Group C Design Notebook for Water Sampler Project

Group Members: Grant Eyer, Christian Gratz, Garrett Matheny, Dorian Lewis, and Joshua Hagan

Table of Contents

Team Information

Team Contract

EGR 103 - Engineering Exploration 2
Water Sampling Project

Team Identity

This is the Group C team contract. This contract centers around the creation of a water sampler that detects the presence of Covid-19 RNA in sewage water. The samples of water that are analyzed will have been previously centrifuged will be placed under a sampling pipette using a rotating base. The pipette will be lowered such that it contains about 1 mL of water from the centrifuged water solution. The pipette will be removed from the solution, and the rotating base will move until a water absorbing polymer is directly under the pipette. The pipette will be lowered into the polymer, will release the water sample, and will return to its original location. This process will be monitored by a vision system, which will ensure that the pipette is always 1 cm above the location of the solids in the sample, regardless of where that line may be.

Membership

List Members and Contact information in alphabetical information

Team Member 1 Name: Grant Eyer Major: Chemical Engineering Cell Phone: (270) 556-2920 Email: granteyer@gmail.com	Strengths: Programming, time-management, calculations
Team Member 2 Name: Dorian Lewis Major: Mechanical Engineering Cell Phone: (270) 254-1663 Email: Dle282@uky.edu	Strengths: problem solving, design, research, building
Team Member 3 Name: Garrett Matheny Major: Chemical Engineering Cell Phone: (270) 970-3250	Strengths: Written and digital design, Hands on assembly,

Email: tksg2015@hotmail.com	
Team Member 4 Name: Christian Gratz Major: Mechanical Engineering Cell Phone: (270) 816-1667 Email: ctgr235@g.uky.edu	Strengths: Detail Oriented, problem-solving, skilled builder
Team Member 5 Name: Joshua Hagan Major: Mechanical Engineering Cell Phone: (270) 331-8162 Email: jtha297@uky.edu	Strengths: Time management, Versatile, Creative thinking

Personal Goals

This section gives a space for each team member to identify their own personal goals for this project. The purpose of this section is to allow each team member to honestly identify what they are attempting to get out of this so that a) no one becomes a negative contributor, and b) help with the step of identifying roles and responsibilities.

Grant Eyer: I hope that this experience will help me learn how to be more time-efficient and that I can become better at programming in MATLAB. I also hope that I can learn how to work more productively with others.

Garrett Matheny: Through this project, I hope to sharpen my skills in creative design and make myself familiar with CAD software. I would like to be able to work as effectively as possible.

Christian Gratz: During this project, I hope to increase my ability to work with others while also being effective in the assembly process of the over-all design. I would like to work in a timely fashion while also doing a working product.

Joshua Hagan: During this group effort, I would like to hone my ability to work with others. In addition, I would like to contribute the most effective way possible. Also, I would like to progress on learning new abilities displayed by other members.

Dorian Lewis: To gain experience with working as a team, building a complex structure, increase time management, and solving complex and everyday problem with science.

Roles and Responsibilities

Design Ownership:

The design project will be divided up into stages. Each member is expected to take **primary** responsibility for one of the design check points and **secondary** responsibility for another. Identify the member that will take primary ownership for ensuring the delivery and function, and a second team member to take the role of assisting that person. Each team member should have ONE primary and ONE secondary module for which they are significantly responsible. The individual elected as **team lead** has the overall design as his/her responsibility.

Overall Concept - Responsibility of the entire team

Sampler Stand Primary: Dorian Lewis
Secondary: Joshua Hagan

Rotational Base Primary: Christian Gratz
Secondary: Garrett Matheny

Pipette Squeeze Primary: Joshua Hagan
Secondary: Christian Gratz

Vision System Primary: Garrett Matheny
Secondary: Grant Eyer

Control Code Primary: Grant Eyer
Secondary: Dorian Lewis

Documentation

Each team will keep a physical copy of a design notebook where documents will be kept and reviewed by the instructor or TAs at different times throughout the semester.

Documentation guidelines: Location (Office 360, Google docs, Network Folder….)

Protection of project information

Documentation Ownership:

When completing a project the system must be handed over to another internal group or to a customer to use and maintain the product. In the case of this project you would be handing the prototype off to a group to bring it to manufacturing readiness. As part of that hand off it is important to create documentation for the device. **Each team member will take ownership of one documentation piece**. That documentation piece will be uploaded into Canvas on Week 12 into the documentation assignment as well as included as a hard copy in the Design Notebook. There is only one owner for each documentation piece. Select the piece that best fits your interests or correlates best to the design piece you have just signed up for. Note: for teams with 6 members, the final member will document the outcome of the Design Review as well as being primary on putting that PowerPoint together as well as the PowerPoint for the final presentation.

1. **Drawings**- in CAD of the device (if you have a system integration engineer he/she will deliver the assembly CAD model) Owner: Christian Gratz	**2. Electrical Schematics and electrical flow chart** Owner: Garrett Matheny
3. **Code & Algorithm** – code and code flow charts Owner: Grant Eyer	4. **Economic Analysis** – cost as well as development cost in engineering hours Owner: Joshua Hagan
5. **Project management** - Gantt Chart and analysis of final performance (calculation of reliability) Owner: Dorian Lewis	**6. Design Review** Owner: Everyone

Factors that might impact performance

This section should be used to identify personal concerns about how you might perform in the project, things that might pose a problem, course schedules, work schedules, illness, working virtually, etc.

Identifying these needs, potential obstacles, and shortcomings upfront makes it much easier for the team to identify roles and responsibilities as well as scheduling to deal with issues early rather than at crunch time.

Each team member should write out his/her concerns in this space and then as a team you can discuss how you will handle these factors should they arise this semester.

Grant Eyer: I do not work any jobs outside of school. I am available on Mondays and Tuesdays from anytime after 5:00 pm. On Wednesdays, I'm available after 12:00 pm. On Thursdays, I have ChemE car meetings, which typically last from 4:00 pm to 6:30 pm. I will be the least available on Thursdays. Unless otherwise noted, I am available all day on Fridays through Sundays. I am concerned about creating the most effective schedule possible, because what limits everyone else's abilities to work towards our objectives. A major concern I have is my anxious personality. Because a lot of my past group experiences were more negative, I am generally reluctant to trust others to carry out their roles responsibly.

Garrett Matheny: As we all take (mostly) the same classes, we should have the same issues there. One issue I have is with my job – When I am not at school, I am usually at work. That leaves me with a small time drame of almost all day Monday (10-4)-(5:30-8) and after school on Tuesdays (5-7) and Wednesdays (4-7). I can not meet at any time on Thursday, Friday, or Saturday, and I can only meet Sunday after 6:45pm.

Christian Gratz: One of my my main issues is when I'm not in school, most of the time I am working. The inclement weather has effected my work schedule although it seems to be back on track. My usual work week consists of Mondays (9am-12am), Wednesdays (9am-5pm), and Friday (9am-5pm). School days consist of Mondays (12:30pm-4:45), Tuesdays (9:30am- 5pm), and Thursday (9:30am-1:45pm). To sum that up, I'm usually available any day after 5:30pm unless class or work is cancelled.

Joshua Hagan: While I am available all during the weekday, the weekends I work all day. On Mondays I have class from 9:30 am - 12:10 pm with a little break occurring between EGR 103. EGR 103 is from 4pm - 5pm. For Tuesdays, I have class from 11am – 1:45pm. After this time, I am available till EGR 103 which occurs 3:30pm - 5:00pm. After EGR 103, I am available all day. For Wednesday, I have the same schedule as Monday, however I will not be available after 4:30 pm. For Thursdays, I have the same schedule as Tuesday, with no proceeding class. Fridays, I am available at any time. For my concerns, I worry about problems that may arise impeding normal times.

Dorian Lewis : Temporary off work Wensday, Thursday, and Friday . One class on Mondays (4pm – 4:50pm), Tuseday three class (12:30pm – 5pm) , Thursday one class (12:30pm –1:45pm).

Meetings

Rules and structure to follow

Method of decision-making

Practices for creating, approving, communicating, and keeping meeting records (template for meeting notes is provided in Canvas)

Out of Class Meetings

Time, place, notification, duration

You should plan for most of your meetings to be virtual!

Meetings will usually occur at 5:30 PM on Wednesdays. The location can be either in-person or virtual. For virtual meetings, Zoom is the preferred platform. If an in-person meeting occurs, it will generally be at the UK-Paducah Campus, unless otherwise agreed upon by the team. Meeting times can vary, but are usually between 45 minutes and 2 hours. Meetings will be held as scheduled so long as at least three of the five members will be able to attend.

Communication: Problems could arise if there is not a good understanding of what happens when a team member can't make a meeting, or how a meeting gets scheduled or cancelled at the last minute. Agree upon the expectations for these events. Does the team member need to contact everyone or just the team lead? How much notice is expected? What is the plan for "getting caught up"?

Our main source of communication will be the group chat. If for whatever reason a group member is unable to participate in our meeting, they must say so in the group chat no later than two days before the meeting. Group meetings will occur if at least three of the five group members can attend; expectations of each member will be listed in the group chat or on zoom meeting recordings. If an individual contracts COVID-19, they will still be expected to attend virtually through zoom unless they provide a valid reason for their inability to attend; they must let the group chat know as soon as they get tested. Moreover, such individuals will be expected to continue their work to the best of their ability. The other members of the team will help assist if the sick member demonstrates a relative degree of commitment to the project.

Handling Impasses

There will be conflict. This is a direct result of working with diverse and different people. In many cases, conflict can be resolved using good discussion techniques which we will address in future courses. In some cases however, there are differences that cannot be addressed quickly or efficiently. Use this section to identify how the team will address these differences. You can highlight the method that fits with what your team decides, or write your own! Some options are:

Method	**Advantages**	**Disadvantages**
Consensus: discuss the issue until everyone comes to an agreement	The team will reach the best possible compromise	The process can be extremely time consuming The consensus may be "fake"--someone may give in just so the team can move on.
Majority Rules: Vote and adopt the majority decision	• The impasse is resolved quickly • The solution does not involve anyone outside of the team	• Some team members may feel left out or that others are "ganging up" on them.

Third Party Decides: Present both sides anonymously to a classmate/client/other third party and let them decide.	• Feelings are less likely to be hurt.	• It may take time to contact a third party and present the information.

Approval and Acceptance of Contract

This contract is subject to the terms and conditions stated herein. By affixing signatures below, the parties verify that they are authorized to enter into this agreement and that they accept and consent to be bound by the terms and conditions stated herein.

Signature Team Member 1	Date
Grant Eyer	2/21/2021
Printed Name Grant Eyer	

Signature Team Member 2	Date
Christian Gratz	2/17/21
Printed Name Christian Gratz	

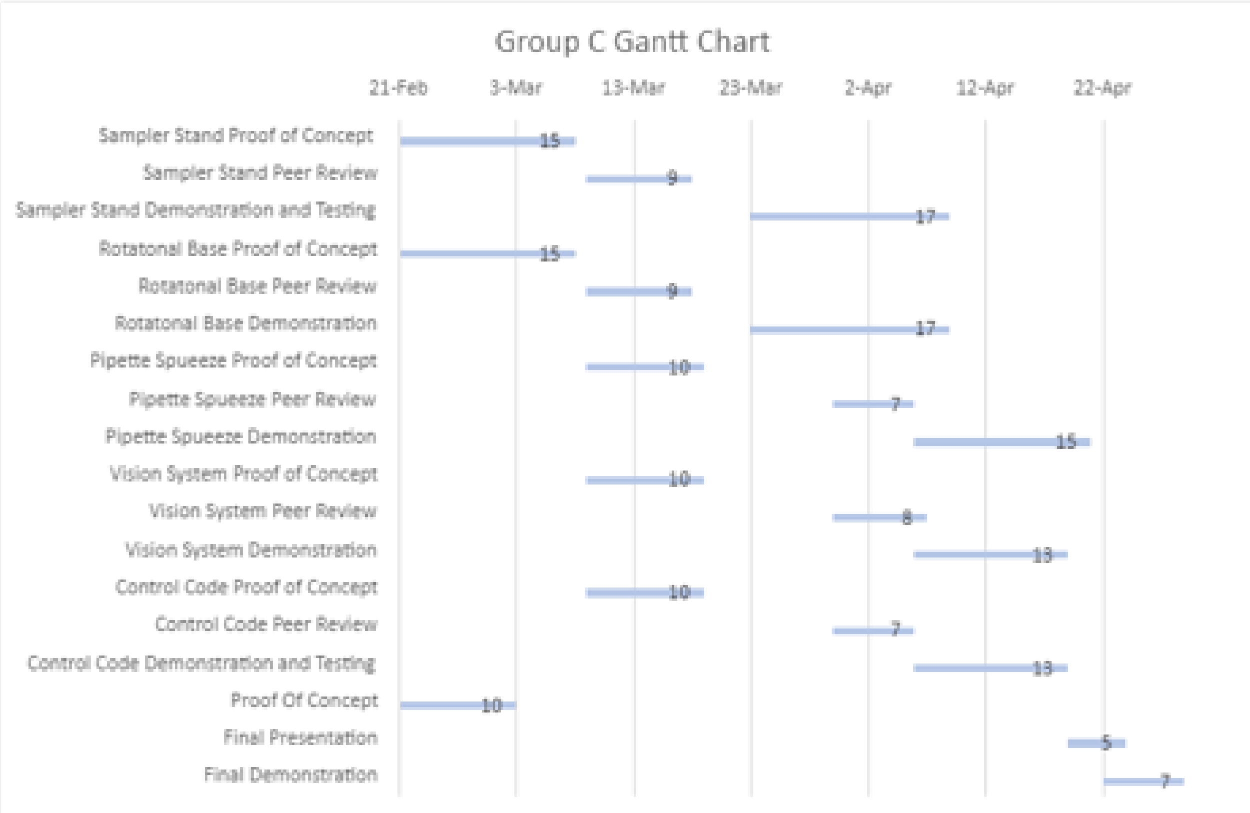
Signature Team Member 3	Date
Dorian Lewis	02/17/21
Printed Name Dorian Lewis	

Signature Team Member 4	Date
Garrett Matheny	02/17/21
Printed Name	

Garrett Matheny	

Signature Team Member 5 Joshua Hagan	Date February 17, 2021
Printed Name Joshua Hagan	

Gantt Chart


Group C Gantt Chart
21-Feb
3-Mar
13-Mar
23-Mar
2-Apr
12-Apr
22-Apr
Sampler Stand Proof of Concept
15
Sampler Stand Peer Review
9
Sampler Stand Demonstration and Testing
17
Rotatonal Base Proof of Concept
15
Rotatonal Base Peer Review
9
Rotatonal Base Demonstration
17
Pipette Spueeze Proof of Concept
10
Pipette Spueeze Peer Review
7
Pipette Spueeze Demonstration
15
Vision System Proof of Concept
10
Vision System Peer Review
8
Vision System Demonstration
13
Control Code Proof of Concept
10
Control Code Peer Review
7
Control Code Demonstration and Testing
13
Proof Of Concept
10
Final Presentation
5
Final Demonstration
7

Meeting Notes

Date: __February 26,2021_________________________

Attendees:

Joshua	
Grant	
Dorian	

Our Objective

Talk about individual sampler stand ideas, rate each design ideas. Reach conclusion of best idea sampler stand.

Important Deadlines

Drawing must be summited in Overall-view POC by Monday.

Project Update

Going with Hydraulic Rotational Base idea for sampler stand.

Roadblocks

Was unable to speak with Garrett and Christian about their design today

Next Steps

Complete this chart so that everyone is aware of what they are doing and have initialed agreement to it.

Task	Completion Date	Person to complete	Initials

Wire stepping motor, Submit individual design work on rotational base	Mar.2,2021	Christian	C.G
Submit individual design , Submit Stepping motor	Mar.3,2021	Garrett	G.M
Record zoom video on Youtube	Feb 27. 2021	Grant	G.E
Record meeting on Meeting note, test Hydraulic system, work on Sampler stand POC	Mar. 1 2021	Dorian	D.L
Finish, Design Matrix, submit Overall POC	Mar.16 2021	Joshua	J.H

Signatures of Attendees

Dorian Lewis Dorian Lewis

(print name) (signature)

_Grant Eyer___________________________________ **___*Grant Eyer*________________________**

(print name) (signature)

__Joshua Hagan___________________________________ _*Joshua Hagan*_________________

(print name) (signature)

__Garrett Matheny_____________________________ *Garrett Matheny*____________________

(print name) (signature)

___Christian Gratz______________ ______________*Christian Gratz*____________________

(print name) (signature)

__________________________________ ________________________________

(print name) (signature)

Meeting Notes

Date: _March 3, 2021______________________

Attendees:

Grant Eyer	
Christian Gratz	
Garrett Matheny	

Our Objective

During this meeting, we discussed general dimensions for the overall design, and we further discussed the rotating base design and its mechanisms.

Important Deadlines

Sampler Stand POC and Rotational Base POC is due March 8, 2021 (Dorian and Christian).

Pipette Squeeze POC, Vision System POC, and Control Code POC are due on March 19,2021 (Joshua, Garrett, and Grant).

Project Update

Christian Gratz: Have generated several designs for my rotational base, just need to do some calculations, dimensioning. Need to fill out rotational base POC

Garrett Matheny: Currently investigating different colored backgrounds for vision system.

Grant Eyer: I have generated a basic graphical user interface for design. I will be waiting on the code for my other members to come in.

Dorian Lewis: I have generated a design for the sampler stand; I am looking for the proper materials to use for the stand and need to perform center of mass calculations.

Joshua Hagan: I have begun looking into designing the pipette squeezer.

Roadblocks

Christian had problems with understanding exactly how to line up the rotational base with the sampler stand. He needs to hear from Dorian in this regard, as Dorian is mainly responsible for sampler stand. The rest of the team will look over the designs and calculations for the rotational base and sampler stand to ensure the designs make sense.

Next Steps

Complete this chart so that everyone is aware of what they are doing and have initialed agreement to it.

Task	**Completion Date**	**Person to complete**	**Initials**
Add hours to Cost Tracking spreadsheet as each individual works on subsystem.	No strict due date, just fill out as project progresses	Everybody	
Create decision matrix for pipette squeezer, and think about ideas for pipette squeezing mechanism.	March 10,2021	Joshua	J.H.
Turn in Sampler Stand POC	March 7, 2021	Dorian	D.L
Turn in Rotational Base POC	March 7, 2021	Christian	C.G.
Vision system background and camera view need to be found. Think about design matrix criteria	March 10,2021	Garrett	G.M.

Think about control code POC, generate design matrix for control code.	March 10,2021	Grant	G.E

Signatures of Attendees

_Grant Eyer___________________________________ **____*Grant Eyer*________**

(print name) (signature)

_Garrett Matheny__________________________ ____*Garrett Matheny*____________

(print name) (signature)

Dorian Lewis Dorian Lewis

(print name) (signature)

Joshua Hagan ***Joshua Hagan***

(print name) (signature)

___**Christian**___**Gratz**____________________ ______***Christian Gratz***____

(print name) (signature)

____________________________________ __________________________________

(print name) (signature)

Meeting Notes

Date: 3/10/2021

Attendees:

Dorian Lewis	Garrett Matheny
Christian Gratz	Grant Eyer
Josh Hagan	

Our Objective

Further consider potential designs for the rotational base and sampler stand while also starting the initial design for the vision system, pipette squeeze, and control code.

Important Deadlines

Sampler Stand Peer review: March 18

Rotational Base Peer review: March 18

Pipette Squeeze POC: March 19

Vision System POC: March 19

Control Code POC: March 19

Projct Update

Grant: Determining the best design for the gui code and is refining and breaking down ideas to ensure the correct design is chosen. Working on POC for Control Code.

Christian: Working with Dorian to determine the best option for meshing the rotational base and sampler stand

Garrett: Determining the best option for mounting the vision system and working on POC for vison system.

Dorian: Working with Christian to determine best option for meshing the rotating base with sampler stand and will be working on Sampler stand peer review. e

Joshua: Working on POC for pipette squeeze.

Roadblocks

Garrett came in to a road block with determining the best way to model the decision matrix for choosing the colors for the background of the vison system. We used prior decision matrix's to as a template.

