting, and control inputs.

5.2.4.4. Testing Evidence:
Signature Link:
https://drive.google.com/file/d/1y4Ig-3Aw7jJvmHvbjHQvr1NgX1yZJ0Oi/view?usp=share_link
Video Link:
https://drive.google.com/file/d/1TplygImpNk04CfuctfRpb52DWLA2U8a7/view?usp=share_link



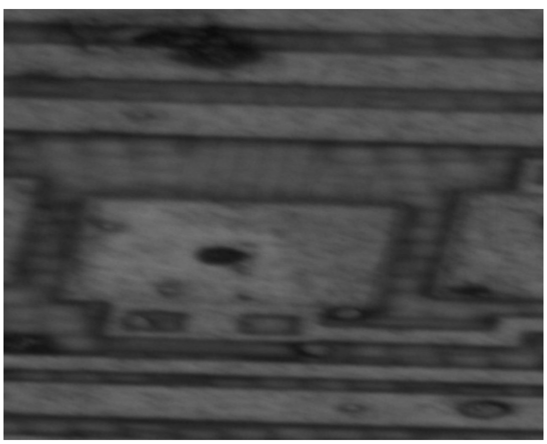
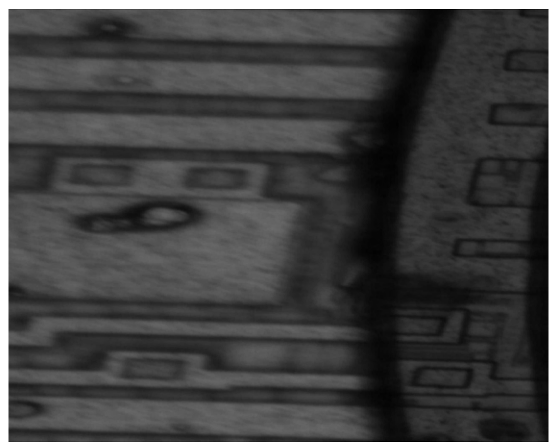
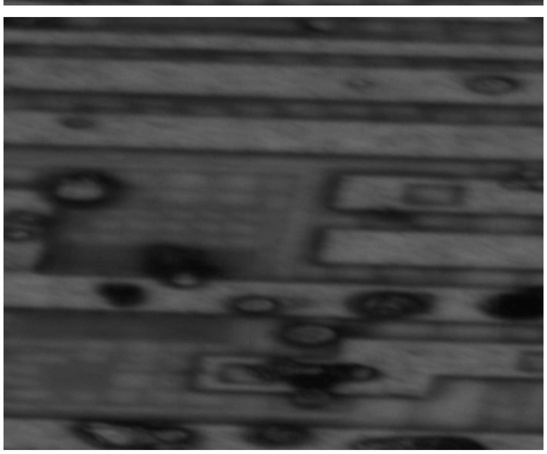
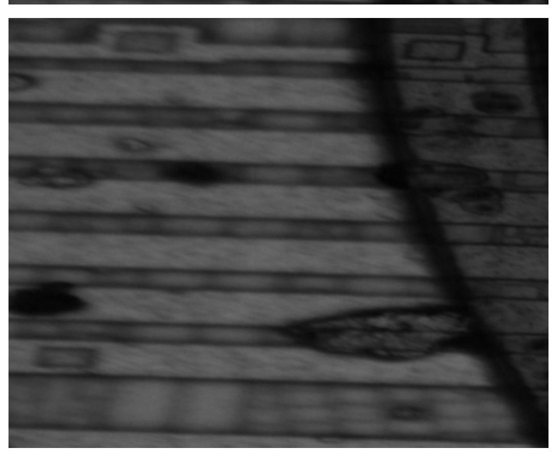
5.2.5. Image Stitching

5.2.5.1. Images are stitched thoroughly and without evidence of cut edges.

5.2.5.2. The system will produce final images that 9 out of 10 university students state they can clearly articulate details on subject matter and do not have artifacts that are overly distracting in the image.