Next Steps

Task	Completion Date	Person(s) to complete	Initials
Finish peer reviews for sampler stand and rotational base	March 18	Christian and Dorian	C.G D.L
Finish Proof of Concept for pipette squeezer	March 19	Joshua Hagan	J.H
Finish Proof of Concept for vision system	March 19	Garrett Matheny	G.M
Finish Proof of Concept for vision system	March 19	Grant Eyer	G.E.

Signatures of Attendees

__Grant Eyer___________________________________ ___***Grant Eyer***_____

(print name) (signature)

___**Joshua Hagan**________________________________ _______***Joshua Hagan***____________

(print name) (signature)

___**Christian Gratz**_________________________________ ___***Christian Gratz***_____

(print name) (signature)

___**Garrett Matheny**_________________________________ _***Garret Matheny***__

(print name) (signature)

___**Dorian Lewis**_________________________________ __**Dorian Lewis**____

(print name) (signature)

____________________________________ __________________________________

(print name) (signature)

Meeting Notes

Date: 03/31/21

Attendees:

Garrett Matheny	Grant Eyer
Joshua Hagan	Dorian Lewis
Christian Gratz	

Our Objective

Give updates about sampler stand gear diameter.

Updates excel sheet with hours worked and cost of materials.

Get an idea of how to connect sampler stand and rotational base.

Double check if sampler stand height will work.

Important Deadlines

Grant, Joshua, and Garrett: Peer reviews on April 6th

Dorian and Christian: Combine sampler stand and rotational base by next Monday.

Garrett: Have a picture with pipet and sediment by next Monday.

Project Update

Grant Eyer: Has Start/Stop Design complete with loading bar.

Joshua Hagan: Has 3D model of pipet holder and squeezer

Garrett Matheny: Has control code to where it will read an image and draw lines at the sediment and pipet.

Christian Gratz: Has rotational base prototype created.

Dorian Lewis: Has sampler stand prototype created.

Roadblocks

Figuring out how to make the motor turn faster.

Next Steps

Task	Completion Date	Person to complete	Initials
Peer Reviews	April 6th	Grant, Joshua, Garrett	GE, JH, GM
Connect Sampler Stand and Rotational Base	April 5th	Dorian and Christian	DL, CG
Have a picture of the pipet and sediment	April 6th	Garrett	GM
Figure out sampler stand gear diameter	April 6th	Dorian	DL

Signatures of Attendees

Garrett Matheny ***Garrett Matheny***

(print name) (signature)

Grant Eyer ***Grant Eyer***
(print name) (signature)

Joshua Hagan ***Joshua Hagan***

(print name) (signature)

Dorian Lewis ***Dorian Lewis***

(print name) (signature)

Christian Gratz ***Christian Gratz***

(print name) (signature)

Design/Build Information

Overall Design POC

EGR 103 - Engineering Exploration 2

Proof of Concept – Overall Design

The following template is designed to guide your team through the selection of an overall design concept. Into the blanks below each set of questions, type the team's responses or upload images or text from other sources to answer what it asked.

When printed out for submission in the hard copy team notebook, please have all team members initial all pages in the areas provided. For soft copy submission include team member names by concepts submitted (Conceptualization).

Define the Problem - All team members need to have read the project description.

- What are the objectives?
- What does this project need to do? What must it NOT do?
- What variables do we need to consider?
- What are the constraints that need to be considered? (size, time, cost…)

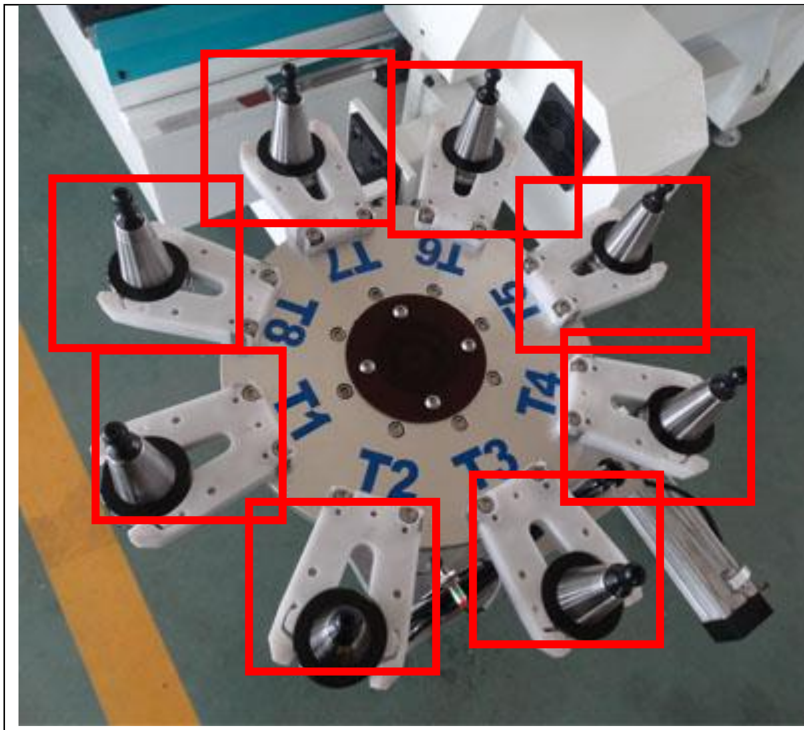
The main objective for this project is to create a vision guided water sampling system that checks for the presence of COVID-19 in sewage water. This water sampler is composed of five main parts: a sampler stand, a rotating base, a pipette sampler, a vision system, and a control code. Once the centrifuged sample and water-absorbing polymer are placed on the rotational base, the control code will cause the rotational base to position the centrifuged sample directly below the pipette. The pipette will be lowered until the pipette tip is about 1 cm above the sediment layer. The pipette sampler will acquire approximately 1 mL of the sample, and the sample stand will lift out of the solution before the rotational base moves the water-absorbing polymer under the pipette. The sampler stand will lower the pipette, the pipette sample will transfer the 1 mL of solution into the water-absorbing polymer, and then be raised by the sampler stand to its neutral location. The previously described process needs to occur in less than four minutes. All parts of this program are expected to run programmatically, though manual placement of the centrifuged sample and water-absorbing polymer is expected.   In addition, our team will only have two attempts to successfully sample the water and move it to the waiting polymer. The machine should operate without additional manual intervention.

The following need to be considered: the time that we have to design, create, and assemble the subsystems, the cost that it takes to make each subsystem and the overall machine, the size of each subsystem and overall machine, the environment that the system is operating in (e.g. light room or slightly-lit room), the variations in sediment layer height, the manner in which the samples are held in place, the torque requirement for the lift mechanism on the sampler stand, the speed provided by the motors of the rotational base, the mechanical advantage needed for the pipette sampler, the location of the camera for the vision system, the markers or locations for the vision system to properly operate, the manner in which the subsystem codes and the control code itself operates, and the appearance and functionality of the Graphical User Interface for the Control Code.

Research

- Review as a team images of similar products that you can find. Reverse engineer at least two images uploaded here. What are the creative/great/interesting parts of the design that you see in those images?
- For each observation it there a way to incorporate or apply that learning to your team's design?

spindle rotate 180 degree wood foam moulding 3d carving 4 axis cnc router machine laser engraving machine,laser cutting machine,laser marking machine,Woodworking cnc router,metal plasma cutter,cnc wood lathe machine-Jinan Wintek CNC Equipment Co.,Ltd



This image could be particularly useful for the rotational base mechanism. Although it is unlikely that our spinning base will be made of metal, the fascinating feature here are the "crab-like" place holders. These could come in handy as we need to be able to secure both the centrifuged sample and water-absorbing polymer containers.

[water sampling lab equipment - Bing images]()

The interesting feature on this product is that it has an arm that holds the water equipment in place. In addition, the holder itself can slide up and down a metal pole. These features could be useful for not only holding the pipette in place, but also for providing a simpler means of moving the pipette up and down.

aquacell-wastewater-sampler-portable-p2multiform.2-e1504682303730.png (377×494) (roycewater.com.au)

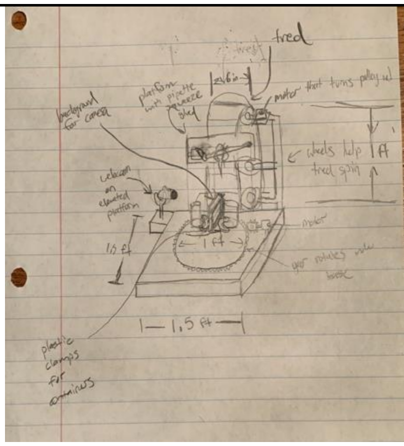
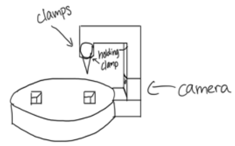
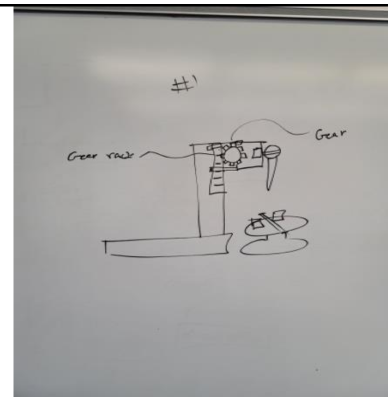
The image displayed offers a variety of characteristics. One of such characteristics is a sensor to collect or test the liquid. The same principle of sensing can be used in the water project. The difference is that a visual system would be used to direct the pipette to collect a sample of liquid.

Conceptualization – Defining possible solutions

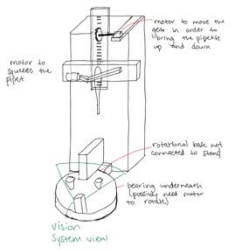
- *Upload 5-6 different design ideas (clear drawings) for the full water sampling project. Each team member should contribute a different idea for each.*
- *Each member should identify what it was about the design ideas that they chose that make it good. Compare and contrast ideas before moving to the next step.*

This will mean replacing the boxes below with images from each team member. Ideally members should come to the meeting with these already in hand. To show participation each team member should put their name on their submission. Having your name on a design doesn't mean that you can't or shouldn't dump that idea if someone else has a better one. It just means you did some background work before coming to the meeting.

Grant Eyer	Joshua Hagan	Dorian Lewis

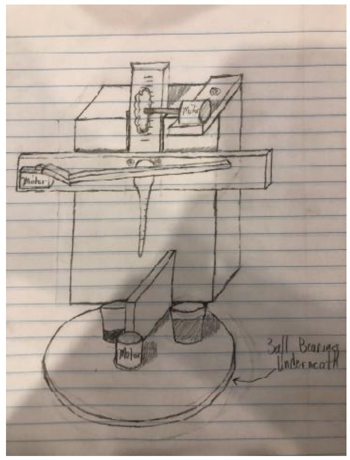
		
Justification: My design’s special traits include plastic clamps that can hold onto both the water absorbing polymer and centrifuged sample. The base is rotated by a stepping motor that is attached to a smaller gear, which rotates a larger gear underneath the base. For the sampler stand, the pipette will be held by a clamp that holds it in place; a servo motor would be used as part of the squeezing mechanism. The pipette would be moved down by a tred-like mechanism; the tred will contain three wheels, which are connected to a side stand. The top wheel will be turned by a stepping motor. The camera is positioned on the same side as the centrifuged sample to minimize obstruction of its vision. A screen would be placed behind the centrifuged sample to improve the vision system’s accuracy. Both the centrifuged samples are placed relatively close to where the pipette will be lowered so that the rotational base won’t have to work as hard to move those samples.	Justification: My Design allows for simplicity for the construction aspect. Some parts are labeled to allow for classification. I placed the location of the camera in the stand because it allows for easier reading of the liquid. Also, I implemented the design of claws to allow for not only aid the clamps in holding the pipette, but to also allow for the pipette to be squeezed. This design also allows for versatility. If change needs to be implemented, it can easily be established. Such additional implementation can be the addition of a pulley. This pully would be used to allow for a stable way of lower the pipette without failure. Another implementation could be to allow for a bearing on the bottom of the base to allow for rotation of the base.	Justification: Although my design is not very detail, it is fairly simple. It allows for additional editing. With a gear track and gear, the pipette will be able to lower in to the designated solution. The stand and pipette will have to be fasten somehow, allowing the gear to ride alone the track. The rotating base can be an and-on or separate from the stand. The visional system will have to be aligned with the pipette, maybe attach along the bottom of the mast aligned with the rotating base, or attach underneath the pipette looking down upon the rotational base. There will have to be motor attached to the gears, piston bar to squeeze the pipette, and rotational base. The rotational base will have a divider to help separate the two liquid and help guide the visional system. The dimensions can be discussed depending on big we would like our structure to be.

Christian Gratz



Justification: My design is more complicated than others and still has flaws but it has potential. My design in mostly labeled but the vision system would be set in a location on the rotational base in order to keep a set POV of the cups being occupied. Moving parts include the rotating base, the pipette, the gear(s), and the board that the pipette is attached to. My design requires 3 motors to move the pipette up and down, squeeze the pipette, and to rotate the base. An issue with the gear system is the room for error as opposed to the pulley system. The gear when moved would have to precisely and seamlessly rotate into the grooves and the gear system would be more difficult to build. Therefore, this design would be better if the pulley system was used.

Garrett Matheny



Justification: My design is more complicated than the others, but I believe it will be the most effective at achieving its purpose. The use of multiple motors allow for seamless automation without worrying about any other electronics or odd gadgets that would be complicated to build. The rotation of the bottom base, the lowering of the pipet, and the squeezing of the pipet uses these motors. The only slight issue that we could run into is building and utilizing the gear and pocket to lower and raise the squeezer. A good change to this design is utilizing a pulley system.

Conceptualization – Evaluating Your Designs

- Create a decision matrix (most teams will use Excel) with the design concepts in columns and the criteria for evaluation in rows. Design criteria can include:
  - Risk – it may be really cool looking but is it going to be hard to make it work reliably?
  - Materials – can you build it with what you are going to be given (or can you easily get the stuff you need).
  - How would the code work with this set-up?
  - If it works, will it work well or will it always be kind of marginal?
- Use this matrix to determine which design will do a better job meeting the goals based on the given criteria. The team should rate it together (you do this at the team meeting)
- Do you want to end up with a hybrid of ideas? If so draw that hybrid so all team members understand what is being considered and put that as an additional column to compare

Put a copy of the decision matrix here

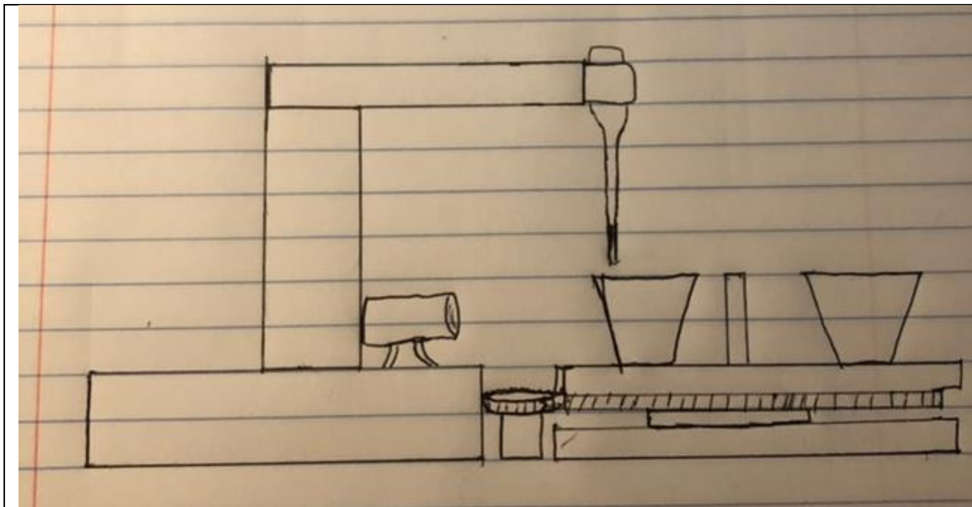
Decision Matrix for Water Tower Designs						
			Scale			
			5 - Excellent		1 - Improbable	
Specification	Weight	Design A	Design B	Design C	Design D	Design E
Time to Create	5	3	3	3	3	3
Cost for Materials	4	2	3	3	3	3
Effencicy of System	3	4	2	4	4	4
Reparibility	2	2	4	3	2	1
Portability of System	1	3	3	3	3	4
Total		42	40	47	49	48

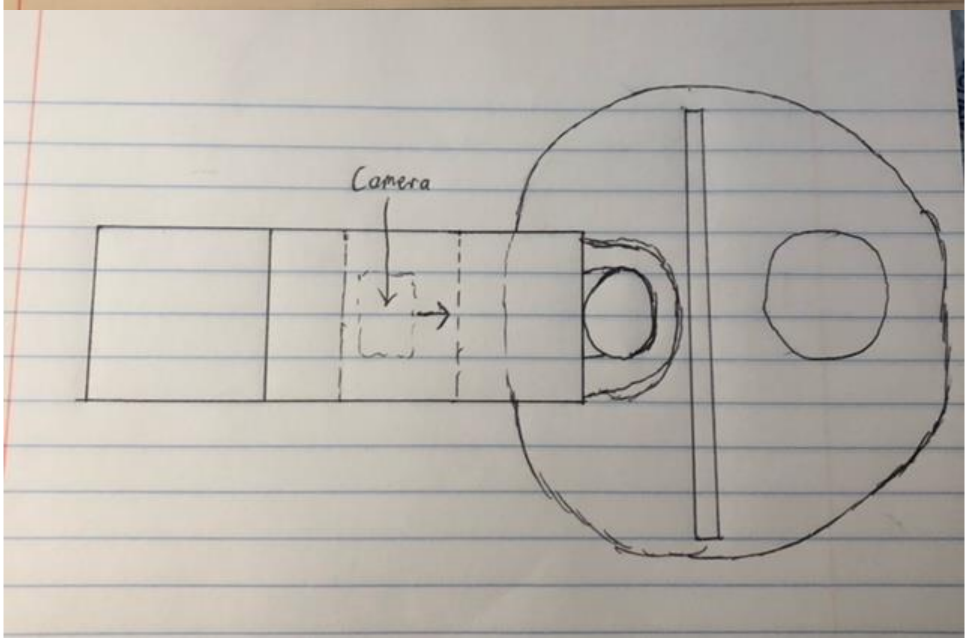
Do you have a winning design or more than one? Which one(s)?
While the totals are in the same general range, we have one official winning design. This winning design is Design E, which belongs to Christian Gratz.

Synthesis - First Cut of your design
Now that you have done this work, the team needs to converge on a final design concept. There are many ways for a team to decide and the matrix above, while helpful, doesn't make the final decision.

Draw an image below of the final design concept. Top and front or side view to show design. Make it a nice drawing as the better job you do the more likely you are to flush out critical parts of the design you hadn't thought of. CAD general concepts are great if you want to make a super-simple solid model. All parts including scoring system need to be included. Code for now is not on the drawing but can be part of the discussion.

- Think about the materials you have to solve this problem:
    - List below the materials you have to work with, star any the team feel are particularly interesting
- What are going to be the most difficult pieces of the total design to get working and why? Identify them as they will demand your attention and energy first. List them below or circle them on your drawing.





- Wood
- Gears
- Motors
- Pipet
- Camera
- Cups
- Socket
- Ball Bearing

The most difficult part of this design to build will be the contraption to squeeze the pipet and the organization of gears below the rotational base. Otherwise, the rest of the design should cause no real problems.

Evaluation – Design • Are there other considerations I need to address? What are they? • Did I identify the critical design parameters?
Several other factors to consider include but aren't limited to the following: the design of the gears themselves, the number of gears, the orientation of those gears, the center of mass of the sampler stand, how to reduce the obstruction of the vision system, how easy it is to access each subsystem (for repair purposes), the length of the lever arm for the servo motor, and the locations and mechanisms that hold each motor in place.

Sampler Stand Proof of Concept

Proof of Concept – Sub-Component Proof of Concept

Define the Problem • What are the objectives for this sub-component? • What does this subassembly need to do? What must it not do? • What variables do we need to consider? • What is already decided based on the team's overall concept? • What questions do you need to answer? (Example: do you know approximately how much weight you will need to support, how much friction you have, where the webcam will need to mount?)
The objectives of the Sampler Stand are to successfully construct a stand capable of holding a pipette squeezer and ensuring that the system remains steady. The subassembly needs to be able to function with other parts of the device such as the squeezing the pipette and able to aid rotating the rotational base. The variables that we need to consider on this sub-component is the calculations of the center of mass and the calculations of torque. We decided that the Sampler Stand should be capable of withstand the mass of the pipette holder and the pulley mechanism that will be used to lower the pipette. A few questions include how much weight the sampler stand needs to be to maintain the position of the pipette squeezer, will the piece be able to successfully connect with other sub-systems of the project, and will the camera be able to function where it is placed on the drawings?

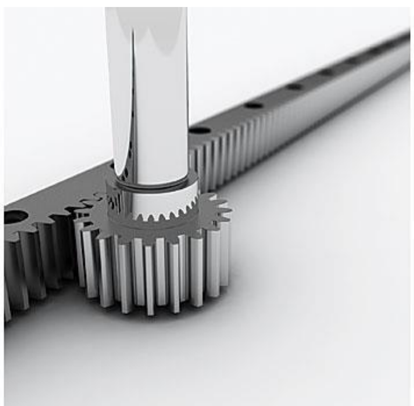
Research • Reverse Engineer- go online and find 4 useful images for a device that is similar to yours. Under each one includes information as to what you think is interesting (good OR bad) about that design.

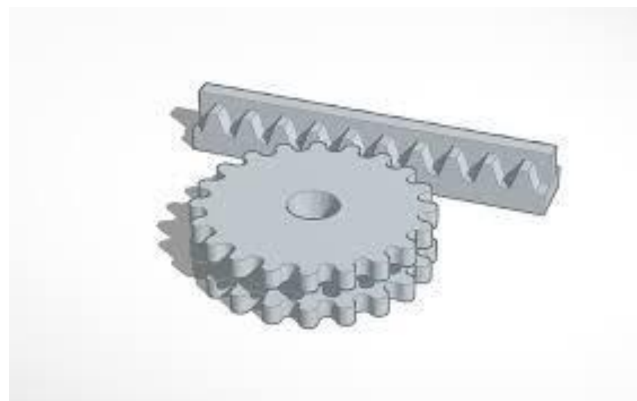
This design of a L shaped wood piece is similar to the main design that was established in the Overall Proof of Concept. The image is good; however, the image is it does not contain a piece that extrudes for a place to hold the camera.

The shaping of this design is similar to the one that was designed as well, but it is not as functional as the original. This is because the piece that is displayed does not have a stand that will be capable of holding the visual system.

The picture displayed above shows the pulley system as well as a similar way of holding the pulley. However, the piece will be hovering vertically and not at an angle.

The image above shows the same usage of a pulley as the picture above. This image, however, provides the horizontal parallel of the top part of the crane to the ground. This parallel is more accurate than the image above due to the similarity of the original design in the Overall Proof of Concept.

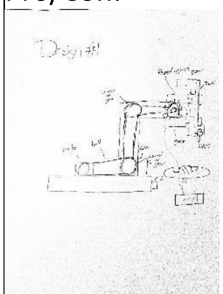
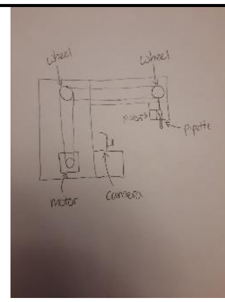
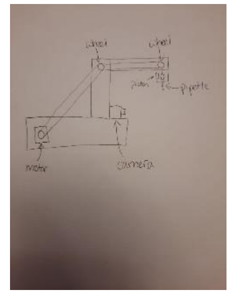
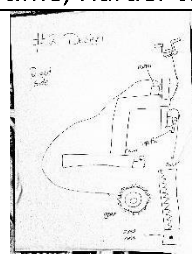




This images above depicts the mechanical lift idea for the operation of the pipette. This will be fastened to the arm of the structure.

Conceptualization – Defining possible solutions

- *Draw 4-6 different design ideas for the subcomponent. Place those below. These should be your best design concepts, but some can be minor variations of others. You want at least 3 really different ideas to compare.*
- *Under each list the pros and cons as you see them*

	Pro/Con:  Pro: May be more efficient / Con: Cost for material, time, Harder to fix	 Pro/Con: Pro: Have the possibility of functioning by the cheapest means possible / Con: The stand may not be able to stay standing	 Pro/Con: Pro: Will be able to stand, less cost, simplicity / Con: Simplicity of the design can lead to broad problems	
	 Pro/Con: Pro: Little cost, easy fix, less time to build / Con: Time to print 3D part	Image of Design E Pro/Con:	Image of Design F Pro/Con:	

Conceptualization – Evaluating Your Designs in a Decision Matrix

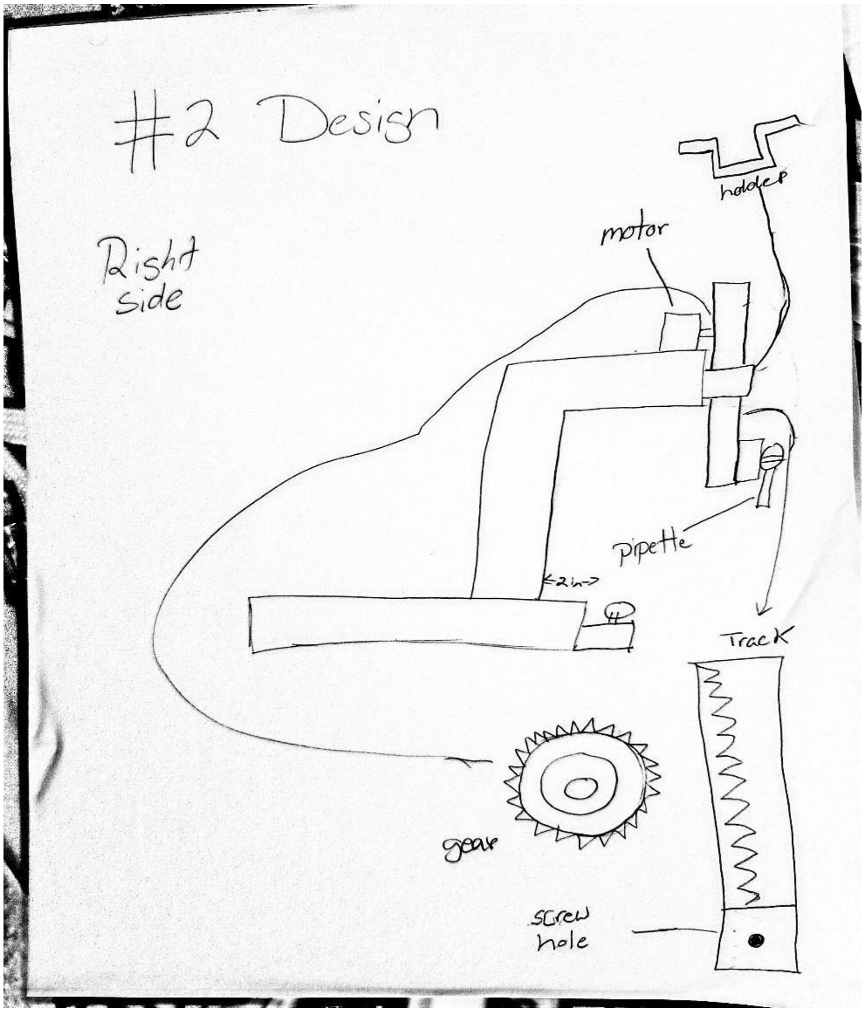
- Review the decision criteria with your team before beginning.
- Which design will do a better job meeting the goals based on the given criteria?
- Be sure your team thinks through the following: risk, time required, materials, cost, system interfaces
- End the meeting with a decision on the design to pursue, or with a date to come back with additional information so the decision can be finalized.
- Load your team's decision matrix in the space below and highlight the design you will move forward with. Under the decision matrix include a justification for your final selection.

Decision Matrix for Water Tower Designs

		Scale				
		5 - Excellent			1 - Improbable	
Specification	Weight	Design A	Design B	Design C	Design D	Design E
Time to Create	5	3	4	3	3	0
Cost for Materials	4	3	3	3	3	0
Efficiency of System	3	5	2	4	5	0
Repairability	2	2	3	3	3	0
Portability of System	1	3	3	4	3	0
Total		49	47	49	51	0

Justification for final selection: Design D was the final selection according to the specifications that were established in the Overall Proof of Concept generation. The selection of it was due to its functionality in every category. Design D offered exceptional efficiency following the goal, which is keeping the system stable and being able to withstand the other components of the project. The cost of the materials was approximately around the range of the other designs. In terms of Repairability, it will be able to be repaired or adjusted with little to no issues. Finally, the Portability was also approximately around the same range as the other designs. Overall, Design D offered a variety of positive aspects, which allowed for the greatest chance of success.

Synthesis - First Cut of your design • Paste a drawing (hand drawing is acceptable as long as it is a quality drawing!). • Mark the drawing or make a list below the image of what materials will be used to make the components • Put in a one to two sentence description of what you think will be the most difficult / highest risk areas. • Think about what you will need to do to make this a functioning design. You will need to build it (what will that take?). You will need to test it (how, and how will you evaluate its performance?). Then you will need to integrate it into the team design. Now think about how long each of these steps will take you, given you have lots of things you need to do in your life. Come up with a general timeline. List those key steps and associated time required. Include a picture of the subcomponent and the evaluation process you will follow.
Picture of subcomponent:



The track will probably the hardest part of the sub-system to illustrate in the image. Also, being able to have the gear function challenging.

#	Part	Material	Size (L x W x H)
1	Base	Wood	20" x 11" x 2”
2	Mast	Wood	15” x 2” x 4”
3	Arm	Wood	5” x 4” x 4”
4	Track	May do 3D printing	N/A

	5	Gear	3D Printing	N/A

Evaluation Process:

Development		
Activity	Duration	Date to complete
Ex. Base	30min	March 17, 2021
Mast	15 mis	March 17, 2021
Arm	45mins	March 17, 2021
Track	60 mins	N/A
Gear	60 mins	N/A
Assembly	60min	N/A

Synthesis – Model

- A document has been created for you to help you through modeling some of the critical aspects of your design. Use that document to create the calculations and modeling listed. Place those calculations and results in the section below.

Dimensions

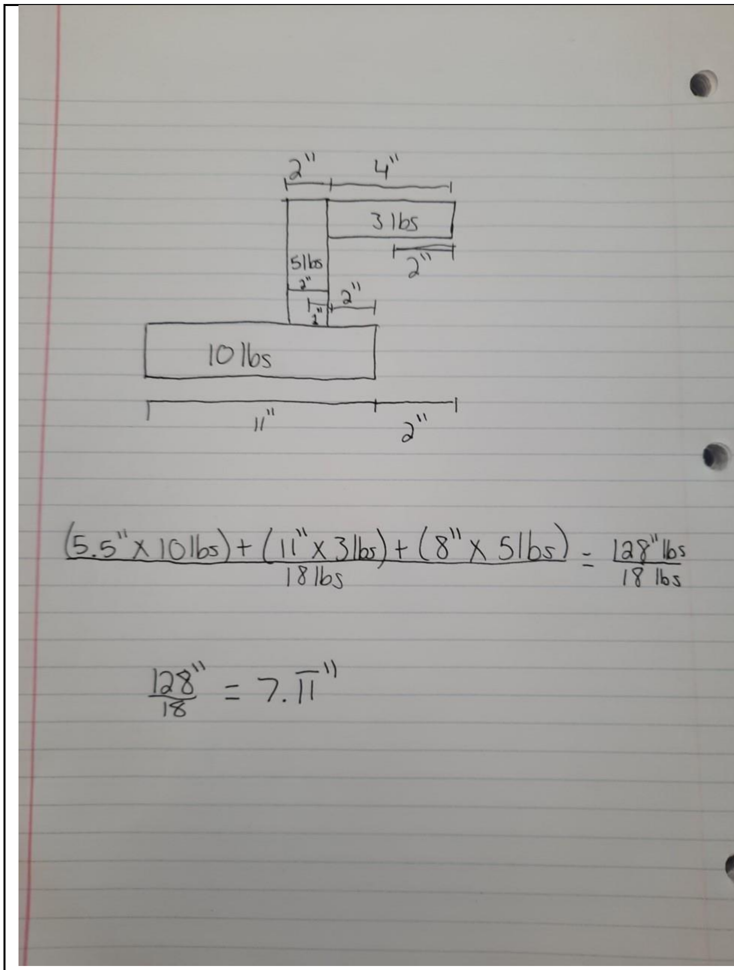
Base 20" x 11" x 2”

Mast 15” x 2” x 4”

Arm 5” x 4” x 4”

Center of mass

7.11 in.



Evaluation – Calculations

- What conclusions are you able to make from your model?

Our model will be able to withstand the weight of the additional pieces, as well as sustaining its' own weight. This allows for a smooth transition into the creation of other parts. Also, according to the data that was collected, this subsystem will be able to handle the extension of the pipette.

Evaluation – Prototype

- You will create a working example of your design concept. Take an image and place it in the space below.

Currently Not Available

D.L	03/08/2021
J.H.	03/08/2021
G.M	03/08/2021
G.E	03/08/2021
C.G	03/08/2021

Rotational Base Proof of Concept

Proof of Concept – Sub-Component Proof of Concept

Define the Problem • What are the objectives for this sub-component? • What does this subassembly need to do? What must it not do? • What variables do we need to consider? • What is already decided based on the team's overall concept? • What questions do you need to answer? (Example: do you know approximately how much weight you will need to support, how much friction you have, where the webcam will need to mount?)
The objectives for the rotational base are to provide a platform for the sampling containers to sit, have a solid background so the vision system can identify the level of the liquid, and it should spin fast enough to complete a cycle but not too fast to where the cups become to unstable and fall over. Variables that need to be considered are the number of teeth on each gear, the radius of each gear, the mechanical advantage, and radius of the base. Questions that need to be considered: How much weight the platform needs to support? How fast the platform should spin? If the mechanical advantage is exceptional What color should the background be for the vision system? The number of teeth the gears should have Based on the overall concept of the design our team has concluded that our rotational base will include 2 gears that are potentially going to be 7" in diameter, the platform made of wood should be approximately 8" in diameter, and the rotating base will possibly need to be elevated to level the gears so that they can align properly.

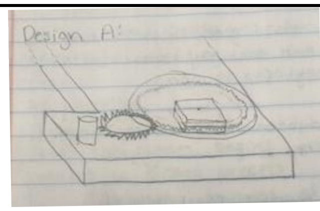
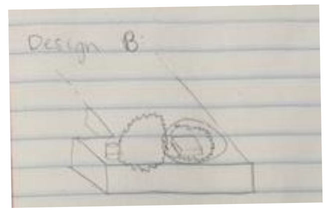
Research • Reverse Engineer- go online and find 4 useful images for a device that is similar to yours. Under each one include information as to what you think is interesting (good OR bad) about that design.

Sprinkler: a good representation in the since that certain sprinklers rotate to a certain degree then changes direction and but it does not go back to it's original place. The concern with the sprinkler is it "clicks" when rotating rather than a smooth rotation and when it changes direction of rotation it does not pause. The need for a pause in between rotations allows the pipette squeeze to descend, acquire the liquid, then ascend.

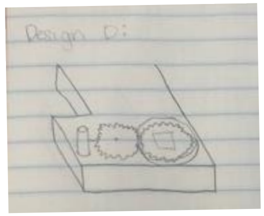
Swivel Plate: very good potential to becoming our rotation device. The bottom half of the swivel plate would be attached to the sampler stand while the top half of the swivel will be attached to the gear allowing the motor to rotate the platform.

Ball Bearing: this product also has potential. This specific bearing could be placed in the center of the gear while on the sampler stand a simple peg could ensure that the gear stays on track and this would eliminate overall weight and size.

Weed Eater: the problem with a weed eater set up is that the weed eater head spins in one direction (typically clockwise). If this concept was integrated in our design we would have to do a 180 in order to get to one cup, then another 180 to get back to starting position.

	 Pros: elevating the rotating base to ensure that the driven and driving gears match up correctly Cons: the driving gear in this case is much smaller than the driven gear making it more difficult for the motor to rotate the platform.	 Pros: this design features a cut out for the motor to sit. The driving gear is much larger than the driven gear so the base will rotate quicker. Motor is sitting in a hole in the wood to connect the gears. Cons: smaller gear underneath the base may be too fast and maintenance on the motor would be difficult given that it sunken down in a hole.	 Pros: this design does not involve a separate section for the motor to be placed saving room, cost, and overall weight. Cons: the motor would have to work harder since the motor is directly connected to the platform decreasing the mechanical advantage.	

Conceptualization – *Defining possible solutions* • *Draw 4-6 different design ideas for the subcomponent. Place those below. These should be your best design concepts, but some can be minor variations of others. You want at least 3 really different ideas to compare.* • *Under each list the pros and cons as you see them*

 Pros: this design is the most liked in our group and it consists of 2 gears the same size one gear driving the other with an elevated base to allow the gears to lock in more efficiently. Cons: the ability to determine the amount of teeth to fit all the components to fit and to achieve the greatest mechanical advantage.		

Conceptualization – Evaluating Your Designs in a Decision Matrix

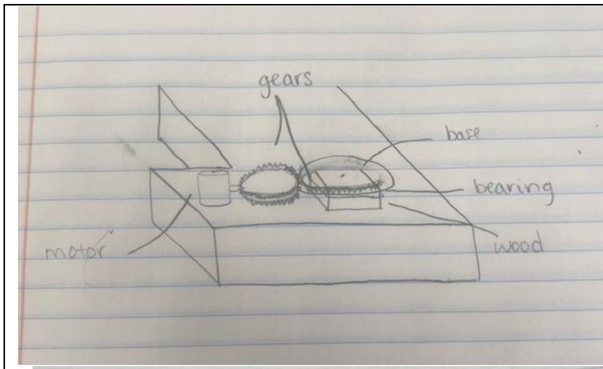
- Review the decision criteria with your team before beginning.
- Which design will do a better job meeting the goals based on the given criteria?
- Be sure your team thinks through the following: risk, time required, materials, cost, system interfaces
- End the meeting with a decision on the design to pursue, or with a date to come back with additional information so the decision can be finalized.
- Load your team's decision matrix in the space below and highlight the design you will move forward with. Under the decision matrix include a justification for your final selection.

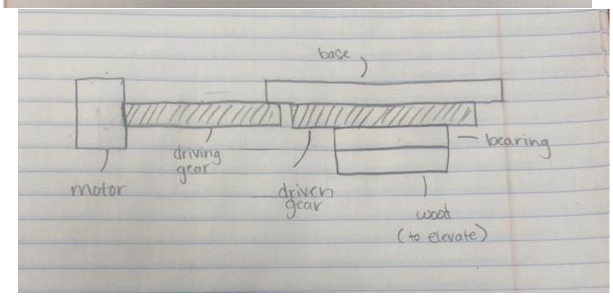
Decision Matrix for Rotational Base					
Evaluators_________.		5=high ability to meet; 1=low ability to meet			
Specifications	Weight	Design A	Design B	Design C	Design D
Low cost materials	7	4	5	5	4
Available tools	1	5	5	5	5
Enough rigidity	2	3	3	4	3
MATLAB integration	4	3	2	3	2
Easily Repairable	3	5	3	3	5
Weight and Size	5	3	3	4	4
Compatiblity	6	5	5	5	5
Effeciency	8	2	3	1	4
Total		30	29	30	32

Justification for final selection: Design D will be the design we go with. This design in the process will likely be altered to an extent but right now it's our best option based on the specifications from the decision matrix.

Synthesis - First Cut of your design

- Paste a drawing (hand drawing is acceptable as long as it is a quality drawing!).
- Mark the drawing or make a list below the image of what materials will be used to make the components
- Put in a one to two sentence description of what you think will be the most difficult / highest risk areas.
- Think about what you will need to do to make this a functioning design. You will need to build it (what will that take?). You will need to test it (how, and how will you evaluate its performance?). Then you will need to integrate it into the team design. Now think about how long each of these steps will take you, given you have lots of things you need to do in your life. Come up with a general timeline. List those key steps and associated time required. Include a picture of the subcomponent and the evaluation process you will follow.