5.2.5.3. 1. Take an image of chip surface larger than sensor size (1280x1024 Pixel) such that image stitching occurs. Download processed images Show processed images to 10 university students and ask "can you clearly articulate details on subject matter and do you see any artifacts that are overly distracting in the image"

5.2.5.4. Testing Evidence: Photos and Signatures and Video


Signature Link:
https://drive.google.com/file/d/1LkyqvYOjfCxSai75nGLjZIxLY1eMdoa0/view?usp=share_link

Video Link:
https://drive.google.com/file/d/1zNCaZT_adbPFt1zMrAW6JUd1kLILmnwI/view?usp=share_link

5.2.6. Z-Axis Accuracy

5.2.6.1. Project Partner Requirement: Project is able to replicate the performance of high end XYZ stages already established.

5.2.6.2. Engineering Requirement: The system will move with a resolution of 10 um in the Z direction with a 10% tolerance.

5.2.6.3. Verification Process:
1) Measure stage position with a dial gauge. 2) Use the GUI to instruct the stage to move 10 um in the Z direction. 3) Measure the new position and ensure the stage really moved 10 um +/- 10%.

5.2.6.4. Testing Evidence: Video:
https://drive.google.com/file/d/1txrZqJ_5FmuRFkGT76KondXMjLUsJGXn/view?usp=share_link

5.2.7. Limit Switches

5.2.7.1. Project Partner Requirement: The stage will be limit switched so that the user can't move the stage beyond its physical limits.

5.2.7.2. Engineering Requirement: Stage motion will be stopped when the stage comes within 1mm of its physical limits.

5.2.7.3. Verification Process:
1) Use the GUI to move the stage in the positive X direction until it hits a limit switch and stops. 2) Measure the position of the stage. 3) Manually move the stage until the physical limit is reached. 4) Measure the new stage position and ensure it is within 1mm of where it was stopped by the limit switch. 3) Repeat for the negative X direction as well as the Y and Z directions.

Pass Condition: The stage is stopped by the limit switches within 1mm of its limits.

5.2.7.4. Testing Evidence: Video: https://drive.google.com/file/d/1t-s1gcU6xqc_Dm63CQhte9jNZ40x5Hcd/view?usp=share_link

5.2.8. Pinning Corners

5.2.8.1. Users can determine a section of the stage for the microscope to take an image of.

5.2.8.2. Dimensions of the processed image will match the user's inputted dimensions by within 15um.

5.2.8.3. Load chip surface with known transistor lengths onto the XYZ stage. Input the first X, Y coordinate of the chips surface denoting the top left corner of the image Input the second X, Y coordinate of the the chips surface denoting the bottom right corner of the image Use the known transistor size to measure the height and width of the image and compare to the inputted coordinates and determine if processed image is within 15um

5.2.8.4. Testing Evidence: Photo and Video
Video: https://drive.google.com/file/d/1bTNnHnOAB6PidRRfIEUD0Swpb5zfhpae/view?usp=share_link