Based on previous discussions, the most difficult part about the design is making it all fit in a sizeable fashion while maximizing efficiency. Furthermore, the gear placement will be a struggle due to the precise alignment and always having 1.5-3 teeth in contact (divider on the rotational base separating the sample containers are not included in any drawings).

#	Part	Material	Size
1	Base	Wood	8in diameter
2	Bearing	Metal	4in x 4in
3	Gear 1	3D printed plastic	7in diameter
4	Gear 2	3D printed plastic	7in diameter

Development			
Step	Activity	Duration	Date to complete
1	3D Model	60min	Tuesday, March 15th
2	Create Prototype	60min	Tuesday, March 22nd
3	Finalize Design	20min	Tuesday, March 22nd
4	Final Assembly	60min	Tuesday, March 30th

Evaluation Process: 3D model will allow the team to visually see the design in a 3D atmosphere and to better understand the functionality and to see that components line up properly or if dimensions need to be altered. Creating a prototype can ensure that the functionality is flawless and to analyze any mistakes or ideas that we can incorporate before finalizing the design. Final design will then be determined allowing the next step to be assembling the final design.

Synthesis – Model

- A document has been created for you to help you through modeling some of the critical aspects of your design. Use that document to create the calculations and modeling listed. Place those calculations and results in the section below.

The stepper motor moves approximately 0.05 radians per second. Given this approximation the sampling container will be around 3in from the center of the base so we can calculate angular velocity:

(0.05 radians/sec) *(3 inches) = 0.15 inches per second

Next, we will need to determine the ratio of diameter speed.

(8 inches)/ (8 inches) = (x)(0.05 radians per second)

Given this equation, that would mean the approximate rotational velocity of the turn table would come out to be 0.05 radians per second.

Next, we will need to calculate teeth shape and size. Circumference of both gears is approximately 22in. For the gears to properly mesh the gear ratio would have to be 1:1. Due to Design D having the driven and driving gears both with the same circumference. Approximately 24 or 32 teeth should be incorporated on our gears. The shape we plan to use for our gears are shown in the picture below:

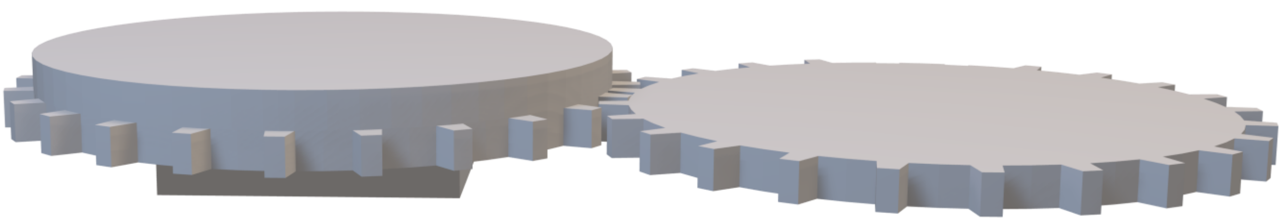
Evaluation – Calculations

- What conclusions are you able to make from your model?

Based on the calculations and overall concepts we have generated we can conclude a view thing. First, the focus needs to be how fast the rotating base will be and how long it will take to do a complete trial. We can generate a better understanding with further calculations, trial and error, and other forms of testing the velocity and angular velocity. Based off our current design the platform will not rotate fast enough to reach the 2-minute limit. Further discussion and ideas will be generated to acquire the proper speed to maximize efficiency, time, and speed.

Evaluation – Prototype

- You will create a working example of your design concept. Take an image and place it in the space below.



This is a rough concept of our design as a 3D Model. The gear on the right will be attached to the motor and the gear on the left is linked with the other gear. The gear on the left has the rotational base (on top) and the swivel (on bottom). After 3D modeling a further flaw came to mind. The gear on the right has no way of staying level with the other gear and it will need a swivel to be able to spin also.

Pipette Squeezer Proof of Concept

College of Engineering

Proof of Concept – Sub-Component Proof of Concept

Define the Problem • What are the objectives for this sub-component? • What does this subassembly need to do? What must it not do? • What variables do we need to consider? • What is already decided based on the team's overall concept? • What questions do you need to answer? (Example: do you know approximately how much weight you will need to support, how much friction you have, where the webcam will need to mount?)
The objectives for this component are to be able to move and squeeze the pipette. Once the pipette is lowered through pulley system, the Arduino should be capable of squeezing the pipette to collect water. After the water is collected it should raise up again. After being raised, the stand will rotate. Finally, the water will be dispensed into the other cup to allow for inspection. This subassembly should not let pressure that the generated by the Arduino lever dissipate. Variables that need to be considered include is time that it takes for the pulley to raise and lower the pipette system, the amount of force that needs to be applied to allow for the pipette to be squeezed, and the probability that water might leak out of the system while it is being squeezed. In the team's overall concept, it has been decided where the pipette will be positioned. It will be position vertically above the cups that the samples will be in. Questions that need to be answered include how much torque is required to allow for the pipette to remain squeezed, how much time will be needed to complete the action of squeezing, how much time is necessary to raise and lower the pipette, and how much distance the pipette will need to be lowered?

Research

- Reverse Engineer- go online and find 4 useful images for a device that is similar to yours. Under each one include information as to what you think is interesting (good OR bad) about that design.

Good: The squeezing mechanism that is used in this image is similar to the one that is going to be used in the pipette squeezer component. Bad: The application of this image is different in comparison to the overall application of the pipette squeezer.

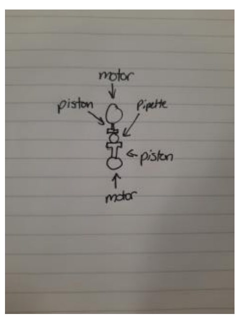
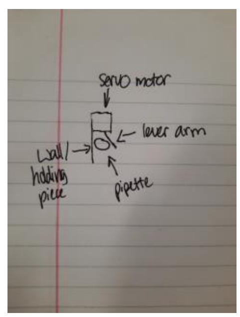
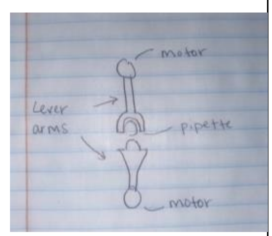
Good: The squeezing mechanism is also similar and is at an elevated operation position. Bad: The application is also widely different compared to the application of the pipette squeezer.

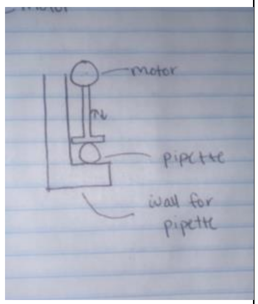
Good: The rotating mechanism used in this image is most applicable to the pipette squeezer that will be used in the over design. Bad: The image requires manual use to operate the system.

Good: The squeezer mechanism is similar to the pipette squeezer. Bad: The image also requires manual use to be operated.

Conceptualization – *Defining possible solutions*

- *Draw 4-6 different design ideas for the subcomponent. Place those below. These should be your best design concepts, but some can be minor variations of others. You want at least 3 really different ideas to compare.*
- *Under each list the pros and cons as you see them*

 Pro/Con: Pro: This design will exhibit maximum efficiency Con: The number of parts required is substantial, the building of this design is unlikely, and the overall design appears expensive	 Pro/Con: Pro: Minimal number of products with the amount of efficiency, expenses are relatively low, and is simple. Con: The simplicity may supply a problem with the calculations.	 Pro/Con: Pro: This design will squeeze the pipette the best. Con: The use of a “male” and “female” lever arms would have to be precise in order to match up and there are other designs that require one motor instead of 2.

	 Pro/Con: Maximizes the efficiency while also taking into consideration the cost. Only utilizes one motor but the job gets done. Con: It appears to be unbalanced and possibly difficult to construct.	Image of Design E Pro/Con:	Image of Design F Pro/Con:	

Conceptualization – Evaluating Your Designs in a Decision Matrix

- Review the decision criteria with your team before beginning.
- Which design will do a better job meeting the goals based on the given criteria?
- Be sure your team thinks through the following: risk, time required, materials, cost, system interfaces
- End the meeting with a decision on the design to pursue, or with a date to come back with additional information so the decision can be finalized.

- Load your team's decision matrix in the space below and highlight the design you will move forward with. Under the decision matrix include a justification for your final selection.

Decision Matrix for Water Tower Designs

		Scale				
		5 - Excellent		1 - Improbable		
Specification	Weight	Design A	Design B	Design C	Design D	Design E
Time to Create	5	2	5	4	3	0
Cost for Materials	4	2	4	3	4	0
Efficiency of System	3	5	4	5	5	0
Repairability	2	3	3	3	3	0
Portability of System	1	3	4	4	3	0
Total		42	63	57	55	0

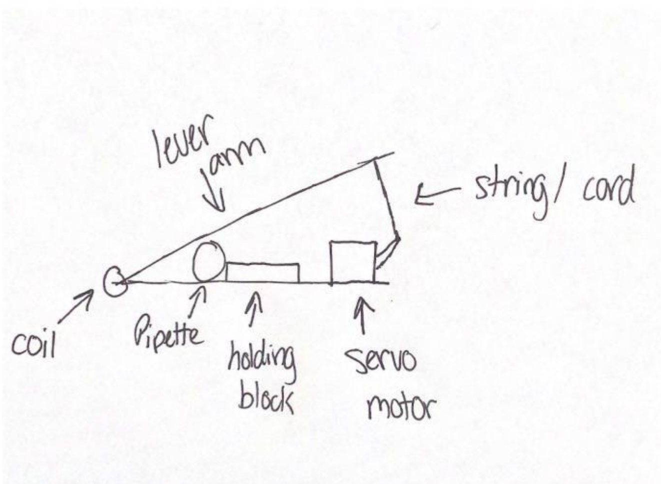
Justification for final selection:
The final selection was justified due to its high ability to meet the specified requirements established by the group member of the project. It displays the quality of being cheap, while being one of the most efficient designs. Design B offered the best options when it came to the category of time to create of the system. This is due to its simplicity of the design. In addition, it was the matched with the highest number of Cost for the materials with Design D. Finally, it offered the ability to be portable and ease to combine to other sub-components of the project.

Synthesis - First Cut of your design

- Paste a drawing (hand drawing is acceptable as long as it is a quality drawing!).
- Mark the drawing or make a list below the image of what materials will be used to make the components

- Put in a one to two sentence description of what you think will be the most difficult / highest risk areas.
- Think about what you will need to do to make this a functioning design. You will need to build it (what will that take?). You will need to test it (how, and how will you evaluate its performance?). Then you will need to integrate it into the team design. Now think about how long each of these steps will take you, given you have lots of things you need to do in your life. Come up with a general timeline. List those key steps and associated time required. Include a picture of the subcomponent and the evaluation process you will follow.

Picture of subcomponent:



For the difficult/high risk areas, there are 2 main concerns. These 2 concerns include the servo motor not having enough torque to press the pipette and how easy it will be to take off the clamps that is holding the pipette. The areas of risk can be addressed and modified to allow for an easier set up and allow for easier repairment.

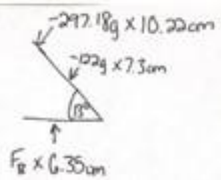
#	Part	Material	Size
1	Servo Motor	Plastic	1" x 1”
2	Pipette	Plastic	12” x 1/2”
3	Coil	Plastic or Metal	2” thick
4	Lever arm	wood	5” x 2”
5	Holding block	wood	1” x 1.5”
6	String/ cord	Nylon or other material	About 2” long

Development			
Step	Activity	Duration	Date to complete
1	Wiring the Servo	15min	Wednesday, March 17th
2	Build the sub pieces	2 hours	Tuesday, March
3			
4			

Evaluation Process:
Building each piece in a specific time frame will allow for the team to be able to build a prototype before the day of presentation. While there are other activities that are minimal to do that are not listed in the development chart, they will need to be done to complete the project. One of the minimal activities is picking up the materials. After the materials are gathered, the individual parts will be constructed.

Synthesis – Model

- A document has been created for you to help you through modeling some of the critical aspects of your design. Use that document to create the calculations and modeling listed. Place those calculations and results in the section below.



$$0=(-297.18g\times 10.22cm)+(-122g\times 7.3cm)+(F_B\times 6.35cm)$$

$$(6.35)F_B = 3927.83$$

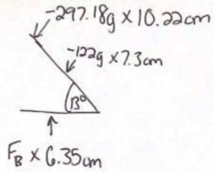
$$F_B = 618.56g$$

$$0=(F_L\times 8.76)+(-122g\times 7.3cm)+(618.56g\times 6.35cm)$$

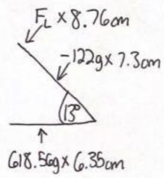
$$F_L = 346.6g$$

$$T_{motor} = (346.6g)(2cm)$$

$$T_{motor} = 693.2gcm$$



$$0=(-297.18g\times 10.22cm)+(-122g\times 7.3cm)+(F_B\times 6.35cm)$$
$$(6.35)F_B = 3927.83$$
$$F_B = 618.56g$$



$$0=(F_L\times 8.76)+(-122g\times 7.3cm)+(618.56g\times 6.35cm)$$
$$F_L = 346.6g$$
$$T_{motor} = (346.6g)(2cm)$$
$$T_{motor} = 693.2gcm$$

According to the calculations of the force, the servo motor is capable of squeezing the pipette with ease using a mechanical advantage. The servo motor that was supplied is capable of generating a torque of 1,600 g*cm, while only 693.2 g*cm is required to squeeze the pipette. Using these calculations, the volume that the pipette can absorb will be over 1 mL of water.

Evaluation – Calculations

- What conclusions are you able to make from your model?

Through the calculations that were figured in the Synthesis – Model, the pipette should be successfully squeezed and be able to absorb 1 mL of testing liquid. In addition, the model should be able to connect effectively with the Sampler Stand. After it is connected successfully, the sub-component should be able to remain in one stationary position and should be able to be squeezed by the servo motor. This model should be able to be created with ease and display minimal to no errors.

Evaluation – Prototype • You will create a working example of your design concept. Take an image and place it in the space below.
N/A

Vision System Proof of Concept

College of Engineering

EGR 103 - Engineering Exploration 2

Proof of Concept – Vison System Proof of Concept

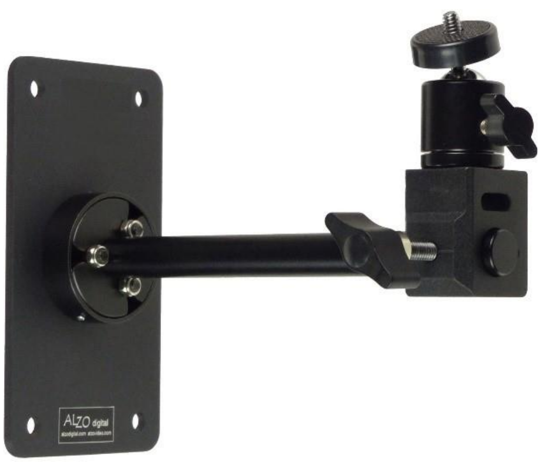
Define the Problem for the Vision System • What are the objectives for this sub-component? • What does this subassembly need to do? What must it not do? • What variables do we need to consider? (see your "Getting to POC" document) • What is already decided based on the team's overall concept? • What questions do you need to answer? (Example: what is your field of view, what will the areas of interest look like and where will they be placed, where will the webcam be mounted?)
The objective for the Vision System is to be able to effectively view the distance between the bottom of the pipet and the sediment layer and to stop the sampler stand once there is 1 cm left between the two. The assembly will utilize the camera received in the EGR 103 Kit. The camera must be able to distinguish the pipet tip and sediment layer from its surroundings, so we need to choose a good background color to assist. There are a few important variables to consider. First, we need to worry about how the code will connect with the functions of the camera. We also need to look into how wide the camera angle is and whether or not it will affect the accuracy. Based on our current concept, we have decided to place the camera on the sampler stand, underneath where the pipet will lower. There are a few primary questions of interest: • What is the best color for the background so that the camera can distinguish the pipet and sediment layer from the rest of the picture? • What is the FOV for the camera? • What should the height be for the camera? • How should the camera be mounted? • Where is the best place to put the camera?

Research

- Reverse Engineer- go out online and find 4 useful images for camera mounts. What are the good and bad aspects of these?
- Make sure that you can connect your webcam to your computer and control it through MATLAB. The "Getting to POC" document will help you through this.

Pros: Small, good mount shape
Cons: No support, not very adjustable

Pros: Very Adjustable, Good support, Good mount
Cons: Large, Expensive



Pros: Very good support, Good mount
Cons: Not too adjustable

Pros: Very Small, Fairly adjustable
Cons: Method to attach will be tricky, Mount is not great

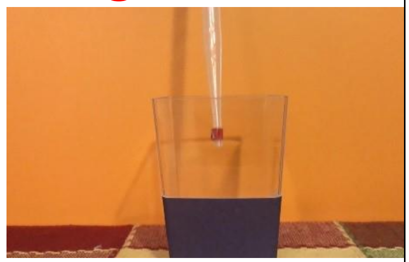
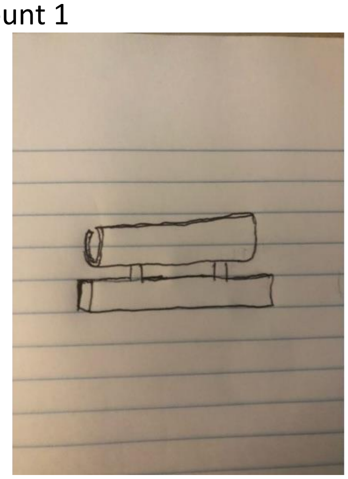
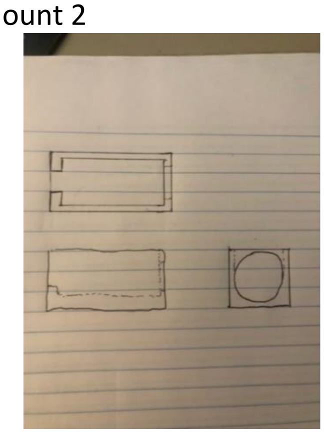
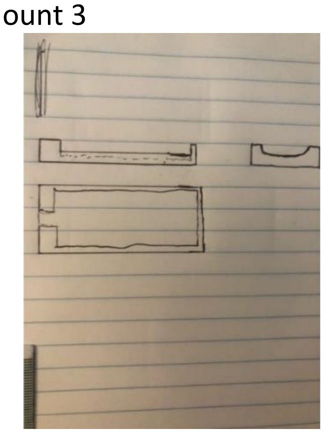
Conceptualization – Defining possible solutions

Decision Matrix A

- *Take 3 images of the sample cup with black/dark material at the bottom and a piece of colored tape on the bottom of the pipette. Try different backgrounds behind the cup and different colors on the end of the pipette. Save them as .jpeg documents and place them below.*
- *For each, try some initial image processing and compare the pixel RGB values for the areas of interest vs. the background. Under each image record the area of interest RGB values and background RGB values.*

Decision Matrix B

- *Put 3 images below of design concepts for mounting the camera and camera placement on the total system.*
- *At your next team meeting review these results with your team for feedback. The feedback will be recorded in the Decision Matrix in the next section.*

Image A  Justification: A white background is the first color I thought of when we said we needed a neutral background. Both the black and red stand out very well.	Image B  Justification: I chose to try an orange background because I wanted to use a bright color that was not white. While the sediment layer stands out extremely well, the pipet does not stand out as well as it could.	Image C  Justification: I chose to use a blue background for the final selection because I wanted a dark color to contrast the red tape on the pipet. Unfortunately, it seems that a darker color does not go well with the sediment layer as I expected.
Mount 1  Justification: This design is a bit difficult to create, but I wanted a design that would allow the camera to be easily adjusted inside of its mount. This one allows for the user to adjust it forawrd and backward according to how far away they need it from the sample.	Mount 2  Justification: This design is made to be a pocket for the camera to sit inside of, where the camera will not be allowed to move at all. This means that the distance from the sample will be consistent, but obviously it can not be adjusted without moving the entire mount.	Mount 3  Justification: This design is very similar to mount 2, except that it does not have the high walls. This design allows for simpler placement and movement of the mount, although it has the same downsides as the second design.

Conceptualization – Evaluating Your Designs
Bring a blank decision matrix to the team meeting with your best ideas for the criteria against which the design should be evaluated. You will have two decision matrices. The first one will be for the visual area (pipette tip color and backdrop color). The other will be for the camera mounting strategy.

- Review the decision criteria with your team before beginning.
- Which design will do a better job meeting the goals based on the given criteria?
- Be sure your team thinks through: risk, time required, materials, cost, system interfaces
- End the meeting with a decision on the design to pursue, or with a date to come back with additional information so the decision can be finalized.
- Load your team's decision matrices in the space below.

Decision Matrix for Vision System Background & Pipet Color

Evaluators: Grant Eyer, Garrett Matheny, Dorain Lewis, Christian Gratz, and Joshua Hagan

Scale: 5 - High ability to meet, 1-Low ability to mee

Specification	Weight	Design 1	Design 2	Design 3
Color Visibility to Camera	4	3	4	3
Color Difference from Sediment	3	5	4	2
Material Cost	1	3	3	3
Enviromental Adaptability	5	4	3	2
Color Difference from Pipet	3	4	4	4
	Sum	62	58	43

Justification: Based on the results of the decision matrix, out team should use a white color for the background. White stands out from the sediment layer and the red tape on the pipet very well compared to either orange or blue. When considering the visibility in different environments, it should also not change much since it is a neutral color. While there are 3's for the material cost for each color, white printer paper may actually cost a bit less than colored cardstock.

Decision Matrix for Vision System Camera Mount

Evaluators: Grant Eyer, Garrett Matheny, Dorain Lewis, Christian Gratz, and Joshua Hagan

Scale: 5 - High ability to meet, 1-Low ability to mee

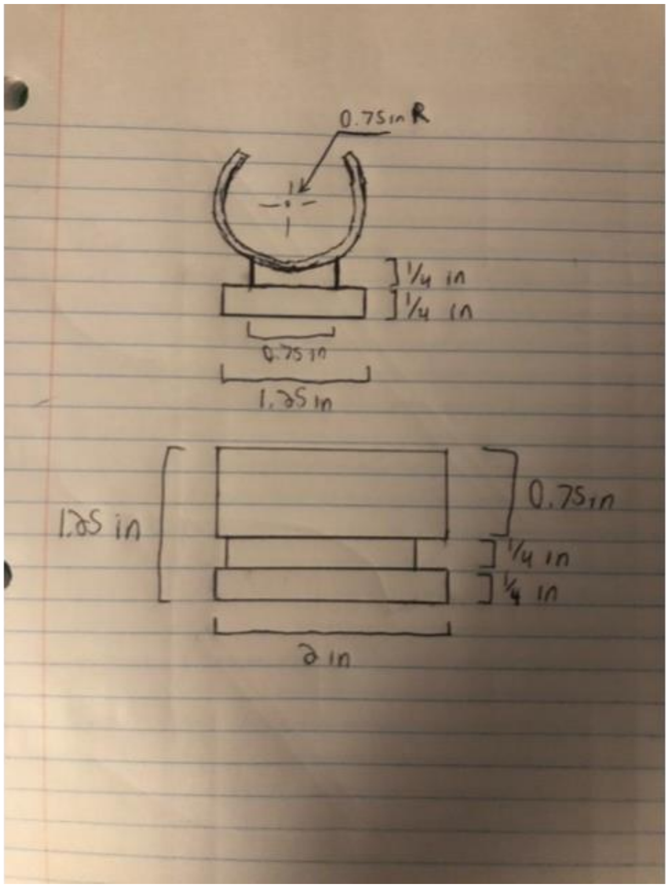
Specification	Weight	Design 1	Design 2	Design 3
Stability	4	3	4	2
Material Cost	2	4	3	4
Size	4	3	2	4
Adjustability	1	3	2	2
Complexity	3	3	2	4
	Sum	51	43	46

Justification: Based on the results of the decision matrix, design 1 will be the one we will use for the camera mount. It is stable when screwed down, has adjustability, does not cost too much, and it not too big. Its only issue is that it could prove a bit complex to build, as getting the rounded shape for the pocket to fit the camera would have to be fairly accurate.

Synthesis - First Cut of your design

- Load a dimensioned drawing of the webcam mount in the space below.
- Put in a one to two sentence description of what you think will be the most difficult / highest risk areas.

- Think about what you will need to do to make this a functioning design.
- Now think about how long each of these steps will take you, given you have lots of things you need to do in your life. Come up with a general timeline. List those key steps and time required. When your team makes a Gantt chart, use this to put into your team's schedule plan.



The most difficult area of this mount will be the design of the circular pocket to fit the camera in. In general, it will be a bit difficult to shape a few of these pieces.

The first thing will be to cut out the base, which will be fairly simple, then a support will have to be rounded out using the same radius as the pocket. The pocket will be a hollowed-out cylinder with a part of the top cut off as seen above. All of this will be held together with gorilla glue. This will probably be complete within the next two weeks.

Synthesis – Model

- A document has been created for you to help you through modeling some of the critical aspects of your design. Use that document to create the calculations and modeling listed. Place those calculations and results in the section below.

***Inches have been converted to meters for all calculations**

Horizontal Camera Angle:

Distance between Markers = 1.5 m
Distance from Camera to Markers: 2.04 m
Half Horizontal Angle = sin^-1(0.75m/2.04m)
Half Horizontal Angle = 21.57 degrees
Full Horizontal Angle = 43.14 degrees

Vertical Camera Angle:

Distance between floor and marker: 1.02 m
Distance between camera and wall: 2 m
Full Vertical Angle = sin^-1(1.02m/2m)
Full Vertical Angle = 30.66 degrees

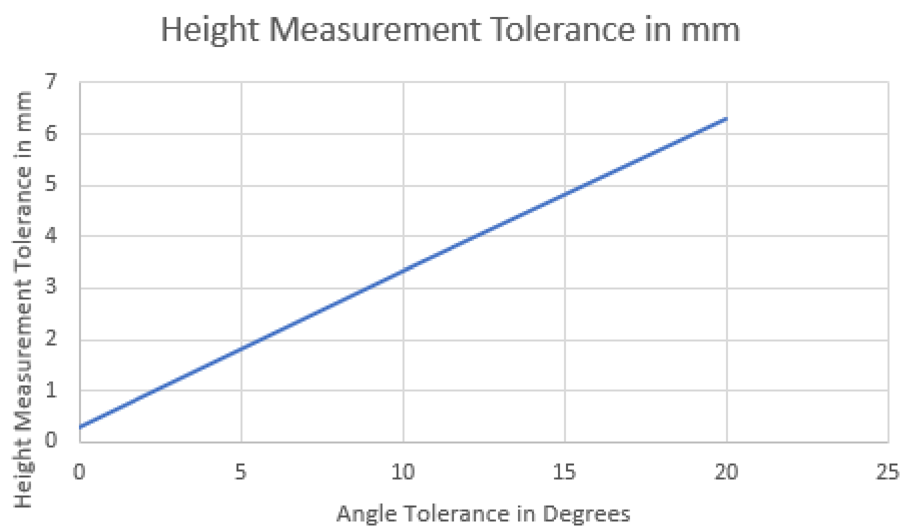
Fiducial Resolution:
Proposed distance from camera to cup: 0.127 m
Field of view width: 100.4 mm
Field of view height: 69.6 mm
Resolution of camera: 640 x 480 pixels
Pixel width and height: 0.156 mm x 0.145 mm
Pixels per mm = 6.4 pixels

Pixel and Height Tolerance:

Based on the threshold between the sediment layer and the pipet, I should make a tolerance of 1.78 mm when compensating for any tilt to a max of 5 degrees. Work shown below:

Pixel tolerance (+/- 2 pixel on each edge) = 2 pixels / 6.3 pixels/mm
Pixel Tolerance = 0.32 mm

Difference in cup lid height = (35.14 mm) * sin (5)
Difference in cup lid height = 3.06 mm
Pipet Height Tolerance = +/- 1.53 mm
Pipet Pixel Tolerance = 0.32 mm
Pipet Height & Pixel Tolerance (at 5 degrees) = 1.81mm



Evaluation – Calculations • What conclusions are you able to make from your model?
Based on the results of the model above, there is a strict angle that we need to keep our mount and cup between. Making both the mount's angle and the cup's angle both close to 0 will be a difficult task, especially considering that the sediment layer could be at varying heights. The best thing that the group can do is attempt to make the angle difference between the camera and the cup/rotating base less than 20 degrees. The group has made specific effort to make sure that the rotating base will be almost equal if not exactly equal to the top of the base of the sampler stand, so the angle should not bypass 15 degrees. One more thing that we could consider is the difference made when placing the cups in a divot on the rotating base so that they do not move. Because the divot will be so small, it should cause next to no problem, but it is still something that needs to be considered when finalizing dimensions so that it does not exceed the 20-degree angle threshold.

Evaluation – Prototype • You will create a working example of your design concept. Take an image and place it in the space below.

Control Code Proof of Concept

College of Engineering

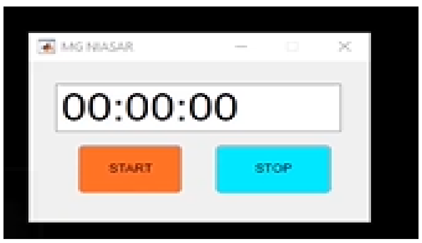
EGR 103 - Engineering Exploration 2

Proof of Concept – Control Code Proof of Concept

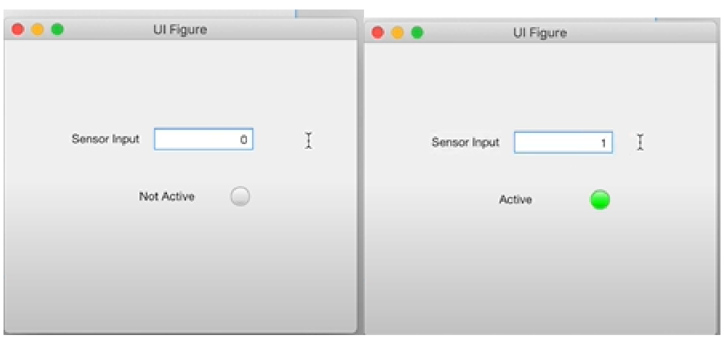
Define the Problem for the Control Code • What are the objectives for this sub-component? • What does this system need to do? What must it not do? • What variables do we need to consider? • What is already decided based on the team's overall concept? • What questions do you need to answer? (Example: What do users need to be able to control? What kind of feedback will the control algorithm have?)
The main objectives of the control code are to (1) ensure that the mechanical sub-systems operate in a smooth and time-efficient manner and (2) allow the user to manage the execution of code with an easy-to-follow graphical user interface. More specifically, the graphical user interface must allow the user to (1) start the control code, (2) monitor the progress of the control code, and (3) be able to stop the control code. Variables such as program complexity, time, and the likelihood for bugs to occur are examples of factors that need to be considered. One question that must be answered is where exactly the user will be able to stop the program. Other questions that need to be addressed are how exactly the user need to "monitor" the progress of the code, and how to prevent the user from executing unwanted commands while the water sampler is operating(i.e. pressing the start button more than once while program is running).

Research

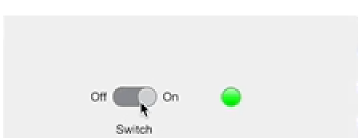
- Do some online research to on MATLAB about user interfaces and GUI's.
- Compare three GUI's done in MATLAB to what you might need to do for this project.



This first GUI has components such as a start and stop button that allows the user to control how long the program runs. Although I will not be specifically creating a timer GUI, this GUI works so that once a user presses start, they will not be able to press the start button again until the stop button is pressed. Such as feature can help prevent the user from accidently ruining the water sampler machinery. In addition, this GUI also has a display feature, which could be useful for the monitor aspect.



This graphical user interface might look simplistic, but what is particularly valuable about this interface is that the text message next to the lamp changes according to input by user. As with the first GUI, this change in text might be particularly useful for helping the user monitor how the water sampler is progressing.

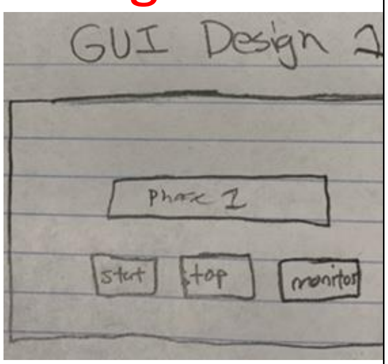


Unlike the other two graphical interfaces, this graphical user interface using a toggle switch instead of two push buttons. Therefore, using a toggle switch is an alternative to push buttons that should be investigated, as this feature can essentially combine a separate start and stop button into one.

Conceptualization – Defining possible solutions

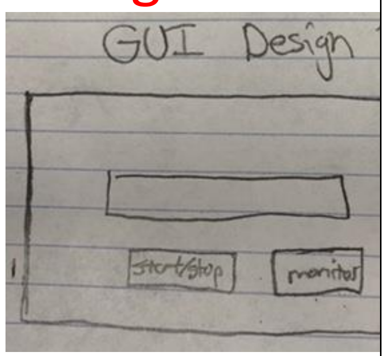
- *Place three images of possible GUI's in the boxes below.*
- *How hard is it going to be to get these functional? Does this address what the user will need to control/monitor? How easy will it be to bring team members' code into this system.*
- *At your next team meeting review these results with your team for feedback. The feedback will be recorded in the Decision Matrix in the Next section.*

Design 1



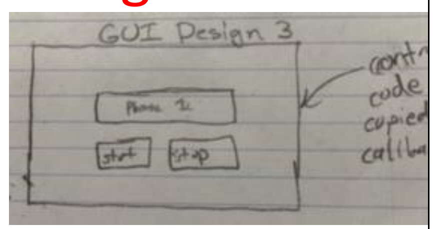
Justification: This design uses three separate buttons. The start button will initiate the program, the stop button will stop the program when needed, and the monitor button would allow the user the control whether or not they could see the progress of the water sampling process.

Design 2



Justification: Unlike designs three and one, this one combines the functions of the start and stop button into one toggle button. As with the first design, the monitor button would allow the user to turn the visibility of the progress text on and off.

Design 3



Justification:
Like design one, this GUI uses two push separate push buttons for starting and stopping the control code. Unlike design 2 and design 1, however, the monitor option is generated automatically. This means that the user will not have control over the visibility of the text on the monitor screen.

Conceptualization – Evaluating Your Designs
Bring a blank decision matrix to the team meeting with your best ideas for the criteria against which the design should be evaluated. At this point the designs are your flowcharts or optionally some initial code that illustrates the flowcharts.

- Review the decision criteria with your team before beginning.
- Which design will do a better job meeting the goals based on the given criteria?
- Be sure your team thinks through: risk, time required, system interfaces
- End the meeting with a decision on the design to pursue, or with a date to come back with additional information so the decision can be finalized.
- Load your team's decision matrix in the space below.

Decision Matrix for Control Code GUI

Evaluators: Grant Eyer, Garrett Matheny, Dorain Lewis, Christian Gratz, and Joshua Hagan

Scale: 5 - High ability to meet; 1-Low ability to meet

Specification	Weight	Design 1	Design 2	Design 3
GUI layout (appearance)	2	4	3	4
Navigability of GUI	3	3	3	4
Complexity of GUI code	4	3	2	3
Functionality	4	3	2	4
User Control	3	4	3	2
	Sum	53	40	54

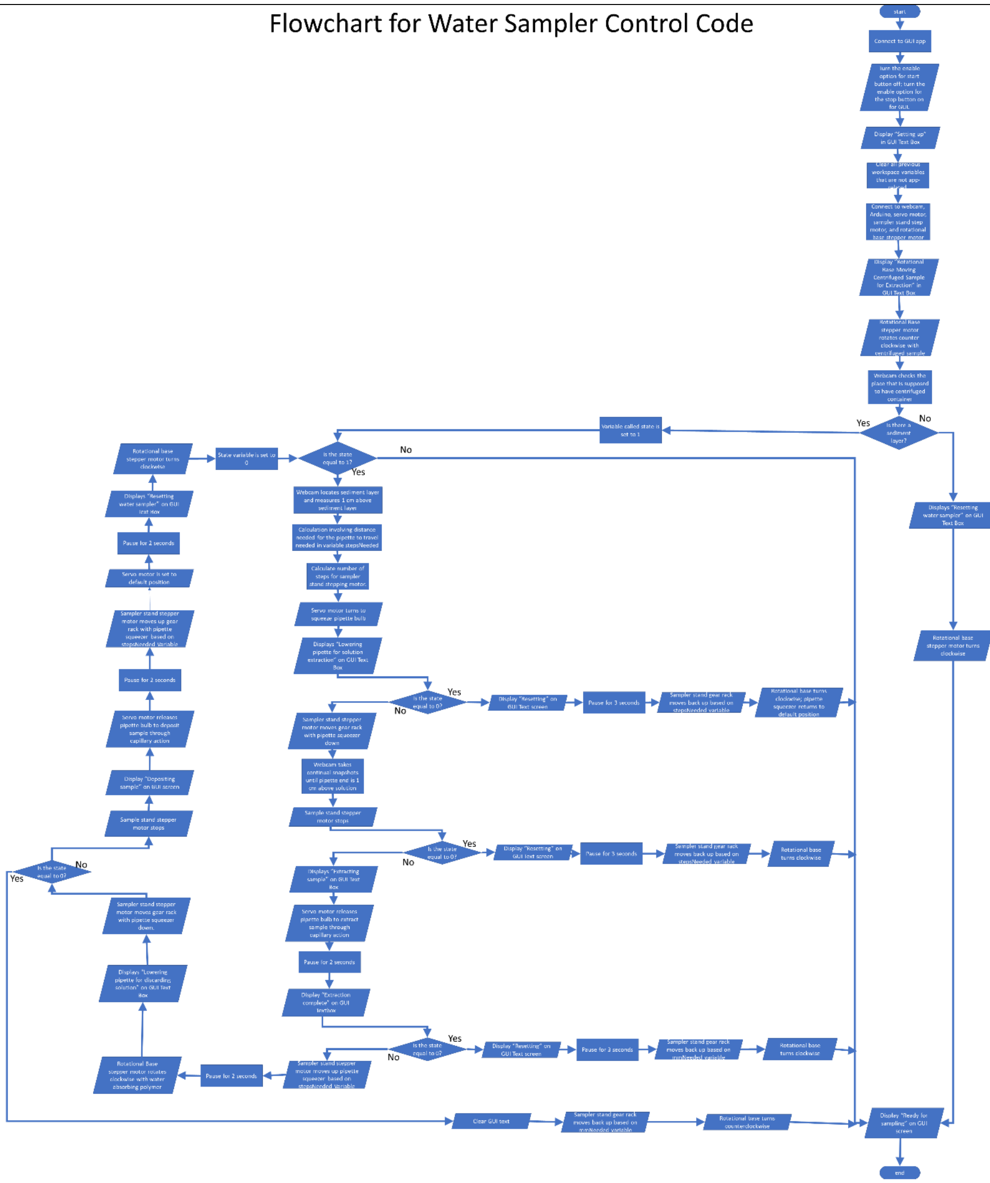
Justification: The design criteria that were used to determine the best GUI were the following: GUI Layout (appearance), navigability of GUI, complexity of GUI code, functionality, and user control. GUI Layout was simply based on how well every component in the GUI was oriented. Navigability referred to the potential for a user to be confused while using the GUI. The GUI code complexity refers to how many callback functions are involved, as well as the content within each callback code. The functionality of the code refers to the overall efficiency of the system. User control refers to extent that the user has control over what is displayed in the GUI. Based on these criteria, Design 3 was the winner. Design three was the weakest when it came to user control, as the user can't turn off the visibility of the monitor screen text. Our team thought that this aspect was negligible, since there was no requirement that stated that the user needed to control the visibility of the monitor text. For design 3, the complexity of the GUI code is moderate because there are only two callback functions involved: one for the start button and one for the push-button. This also allows for easier troubleshooting and regulation of user controls. Moreover, the layout of design 3 is relatively easy to follow and clearly lays out the options for the user to execute.