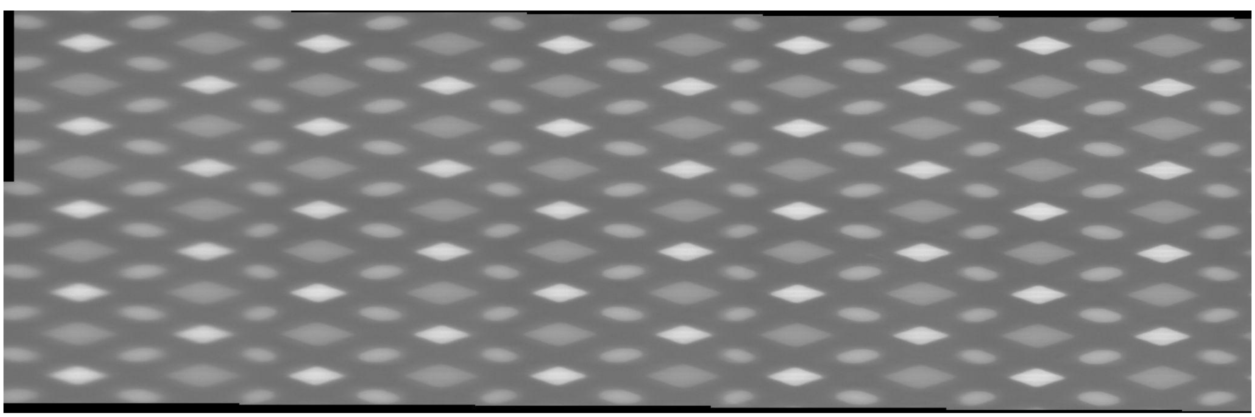


Image is of Samsung Ultra 20 screen pixels are 5um height and width, stage is set to take photo 25um by 25um

5.3. References and File Links
https://drive.google.com/drive/folders/1JxNEmX5gXKGU875qazsJLNR7S4YzRnnx?usp=share_link

5.4. Revision Table

3/12/2023	Aleksi Included content from student portal and updated requirements
3/12/2023	Logan: Updated sections.
5/8/2023	Aleski, Logan, Anthony: Updated sections and added evidence and video links.

6. Project Closing

6.1. Future Recommendations

6.1.1. Technical recommendations

1) Add position encoders to the stage.

Currently, the system relies on the precision of the stage movements in order to track its position. So, if the stage is instructed to move a certain amount by the user or imaging process and the stage doesn't quite move that exact amount, the system does not have a precise reading of the stage's current position. Adding some form of precise position encoder would solve this problem and lead to greater accuracy in stage movement overall. This can be done using linear scales or digital dial gauges for example. A linear scales reference provided by Dr. Immler can be consulted[1].

2) Recreate 3D printed hardware for a larger stage.

The current implementation of the 3D printed gears and adapters (which fix motors to the stage) are designed for a particular microscope stage with limited range of motion. Recreating the design for a larger stage would increase the overall range of motion dramatically, allowing for a larger viewing area and potential for larger chip samples. This might require the use of higher torque motors to move a larger stage as well as entirely new fastening mechanisms, so it is not clear how much of the current design could be reused for this purpose. A larger stage can be purchased from AliExpress[2].

3) Add a homing process to the system.

Allowing the user to automatically move the stage to the center of its viewing area (in all directions) with the push of a button would make for a simpler user experience overall. This could be done by having the stage move to the limit switches in all directions and move a specified amount to the center, depending on the fully measured range of motion, possibly utilizing position encoders as discussed above. A guide to homing with G-Code commands can be utilized[3].

4) Integrate limit switches directly with motor control firmware.

The motor control firmware used for the current implementation of the system, GRBL, allows for directly interfacing with limit switches. That is, the limit switches could connect directly to the motor control board instead of to the input controls PCB. However, GRBL treats hitting limit switches as a fatal error, requiring a reset sequence to be run in order to unlock the firmware. This is due to its common use case for 3D printers and CNC milling machines where hitting limit switches is a truly fatal error. So, for the purposes of this project, utilizing or building new motor control firmware which could directly handle the limit switch input would allow for a simpler and more robust system all around. The GRBL Wiki which discusses the use of limit switches should be considered[4].

References:
[1] Yuriy's Toys. (2023, May 1). *Selecting Scales for a DRO* [Online]. Available:
https://www.yuriystoys.com/2013/12/selecting-scales-for-dro.html

[2] AliExpress. (2023, May 1). *XY Axis 90*90mm Trimming Station Manual Displacement Platform Linear Stage Sliding Table XY90-LM XY90-C LY90-R Cross Rail* [Online]. Available:
https://www.aliexpress.us/item/2251832661130647.html?spm=a2g0o.productlist.main.21.4e4a5cb8VC2sYs&algo_pvid=def58084-1500-4622-9472-89286c8e1f13&algo_exp_id=def58084-1500-4622-9472-89286c8e1f13-10&pdp_npi=3%40dis%21USD%2186.14%2186.14%21%21%21%21%21%40211bea6216835895006687292d07ea%2165593584504%21sea%21US%214299595612&curPageLogUid=ch39UYFoAsgh

[3] All3DP. (2023, May 1). *G28 G-code: All You Need to Know about Homing* [Online]. Available: https://all3dp.com/2/g28-g-code-homing/

[4] Github. (2023, May 1). *GRBL Wiki* [Online]. Available:
https://github.com/gnea/grbl/wiki

6.1.2. Global impact recommendations

The materials for this project directly support the industrialization and globalization of developing countries. Some parts are too expensive to find domestic and ethical alternatives such as the stage hardware, but improvements for sourcing the PLA used for the gear adapters, and using a genuine Arduino Nano for the input controls would fall within the budget, albeit still raising the price and lessen the impact caused by sourcing materials from recently globalized markets. A workers rights index such as the ITUC Global Rights Index [1] can be used to determine overall worker rights in various countries to better help choose less exploitative options for the project.

This project is meant to be open sourced and will be uploaded to a public github repository but there is currently no plan to increase the scope of the project past the graduate students who will use this project working with Dr. Immler at OSU. The format of the github repository can be improved to improve readability as well as improve keyword detection on search engines such as Google. The following paper [2]“Factors for Improving Google Search Rank” explains methodology for increasing search results.

[1]International Trade Union Association,2021 ITUC Global Rights Index. (2022, June 30). *Home*. ITUC GRI 2021. Retrieved April 28, 2023, from https://www.ituc-csi.org/2021-global-rights-index

[2]Ziakis, C.; Vlachopoulou, M.; Kyrkoudis, T.; Karagkiozidou, M. Important Factors for Improving Google Search Rank. *Future Internet* 2019, *11*, 32. https://doi.org/10.3390/fi11020032

6.1.3. Teamwork recommendations

Creating a repository in Github would be helpful for sharing future iterations of the code off of the existing work done on the project. A localized center for the GUI and image processing software would allow for streamlined progress.

Collaboration between the image processing and GUI is key for determining positioning of images for stitching along with the functionality aspect of the user interface. These two blocks are very closely related and were taken on by two people.

Development of the mechanical hardware and firmware are best handled with the established libraries and CAD files. It is recommended to have the team utilize these with one individual managing tolerances and precision.

A central channel of communication is important to clarify any integration issues between the software and hardware. For the current project a communication channel was established through discord and used continuously through the three terms.

An issue with the current input controls is an inaccuracy with the rotary encoders. When turning the knobs at a high velocity the knobs are prone to mis-inputs. Likewise, the encoders currently require two ‘ticks’ per input unit. Improvement on this through a different medium like better encoders, joysticks, or buttons may allow for more effective and consistent controls.

[1] B. Hou, “Design, Optimization and Compensation of A High Precision Single-excitation Absolute Capacitance Angular Encoder Up to ±4",” *66tie10-hou-2886762*. [Online]. Available: https://ieeexplore.ieee.org/ielaam/41/8728044/8595422-aam.pdf. [Accessed: 20-Feb-2023].

6.2. Project Artifact Summary with Links

6.2.1. Mechanical Design

6.2.1.1. Adapters

Link: https://oregonstate.box.com/s/7zghxaol4gmo6c0g44sgqjlmmelgsmu4

Links to a Box folder with FreeCAD files and STL files for each of the 3D printed adapters.