Synthesis - First Cut of your design

- Create a more detailed flow chart from your preferred design. Show more specifically how you want the control code to be structured.
- Put in a one to two sentence description of what you think will be the most difficult / highest risk areas.
- Think about what you will need to do to make this a functioning design:
  - If not already done, you can use the flowchart to write a first pass of the code (what will that take?).
  - You will need to test it, and once you start getting real inputs you will need to finalize how you are going to pull in information from the vision system and interact with the other parts of the pipette squeeze/lifting mechanism.
- You will need to create your user interface. Prove that your push button interface is activated so that something can happen (like moving the motor).
- Then you will need to integrate with the team design adding features to the GUI.
- Now think about how long each of these steps will take you, given you have lots of things you need to do in your life.
  - Come up with a general timeline.
  - List those key steps and time required.
  - When your team makes a Gantt chart, use this to put into your team's schedule plan.

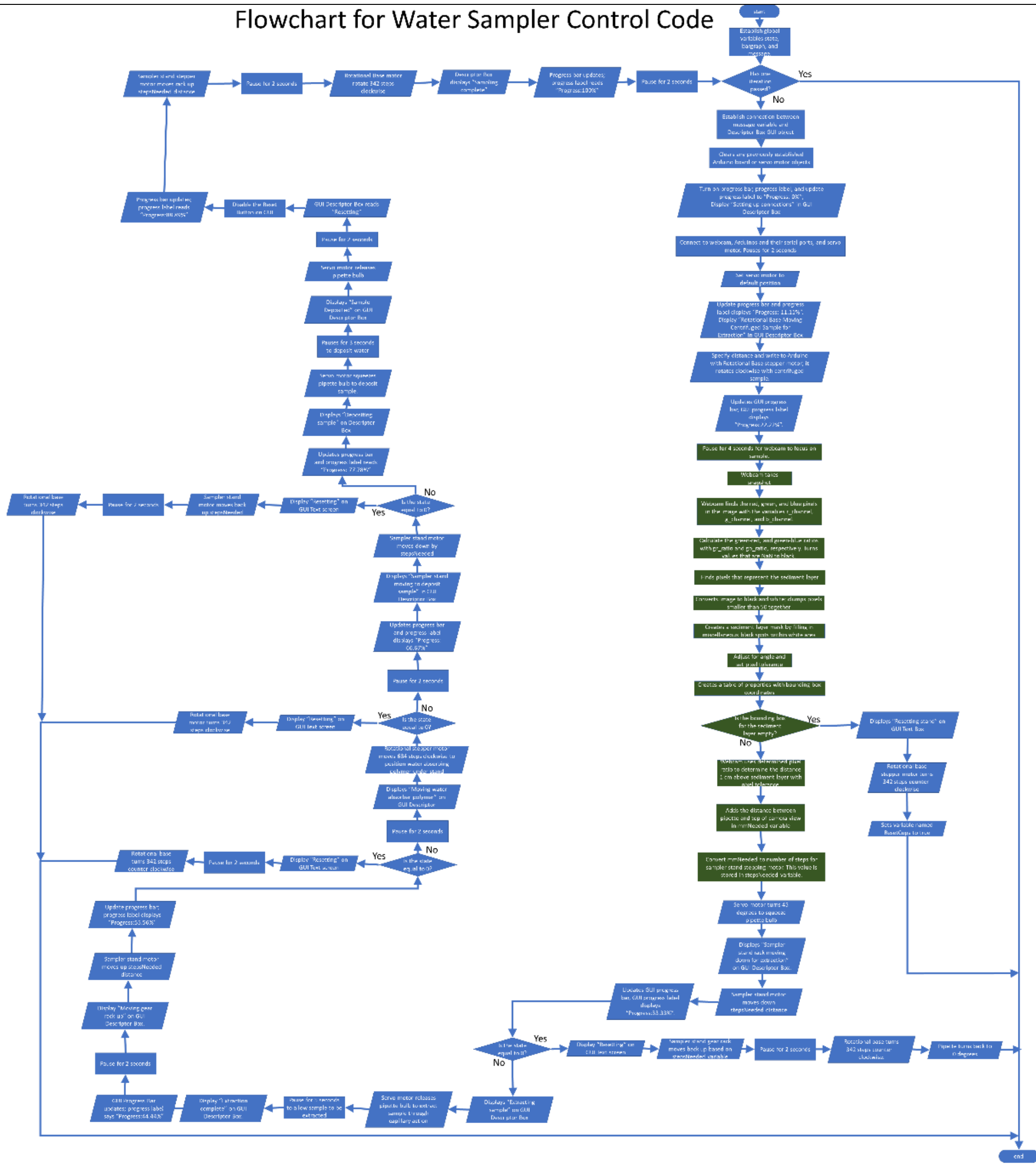
The most difficult or high-risk areas for my control code would be managing the movement of my sampler stand with my rotational base. In other words, I don't want the rotational base moving while the stand is lowered in either the centrifuged sample or water-absorbing polymer containers. Another challenging place in my code will be how to appropriately account for the total distance that the pipette needs to travel so that it is approximately 1 cm above the sediment layer. This aspect of the control code will be very dependent upon the manner that the vision system interprets the pixels in the image in relation to real-world units. Something important to note regarding the flowcharts below is that there more than likely be significant changes to the layout of the program: tests with our assembled water-sampler machine will need to be conducted to create a more appropriate flowchart representation of the control code.

Flowchart for Water Sampler Control Code

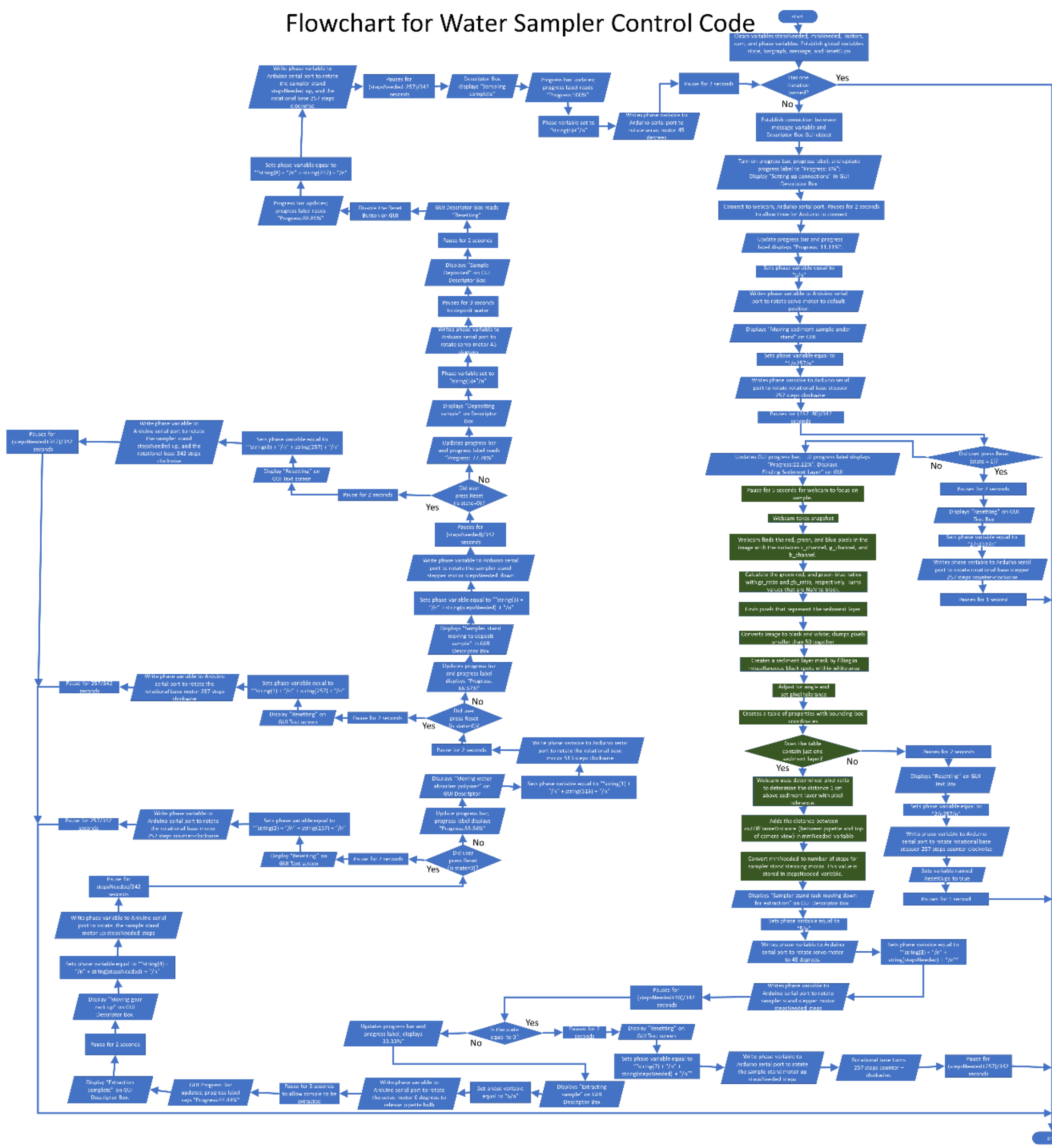


**Updated flow chart on April 15, 2021: Changed while loop to for loop. The green boxes in my image represent the vision system code.

Flowchart for Water Sampler Control Code



**Updated flow chart on April 18, 2021: Since I have established all my motor objects in Arduino rather than Matlab, this causes dome fundamental changes to my design. The green boxes in my image represent the vision system code.



Synthesis – Model

- A document has been created for you to help you through modeling some of the critical aspects of your design. Use that document to create the calculations and modeling listed. Place those calculations and results in the section below.

To create my model graph as shown below, I needed to consider the following:

Our system consists of a gear with a 32.75 mm radius that will be connected to a stepper motor; this gear will help move an attached gear-rack that will move the pipette squeezer containing the pipette up or down. Based on our design, the maximum distance that the stepper motor needs to be able to travel is approximately 13 in, or about 330 mm.

To determine how far our motor would need to move, the circumference of the gear needed to be calculated. This is the distance that our gear would travel in one revolution, in mm:

Circumference = 2*pi*32.75 mm = 205.8 mm

Once the circumference was calculated, the revolutions for specified increments of 2 mm were calculated by dividing the distance that the gear needed to travel (in mm) by the gear circumference. For example, the number of revolutions that the stepper motor would need to travel for 2 mm is calculated as follows:

Number of revolutions = 2 mm * (revolution/205.8 mm) = .00972 revolutions

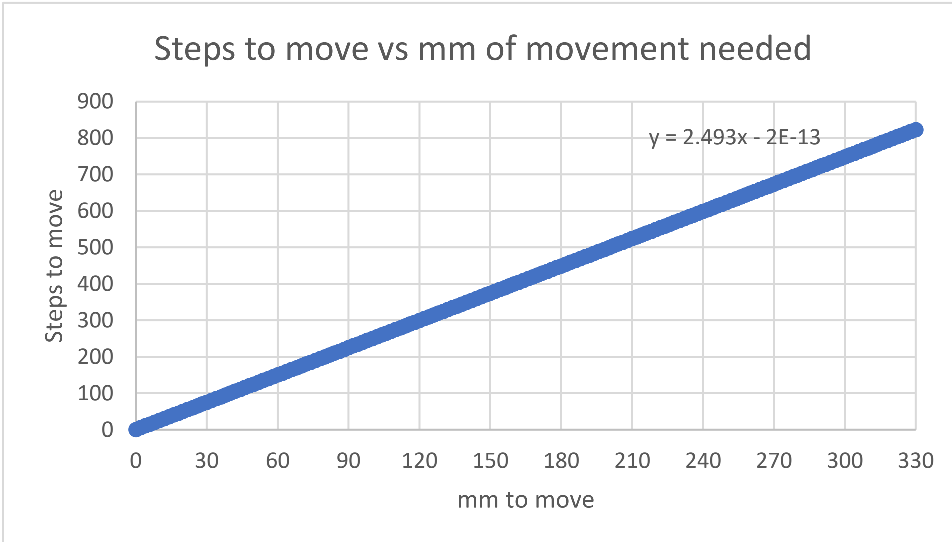
This process was repeated for various distances up to 330 mm on an excel spreadsheet.

Once the revolution per motor was determined, the steps for each distance needed to be calculated. This is because the operating code for the stepper motor requires an input relating to the number of steps that the motor must run, rather than the number of revolutions.

To calculate the steps that the stepper needs to move for 2 mm, for example, we need to know how many steps occur for one revolution of the stepper motor. In our team's case, it takes 513 steps for a stepper motor to use this revolution. Since it was determined it would take approximately .00972 revolutions to move the motor, the number of steps that would be need to be executed would be as follows:

Steps needed = .00972 revolutions * (513 steps/revolution) = 4.986 steps.

This process for calculating the steps needed for distances up until 330 mm was determined with an excel spreadsheet. Once these calculations were finished, the graph below was plotted with the steps needed to move versus the mm of movement needed. The equation that describes the relationship between these two quantities is y = 2.493x, where x represents the distance the stepping motor gear needs to move versus the number of steps required for the gear to move that specified distance.



Development			
Step	Activity	Duration	Date to Complete
1	Create a rough user-interface	7 days	COMPLETED
2	Gather subassembly codes for integration into a reasonable control code structure	Between 2 to 4 days	April 5,2021
3	Modify GUI control code based on peer review	Between 2 to 4 days	April 10, 2021
4	Modify overall layout of GUI design; make more visually appealing and professional-looking	Between 1 to 2 days	April 15, 2021

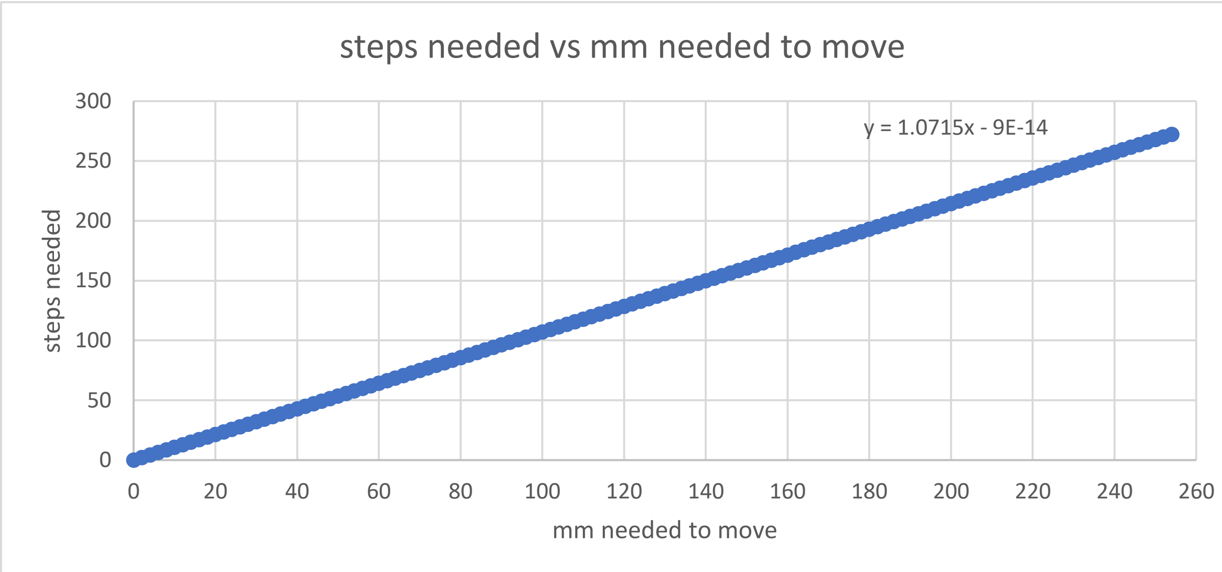
On April 7, 2021: gear size was adjusted to 76.2 mm. The overall maximum distance that the stepper motor would need to travel was also determined to be approximately 10 in, which is 254 mm.

Circumference = 2*pi*76.2 mm = 478.78 mm

Number of revolutions = 2 mm * (revolution/478.78 mm) = .00418 revolutions

Steps needed = .00418 revolutions * (513 steps/revolution) = 2.14 steps.

Updated Graph and model:



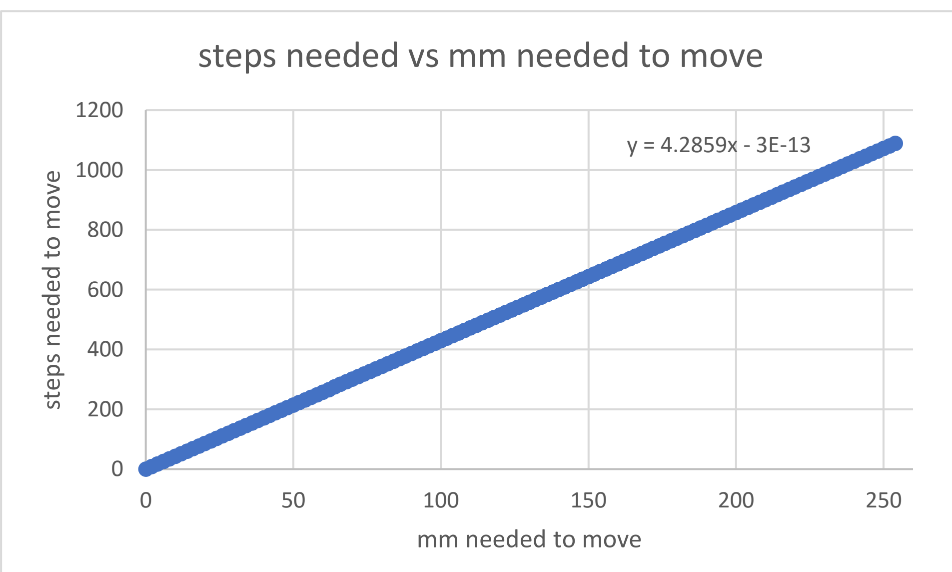
On April 14, 2021: gear size was adjusted to 76.2 mm. The overall maximum distance that the stepper motor would need to travel was also determined to be approximately 10 in, which is 254 mm.

Circumference = 2*pi*76.2 mm = 478.78 mm

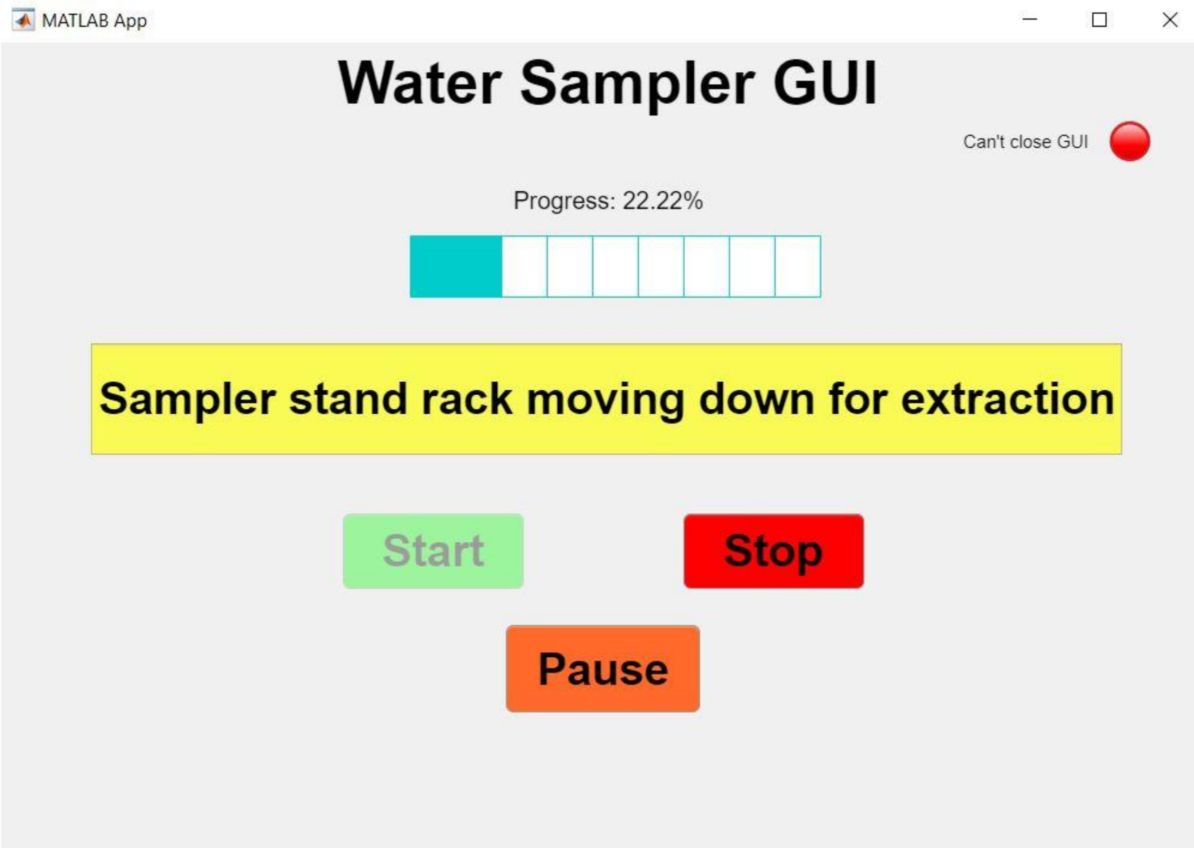
Number of revolutions = 2 mm * (revolution/478.78 mm) = .00418 revolutions

Steps needed = .00418 revolutions * (2052 steps/revolution) = 8.58 steps.

Updated Graph and model:



Evaluation – Calculations • What conclusions are you able to make from your model?
Based on our team's design, the equation that needs to be used in order to tell the stepper motor how far the gear rack needs to descend would be y = 2.493x, which would be incorporated into the control code as follows: stepsNeeded = round(2.493 * mmNeeded) The stepper motor of our gear must be able to travel a maximum distance of 330 mm, which equates to about 823 steps. Therefore, the stepper motor on the sampler stand needs to be able to turn about 823 steps at most. As of when this POC document was submitted, the sampler stand stepper motor was considered to be capable of turning this many steps. **Updated April 10,2021: the new model was determined to be the formula below stepsNeeded = round(1.0715 * mmNeeded) **Updated April 14,2021: the new model was determined to be the formula below stepsNeeded = round(4.2859 * mmNeeded)

Evaluation – Prototype • You will create a working example of your design concept. Take an image and place it in the space below.
**Updated April 15, 2021: Image of GUI inserted below 

EGR 103 - Engineering Exploration 2

Peer Review Template ______Sampler Stand_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
The meeting was delivered with promptness and had only had one suggestion that was advised. The suggestion includes the use of a pin to hold a bolt which is connecting the base and the sampler stand. This displays that, while there is a part that can be improved, the overall design is ready for fabrication. The change that was offered will be considered and implemented as soon as possible.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
• Instead of using a screw, use a bolt that can be held using a clip.

Action Items *Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.*
An action that needs to be taken is to fasten the rotational base to the sampler stand using a bolt with a clamp that can be attached or removed at any point. This will allow fixing to be available with minimal damage to the other components. This part will need to be designed and attached by Tuesday, March 30. This will be the day that the prototype of the system is due.

Performance test • How did the demonstration of you sub-component go? • Upload an image of your built design after answering the questions above.
The subcomponent was not available on the date of presenting the Sampler Stand.

Rotational Base

Pipette Squeezer

Vision System

Control Code

Vision System Peer Review

College of Engineering

EGR 103 - Engineering Exploration 2

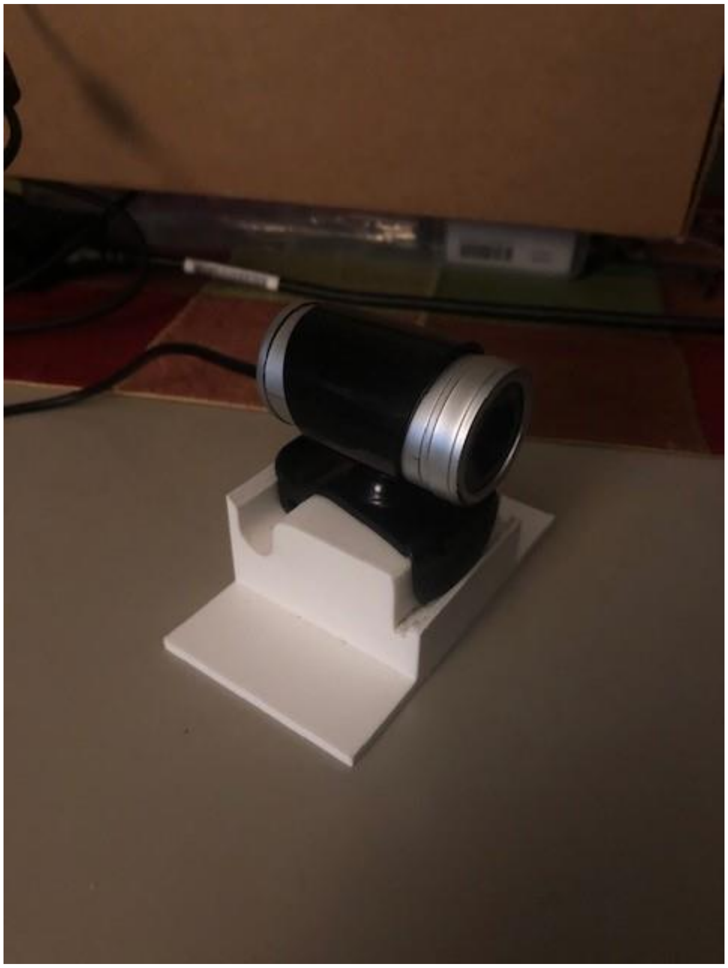
Peer Review Template ______Vision System_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
The vision system code was shown and how it accounts for angle tolerance, and I discussed the placement of the vision system (under the stand). Unfortunately, our tower was not assembled so I was unable to show the actual vision system in the design. I just used a picture I had taken previously to showcase the ability of the code to distinguish a pipette layer and a sediment layer.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
• Make sure the computer has enough USB ports • Instead of having the camera wait for the pipette to move into position, have the camera find just the sediment layer and move the pipet from position 1 accordingly, as the pipette's base position should stay the same.

Action Items *Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.*
Before the final demonstration, I need to work on the code for the vision system and change it to the suggestion that could remove an entire step in the process. If I code the vision system to only have to find the sediment layer, it removes an entity which could go wrong in the demonstration. With this change, the chance that the vision system will make a mistake is significantly lowered. I will have this done at least the week before the final demonstration.

After some time, a team member figured out how we could do the design without using more than one Arduino board, so we do not have a problem with the USB ports any longer. That design will be put together the week before the presentation.

Performance test • How did the demonstration of you sub-component go? • Upload an image of your built design after answering the questions above.
The demonstration of my subcomponent went well other than the fact that I could not show it connected to the entire tower as we did not have it assembled yet. 

Control Code Peer Review

EGR 103 - Engineering Exploration 2

Peer Review Template ______Control Code_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
During the peer review, I was not able to demonstrate the full functionality of the GUI design because the water sampler machine was not assembled. However, I did use a "pseudo-code" to demonstrate how the GUI would potentially run. There was also a brief discussion on how the control code would be structured.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design

The meeting with the peer review teams went well. There were little suggestions provide and questions asked. One concern was will there be a pause function added for emergency purposes (i.e. water sampler machine malfunctions and can't properly stop).

- Add a pause button

Action Items

Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.

The only suggestion that the groups gave me has been implemented into the GUI design (see Performance Test block below for GUI image).

Although there were no other suggestions made, I have recognized that I need to implement a code structure that increases the speed of the stepper motors. The current problems that I am facing as of writing this are the following:

- By default, the stepper motors are incredibly slow using Matlab. A solution to this problem is using Arduino IDE code to speed up the stepper motors. However, this solution also creates a problem. That problem relates to the fact that Matlab needs to be able to send commands to an Arduino sketch that tells it which stepper motor(s) need to move, so that the Arduino can then control how many steps the stepper motors need to move.
- The servo motor connection, along with the stepper motors, causes some COM port problems.

I have planned to resolve the second problem by incorporating an additional bread-board and Arduino to the design. The plan is to have the servo-motor run on one Arduino board with its respective Matlab commands. The other breadboard will just have the servo motor. Of course, this additional breadboard would require that my computer have enough ports to also include the webcam for the vision system. One proposed way around this problem is to receive a hub, which would allow me to connect more devices to my computer.

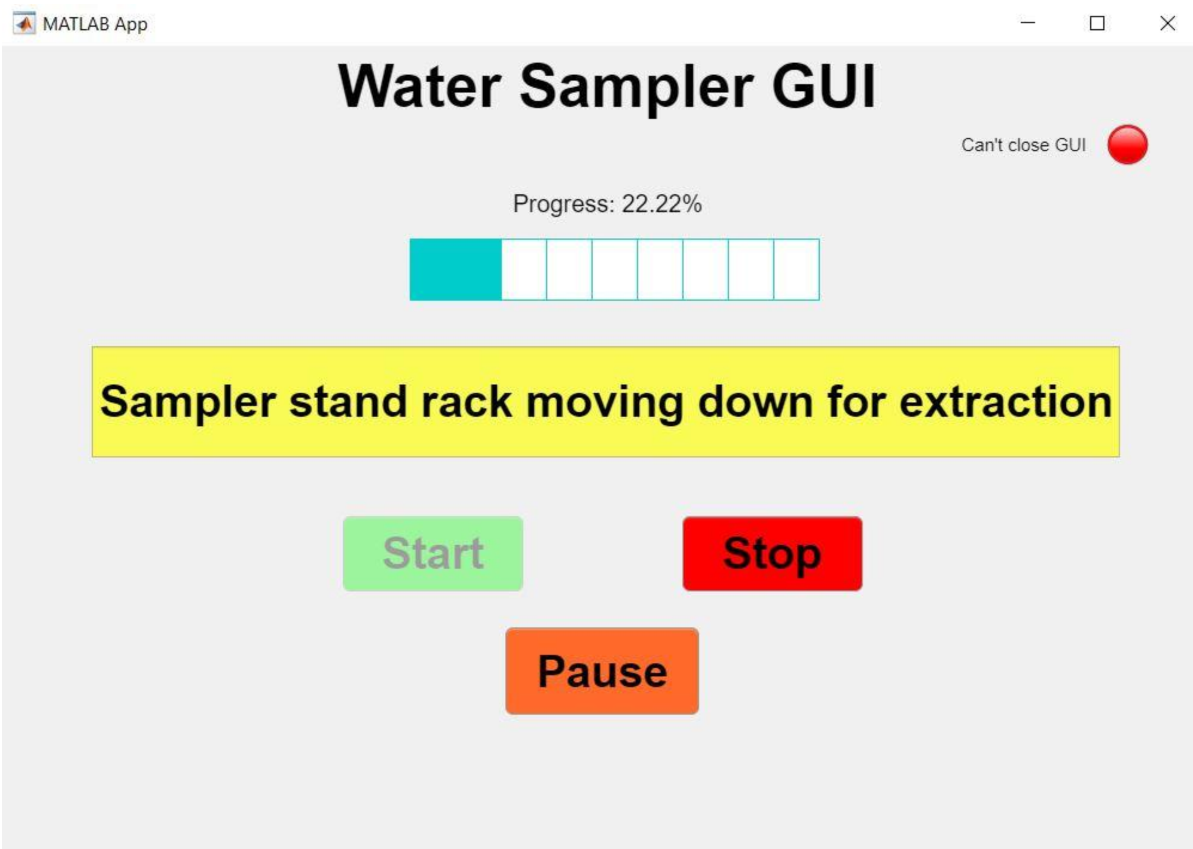
The first problem I have not entirely resolved yet, though my Arduino code works within the Arduino IDE and the serial motor as of submitting this peer review. The remaining obstacle to fully addressing this problem is to figure out how Matlab will write to the serial port on my Arduino board. I have consulted with Dr.Maddox on April 8 to give me ideas on how to resolve this issue.

My plan is that I have both of these issues largely resolved, if not entirely, by April 13. This is so that I would have time to submit a relatively functional form of my control code by April 16.

Performance test

- How did the demonstration of your sub-component go?
- Upload an image of your built design after answering the questions above.

As mentioned in the Peer Review Meeting Summary, I was not able to demonstrate the full functionality of my GUI and control code because the water sampler had not been assembled at the time. To compensate for this fact, I decided to create a "pseudo-code" to show the other teams how my GUI would operate. I have shown a picture of the GUI functionality below for reference.



Materials Used / Material List (BOM)

Bill of Materials				
Description	Supplier	Quantity	Part Cost	Total Cost
Arduino Breadboard	UK-Bookstore	1	$ 5.00	$ 5.00
Insignia - 4-Port USB 2.0 Hub - Black	Best-Buy	1	$ 21.19	$ 21.19
Arduino Uno	UK-Bookstore	1	$ 23.00	$ 23.00
28BYJ-48 Stepper Motor	Adafruit	2	$ 4.95	$ 9.90
Pipette Squeezer 3D-print	UK-Paducah Campu	1	$ 0.26	$ 0.26
4x4x8 Pressure-Treated Timber	Hobby Lobby	1	$ 14.00	$ 14.00
20 in x 30 in Plastic Corrugated Sheet	Michaels	1	$ 3.81	$ 3.81
T - Star Washer Head PowerLags 1/4" x 6	Hobby Lobby	3	$ 0.90	$ 2.70
T - Star Washer Head PowerLags 1/4" x 5	Hobby Lobby	3	$ 0.68	$ 2.04
Sandpaper	Harbor Freight	1	$ 4.11	$ 4.11
Balsa 5 pack 3/16 in x 3/16 in x 36 in	Hobby Lobby	1	$ 4.99	$ 4.99
VILROS SG90 Servo Motor	UK-Bookstore	1	$ 2.49	$ 2.49
Milwaukee Compact Billet Torpedo Leve	Lowes	1	$ 29.99	$ 29.99
			Total	$ 123.48

Development

Data

Sampler Stand: **Synthesis – Model**

Rotational Base: **Synthesis – Model**

Pipette Squeezer: **Synthesis – Model**

Vision System: **Synthesis – Model**

Control Code: **Synthesis – Model**

Ideas/Sketches

Sampler Stand: **Conceptualization – Defining possible solutions**

Rotational Base: **Conceptualization** – Defining possible solutions

Pipette Squeezer: **Conceptualization** – Defining possible solutions

Vision System: **Conceptualization – Defining possible solutions**

Control Code: **Conceptualization – Defining possible solutions**

Calculations

Sampler Stand

We use the center of mass calculation to determine how much weight the stand can handle before tipping over.

Center of Mass:

((5.5” * 10lbs)+(11” * 3lbs)+(8” * 5lbs))/(18lbs)

128”lbs/18lbs = 7.11”

Rotational Base

The rotational base involves two gears: a driver gear that is 6 in. in diameter and a gear under the base that is 3 in.

We needed to be able to determine the number of steps that the stepper motor under the driver gear needed to turn so that the rotational base would move 90 degrees.

$$r_{driver}\theta_{driver} = r_{driven}\theta_{driven}$$

$$3\ in * \theta_{driver} = 1.5\ in * 90°$$

$$\theta_{driver} = 45°$$

Then, we convert the angle to steps using following facts:

- 360 degrees is in one revolution, so 45 degrees is an eighth of a revolution
- One revolution of the rotational base stepper motor is 2048 steps

Converting angle to steps:

2048 steps/8 = 256 steps.

Therefore, the stepper motor needs to turn 256 steps in order to rotate the rotational base 90 degrees. The number of steps needed to rotate the base 180 degrees would be double that number of steps, or 512 steps.

**April 28,2021: due to the rotational base not fully functioning with the two gears underneath, such calculations were deemed unnecessary. Signed by Grant Eyer

Pipette Squeezer

O = (-297.18g * 10.22cm) + (-122g * 7.3cm)+ (F_B * 6.35cm)

(6.35) * F_B = 3927.83g*cm

F_B = 618.56g (Force pushing up on the base)

-

$O = (F_L * 8.76cm) + (-122g * 7.3cm) + (618.56g * 6.35cm)$

$F_L = 346.6g$

$T_{motor} = (346.6g) * (2cm)$

$T_{motor} = 693.2g*cm$ (torque of the lever arm)

Vision System

Camera Angles:

Horizontal Camera Angle:

Distance between Markers = 1.5 m
Distance from Camera to Markers: 2.04 m
Half Horizontal Angle = sin^-1(0.75m/2.04m)
Half Horizontal Angle = 21.57 degrees
Full Horizontal Angle = 43.14 degrees

Vertical Camera Angle:

Distance between floor and marker: 1.02 m
Distance between camera and wall: 2 m
Full Vertical Angle = sin^-1(1.02m/2m)
Full Vertical Angle = 30.66 degrees

Pixel Conversion Factor and Tolerance

Fiducial Resolution:
Proposed distance from camera to cup: 0.127 m
Field of view width: 100.4 mm
Field of view height: 69.6 mm
Resolution of camera: 640 x 480 pixels
Pixel width and height: 0.156 mm x 0.145 mm
Pixels per mm = 6.4 pixels

Pixel and Height Tolerance:

Based on the threshold between the sediment layer and the pipet, I should make a tolerance of 1.78 mm when compensating for any tilt to a max of 5 degrees. Work shown below:

Pixel tolerance (+/- 2 pixel on each edge) = 2 pixels / 6.3 pixels/mm
Pixel Tolerance = 0.32 mm

Difference in cup lid height = (35.14 mm) * sin (5)
Difference in cup lid height = 3.06 mm
Pipet Height Tolerance = +/- 1.53 mm
Pipet Pixel Tolerance = 0.32 mm
Pipet Height & Pixel Tolerance (at 5 degrees) = 1.81mm

Control Code

In order to ensure the control code operated properly, the GUI needed to be able to change visuals in an appropriate manner. If not accounted for, the Matlab visuals will not line up properly with the water sampler execution. In other words, Matlab will outrun Arduino.

Using the fact that the stepper motors have a speed of 10 revolutions per second, I needed to be able to determine the number of steps that would occur in a second.

$$\frac{10\ revolutions}{min} * \frac{2052\ steps}{1\ revolution} * \frac{1\ min}{60\ seconds} = 342\ steps/sec$$

In Matlab, the pause() function was utilized to ensure that the visuals lined up with the action of Matlab command. For example, if I needed the amount of time it would take for Matlab to turn the sampler stand stepper motor stepsNeeded amount of steps, I would insert the following:

```
Pause(stepsNeeded/342);
```

Brainstorms (Peer Reviews)

Sampler Stand Peer Review

EGR 103 - Engineering Exploration 2

Peer Review Template ______Sampler Stand_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
The meeting was delivered with promptness and had only had one suggestion that was advised. This displays that, while there is a part that can be improved, the overall design is ready for fabrication. The change that was offered will be considered and implemented as soon as possible.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
• Instead of using a screw, use a bolt that can be held using a clip.

Action Items

Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.

An action that needs to be taken is to fasten the rotational base to the sampler stand using a bolt with a clamp that can be attached or removed at any point. This will allow fixing to be available with minial damage to the other components. This part will need to be designed and attached by Tuesday, March 30. This will be the day that the prototype of the system is due.

Performance test

- How did the demonstration of you sub-component go?
- Upload an image of your built design after answering the questions above.

The subcomponent was not available on the date of presenting the Sampler Stand.

College of Engineering

EGR 103 - Engineering Exploration 2

Peer Review Template ______Rotational Base _____

Peer Review Meeting- Summary

- In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.

The meetings for the rotational base mostly consisted of discussions over the method of turning. Some groups used gears while some relied on a simple motor-ball bearing assembly. It was extremely helpful being able to see the designs of other rotational bases, as it helps us realize which of our ideas may or may not work. Overall, I believe the peer reviews help close in our design for the rotational base.

Specific Suggestions

- In a bullet form below list the specific suggestions you got for your design

- Make sure there is a space big enough for the driver gear and motor between the rotational base and the stand.
- Make sure the rotational base is tall enough to be level with the vision system.
- Make sure that the rotational base will stay stable.
- Make sure that the gear ratio will be big enough.
- Make sure that the gears will stay linked.