X-adapter:

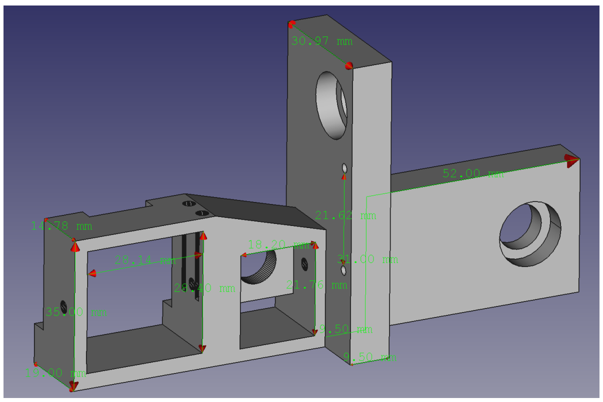


Fig. X-adapter CAD 1

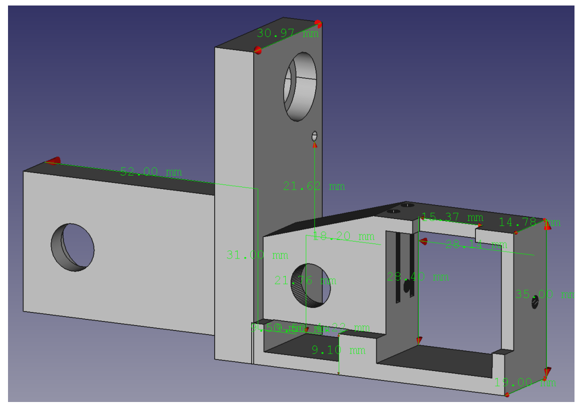


Fig.. X-adapter CAD 2

Y-adapter:

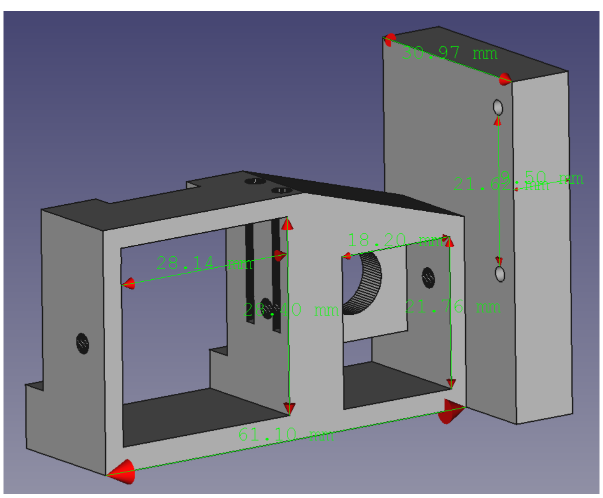


Fig. Y-adapter CAD 1

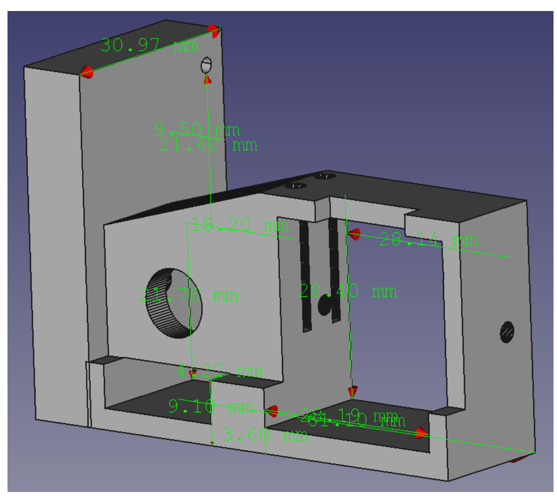


Fig. Y-adapter CAD 2

Z-adapter:

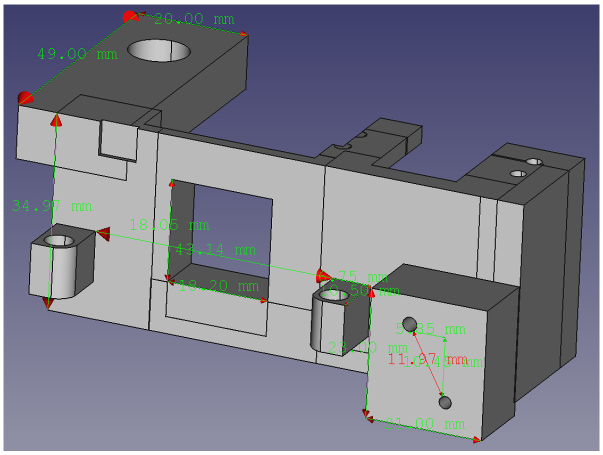


Fig. Z-adapter CAD 1

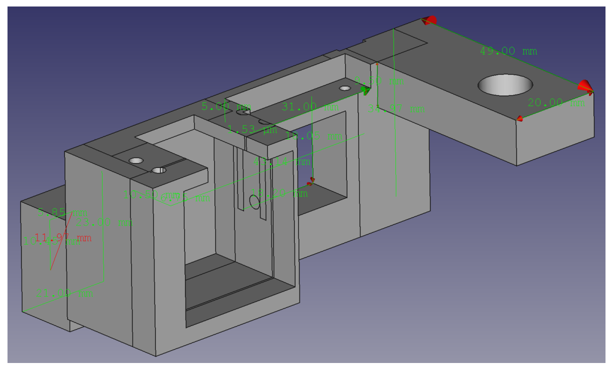


Fig. Z-adapter CAD 2

6.2.1.2. Gears

Link: https://oregonstate.box.com/s/aoq23dknqe1wh5slnc5o2g2pivapz5ma

Links to a Box folder with FreeCAD files and STL files for each of the 3D printed gears.

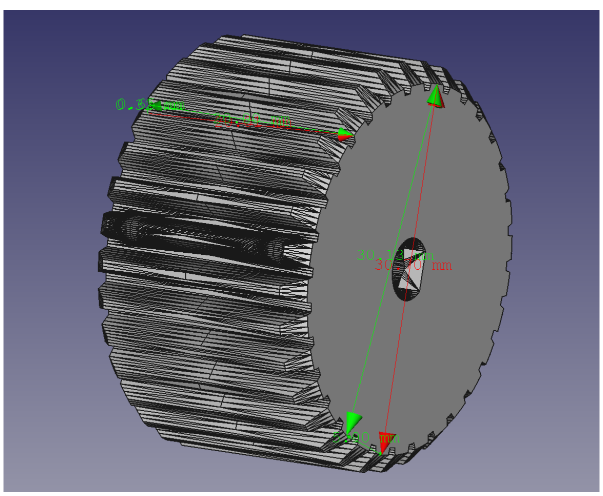


Fig. Motor gear CAD

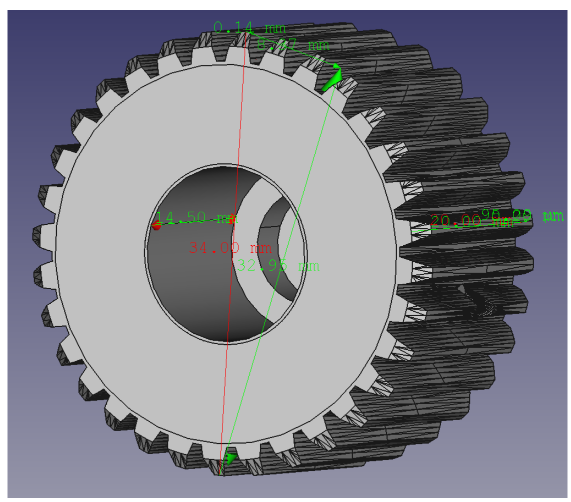


Fig. XY knob gear CAD

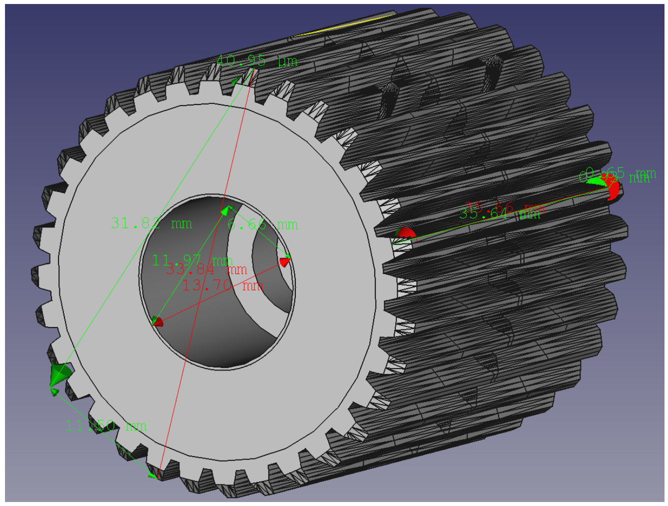


Fig. Z knob gear CAD

6.2.2. Input Controls PCB and Schematic

Link: https://oregonstate.box.com/s/kf052cqw3niu52jp6q8m6oj1bwyrams6

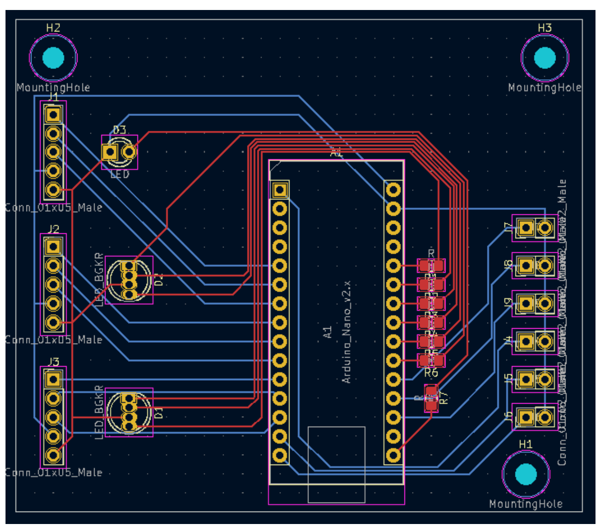


Fig. Input Controls PCB Layout

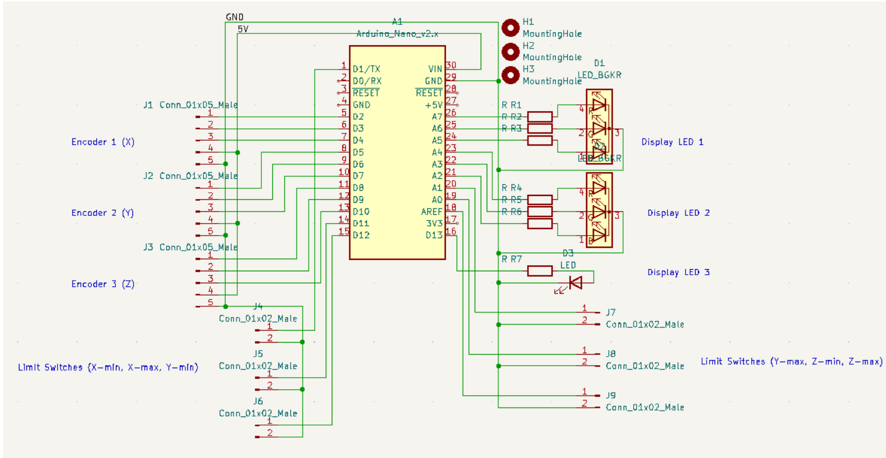


Fig. Input Controls Schematic

6.2.3. GUI Code

Link: https://oregonstate.box.com/s/mp35jzfc3jo5xlucfd11hom8qxyija4b

6.2.4. Image Processing

6.3. Presentation Materials

Project poster:

COLLEGE OF ENGINEERING | Electrical Engineering and Computer Science | ECE.04

PROJECT SUMMARY

- The purpose of this project is to create a low cost, high precision stage capable of XYZ directional movement along with a functional user interface and image processing software. When paired with a microscope setup, the system allows a user take high resolution images of computer chip surfaces.
- This project will be used by graduate students going to OSU to introduce them to the imaging processes for computer hardware security headed by Dr. Vincent Immler.
- The project contains a manual microscope stage, 3D printed adapters and gears, a motor control board, user control knobs, and an image sensor connected through a python based linux backend and GUI.

ENGINEERING REQUIREMENTS

- 10um resolution for stage motion.
- Full range of motion of at least 6mm in the X, Y, and Z directions.
- Focus stacking algorithm to improve image resolution.
- Graphical user-interface which can be easily utilized by any user.
- Image stitching algorithm to increase image size.
- Precision input control knobs for manual control of the stage.
- Limit switches will protect stage and motors from exceeding physical boundaries.
- Stage coordinates can be used to take images from user specified locations.

Oregon State University

PRECISION CHIP INSPECTION

10 micron precision, low-cost, motorized XYZ stage with image processing and interactive GUI.

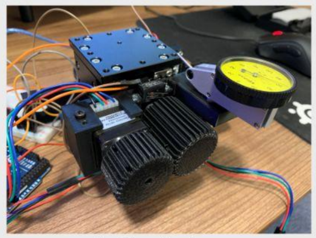


Fig. Stage Hardware

HARDWARE

- The stage itself was created by modifying a manual microscope stage. 3D printed adapters and gears were created in order to fix motors to the microscope stage knobs, allowing for motorized control of the stage in the X, Y, and Z directions.
- The motors are controlled using a control board running an open-source motor control firmware called GRBL. The firmware was calibrated by using a dial gauge with 2 micron accuracy which was mounted to the stage with custom 3D printed mounts.
- Limit switches are fixed to the stage to sense when it has moved to its physical limit in any direction.

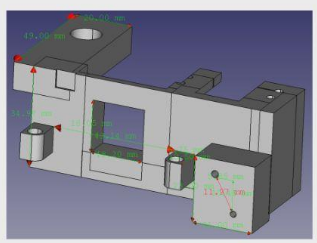
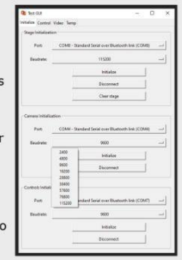


Fig. Z-Motor-Adapter CAD

SOFTWARE

- The interactive GUI is based in Python and utilizes tkinter for various control tabs.
- Threaded serial communication allows for simultaneous use of the stage, camera, and controls.
- Limit switches block movement of the stage to prohibit moving beyond safe ranges.



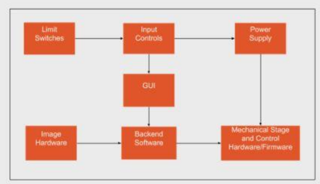


Fig. Block Diagram

BLOCK DIAGRAM

- The system takes in inputs from a user both through physical control knobs as well as the GUI. It also takes input from the limit switches and image sensor.
- The system outputs precise stage motion, graphical monitoring of the stage state through the GUI, as well as high resolution images.

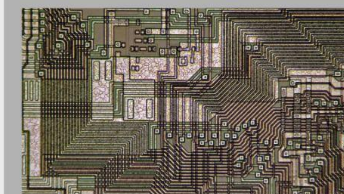


Fig. Filler Chip Image

IMAGE PROCESSING

- Image Stitching algorithm defines edges that are touching, keypoints are extracted from these areas based on areas of high contrast and matched.
- Focus Stacking algorithm also uses keypoints but uses a Laplacian gradient to determine areas where pixels are intensity is changing.

THE TEAM

- Logan York: Senior ECE
  Project Focus: Stage Hardware and Motor Control
  yorklo@oregonstate.edu
- Anthony Bryan: Senior ECE
  Project Focus: Image Processing
  bryanan@oregonstate.edu
- Aleksi Hieta: Senior ECE
  Project Focus: User Controls and GUI
  hietaa@oregonstate.edu
  ahieta2000@gmail.com

Team (left to right): Logan York, Anthony Bryan, Aleksi Hieta

Link to project showcase:
https://eecs.engineering.oregonstate.edu/project-showcase/projects/?id=xCw3xBKzDTT6PbTA