Action Items

Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.

- Over spring break: Put together the rotational base.
- By the next week: Have code connected and make sure base turns.
- By the week after that: Link rotational base and code to rest of sampler.
- Next week: Demonstration

Performance test

- How did the demonstration of you sub-component go?

- Upload an image of your built design after answering the questions above.

The presentation of our design went very well. Our peers seemed to understand what approach we were going for and were able to pose questions that made up think deeper into our design. We enjoyed proposing our concept to them and receiving feedback.

Image N/A

Pipette Squeezer Peer Review

College of Engineering

EGR 103 - Engineering Exploration 2

Peer Review Template ______Pipette Squeezer_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
There were many clarifications about how the squeezer will return to a neutral position, as well as a few questions about where the motor will sit.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
• Have clearance for the servo motor

Action Items *Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.*
We need to 3D print the pipette squeezer that is already modeled by next week so that we can combine all of our Water Sampling parts.

Performance test • How did the demonstration of the sub-component go? • Upload an image of your built design after answering the questions above.

The demonstration for the sub-component went well, but not many people had questions or suggestions so we did not receive much feedback from outside sources. We do not have a picture of the pipette squeezer. We only have it 3D modeled on Onshape and plan to print it out soon.

Vision System Peer Review

College of Engineering

EGR 103 - Engineering Exploration 2

Peer Review Template ______Vision System_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
The vision system code was shown and how it accounts for angle tolerance, and I discussed the placement of the vision system (under the stand). Unfortunately, our tower was not assembled so I was unable to show the actual vision system in the design. I just used a picture I had taken previously to showcase the ability of the code to distinguish a pipette layer and a sediment layer.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
• Make sure the computer has enough USB ports • Instead of having the camera wait for the pipette to move into position, have the camera find just the sediment layer and move the pipet from position 1 accordingly, as the pipette's base position should stay the same.

Action Items *Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.*

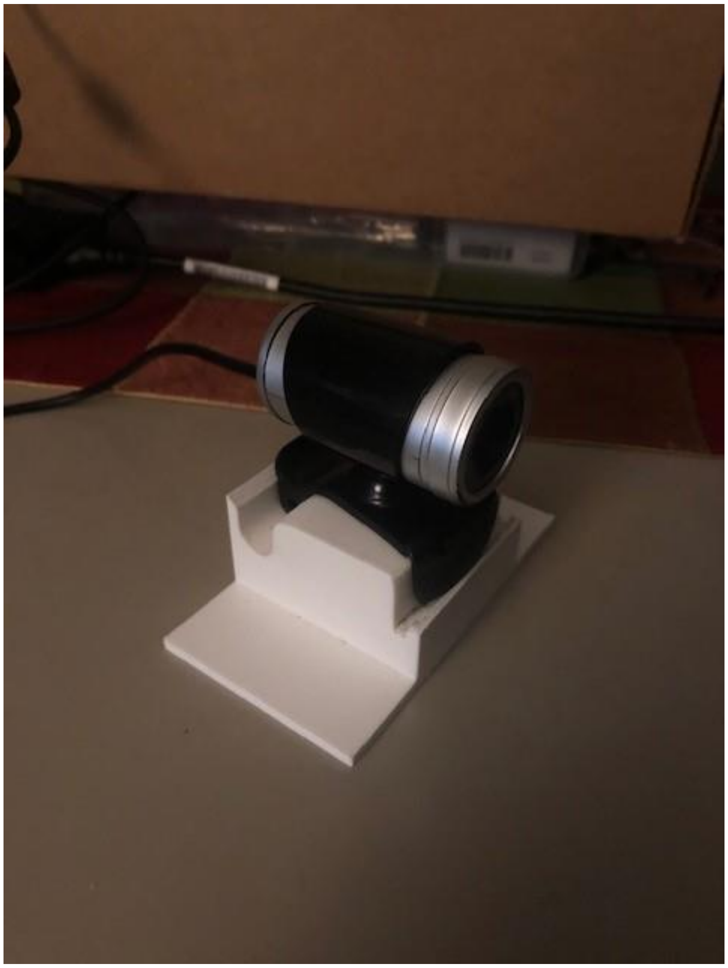
Before the final demonstration, I need to work on the code for the vision system and change it to the suggestion that could remove an entire step in the process. If I code the vision system to only have to find the sediment layer, it removes an entity which could go wrong in the demonstration. With this change, the chance that the vision system will make a mistake is significantly lowered. I will have this done at least the week before the final demonstration.

After some time, a team member figured out how we could do the design without using more than one Arduino board, so we do not have a problem with the USB ports any longer. That design will be put together the week before the presentation.

Performance test

- How did the demonstration of you sub-component go?
- Upload an image of your built design after answering the questions above.

The demonstration of my subcomponent went well other than the fact that I could not show it connected to the entire tower as we did not have it assembled yet.



Control Code Peer Review

College of Engineering

EGR 103 - Engineering Exploration 2

Peer Review Template ______Control Code_____

Peer Review Meeting- Summary • In a paragraph below summarize the results of the meeting in which you reviewed your design with a peer team.
During the peer review, I was not able to demonstrate the full functionality of the GUI design because the water sampler machine was not assembled. However, I did use a "pseudo-code" to demonstrate how the GUI would potentially run. There was also a brief discussion on how the control code would be structured.

Specific Suggestions • In a bullet form below list the specific suggestions you got for your design
The meeting with the peer review teams went well. There were little suggestions provide and questions asked. One concern was will there be a pause function added for emergency purposes (i.e. water sampler machine malfunctions and can't properly stop). • Add a pause button

Action Items *Based on the suggestions you got, what do you thing are actual action items that you could work on for the next few weeks and complete before the design demonstration? Include WHAT you need to do, WHEN you need to do it by and if there are any additional resources (materials or equipment) you need to procure to make this happen.*
The only suggestion that the groups gave me has been implemented into the GUI design (see Performance Test block below for GUI image).

Although there were no other suggestions made, I have recognized that I need to implement a code structure that increases the speed of the stepper motors. The current problems that I am facing as of writing this are the following:

- By default, the stepper motors are incredibly slow using Matlab. A solution to this problem is using Arduino IDE code to speed up the stepper motors. However, this solution also creates a problem. That problem relates to the fact that Matlab needs to be able to send commands to an Arduino sketch that tells it which stepper motor(s) need to move, so that the Arduino can then control how many steps the stepper motors need to move.
- The servo motor connection, along with the stepper motors, causes some COM port problems.

I have planned to resolve the second problem by incorporating an additional bread-board and Arduino to the design. The plan is to have the servo-motor run on one Arduino board with its respective Matlab commands. The other breadboard will just have the servo motor. Of course, this additional breadboard would require that my computer have enough ports to also include the webcam for the vision system. One proposed way around this problem is to receive a hub, which would allow me to connect more devices to my computer.

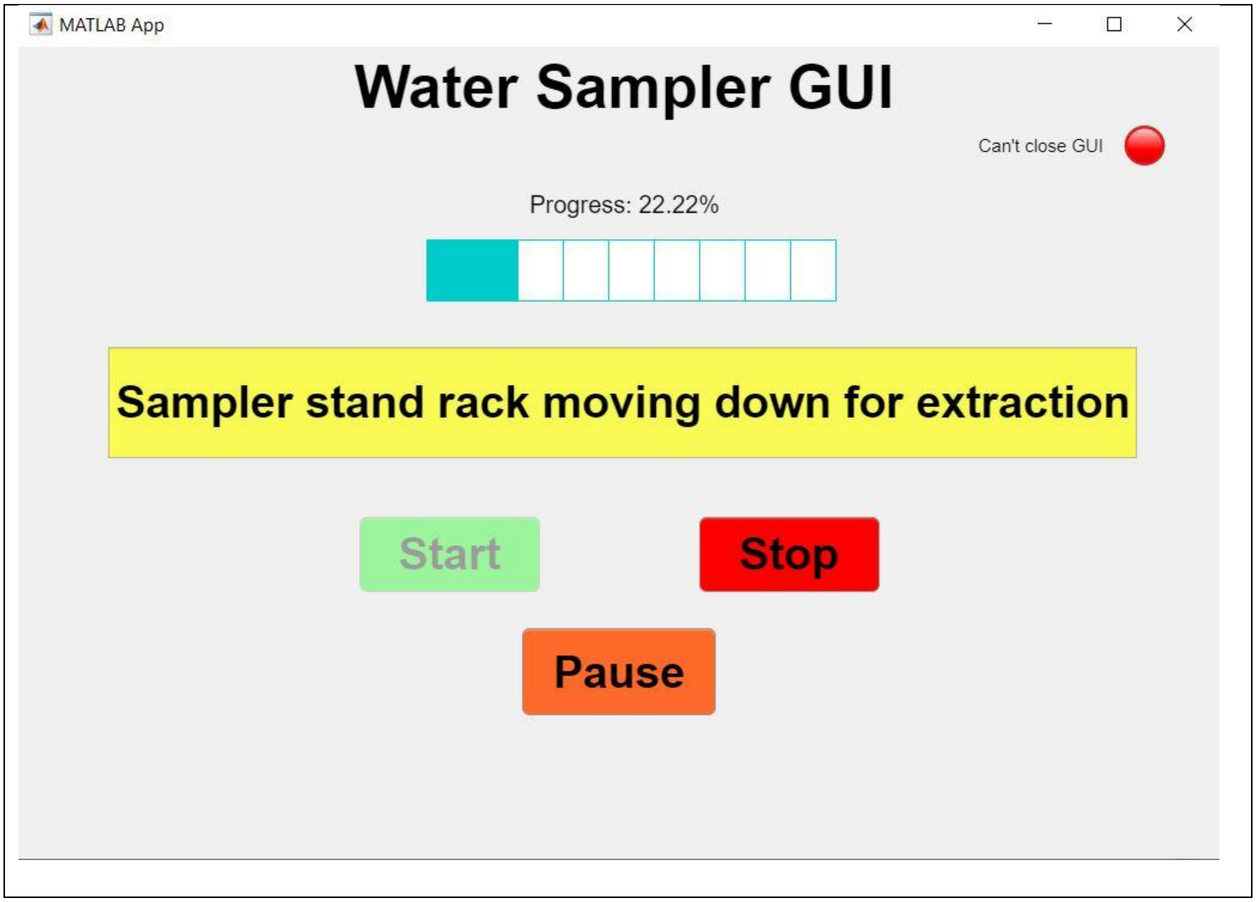
The first problem I have not entirely resolved yet, though my Arduino code works within the Arduino IDE and the serial motor as of submitting this peer review. The remaining obstacle to fully addressing this problem is to figure out how Matlab will write to the serial port on my Arduino board. I have consulted with Dr.Maddox on April 8 to give me ideas on how to resolve this issue.

My plan is that I have both of these issues largely resolved, if not entirely, by April 13. This is so that I would have time to submit a relatively functional form of my control code by April 16.

Performance test

- How did the demonstration of your sub-component go?
- Upload an image of your built design after answering the questions above.

As mentioned in the Peer Review Meeting Summary, I was not able to demonstrate the full functionality of my GUI and control code because the water sampler had not been assembled at the time. To compensate for this fact, I decided to create a “pseudo-code” to show the other teams how my GUI would operate. I have shown a picture of the GUI functionality below for reference.


MATLAB App
Water Sampler GUI
Can't close GUI
Progress: 22.22%
Sampler stand rack moving down for extraction
Start
Stop
Pause

Design/Documentation

Cost Spreadsheet

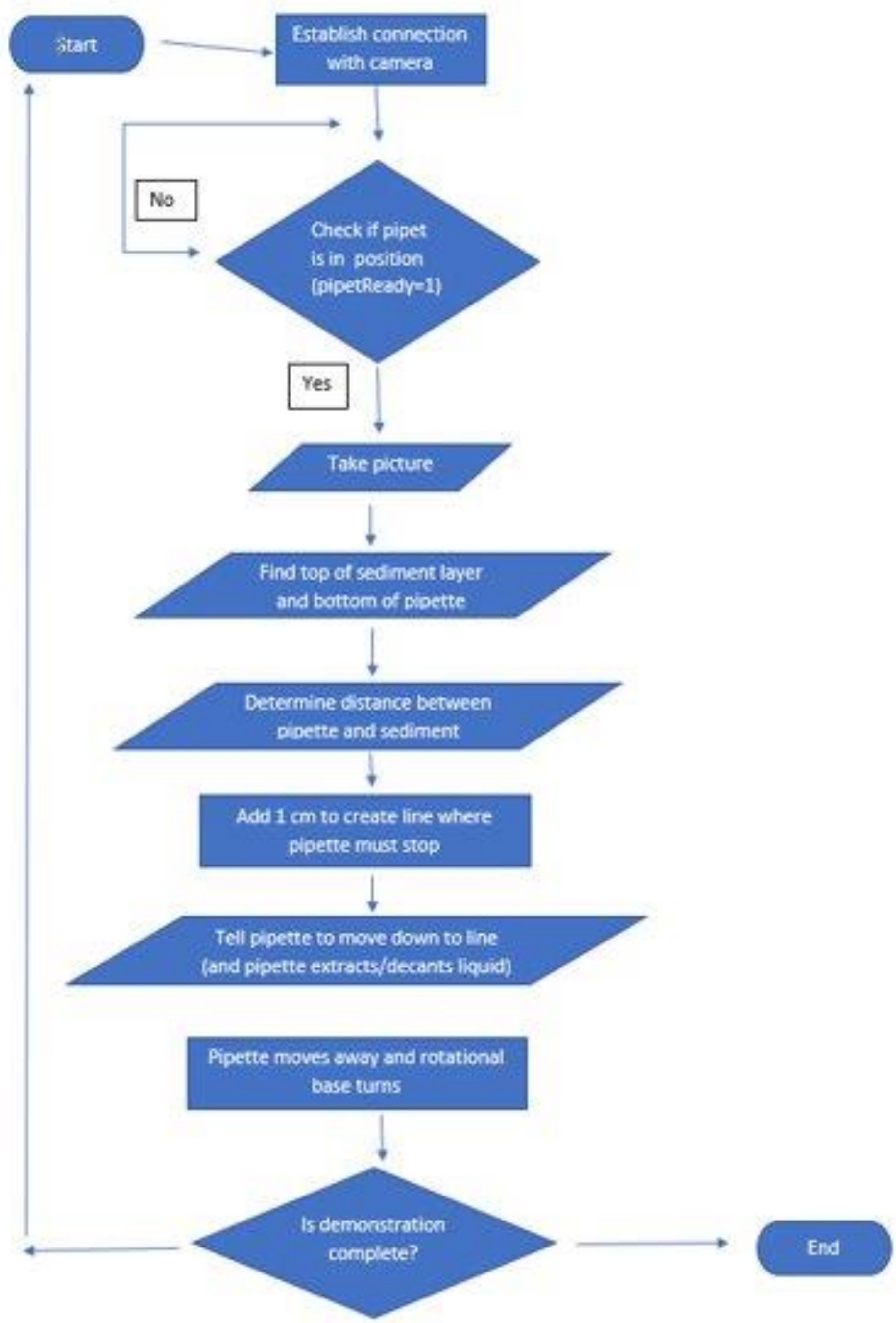
Hours Worked

Group Members	Primary Area	Total Hours	22-Feb	1-Mar	8-Mar	15-Mar	22-Mar	29-Mar	5-Apr	12-Apr	19-Apr	26-Apr	3-May
Garrett Matheny	Vision System	43	3	6	3	3	2	6	6	8	6		
Grant Eyer	Control Code	69	6	9	3	2	1	8	10	27	3		
Christian Gratz	Rotational Base	36	3	2	4	6	1	2	2	6	10		
Dorian Lewis	Sampler Stand	36	5	3	4	4	4	4	4	4	4		
Joshua Hagan	Pipette Squeezer	52	5	6	4	2	3	4	12	10	6		

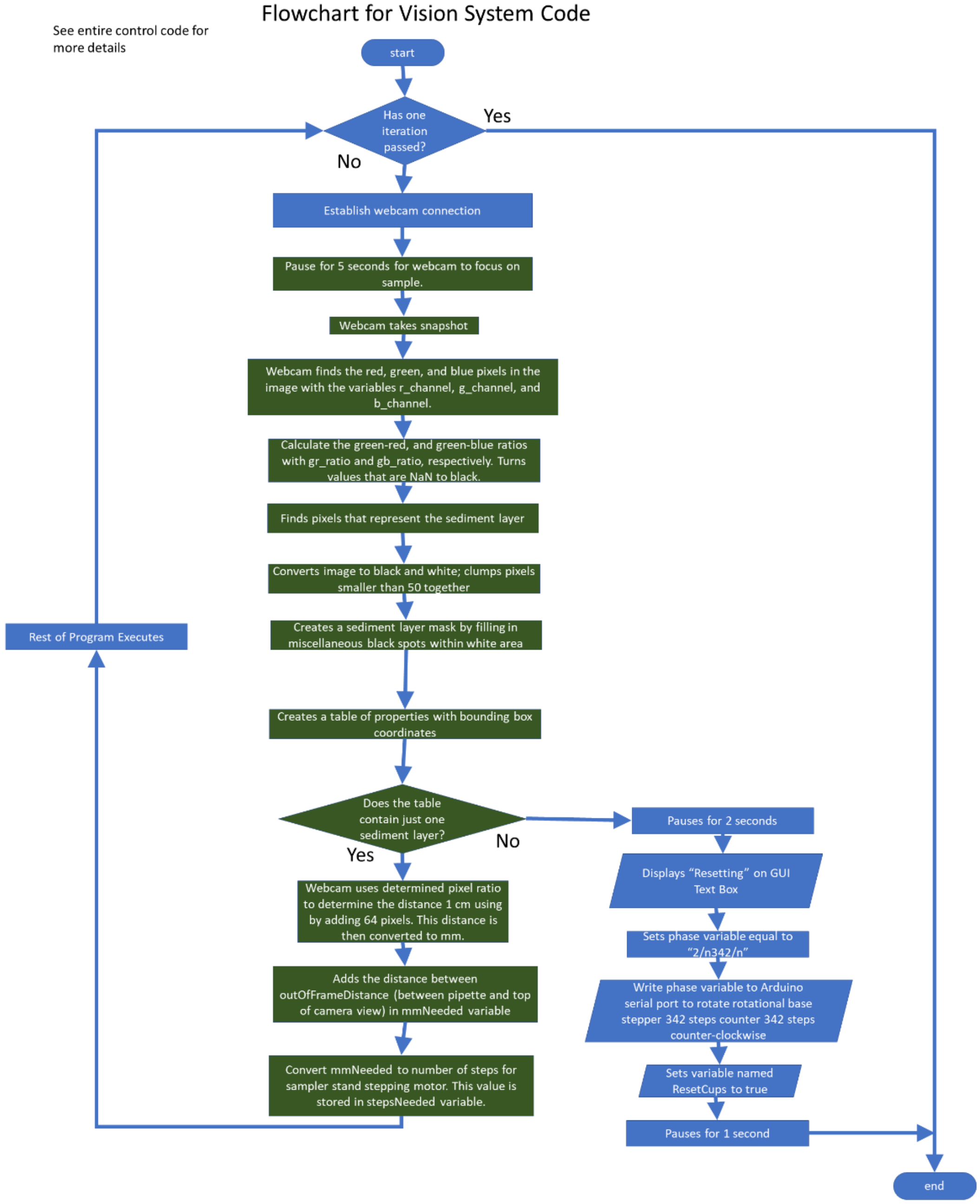
Labor Cost

EGR 103 Engineering Exploration Project Team Member Time Commitment				
Group C			Section Number: 015	
Group Member	Primary Area	Hours Worked	Cost Per Hour ($45 per hour)	Total Cost
Garrett Matheny	Vision System	43	$ 45.00	$ 1,935.00
Grant Eyer	Control Code	69	$ 45.00	$ 3,105.00
Dorian Lewis	Sampler Stand	36	$ 45.00	$ 1,620.00
Christian Gratz	Rotational Base	36	$ 45.00	$ 1,620.00
Joshua Hagan	Pipette Squeezer	52	$ 45.00	$ 2,340.00

Vision System Flowchart (Original)

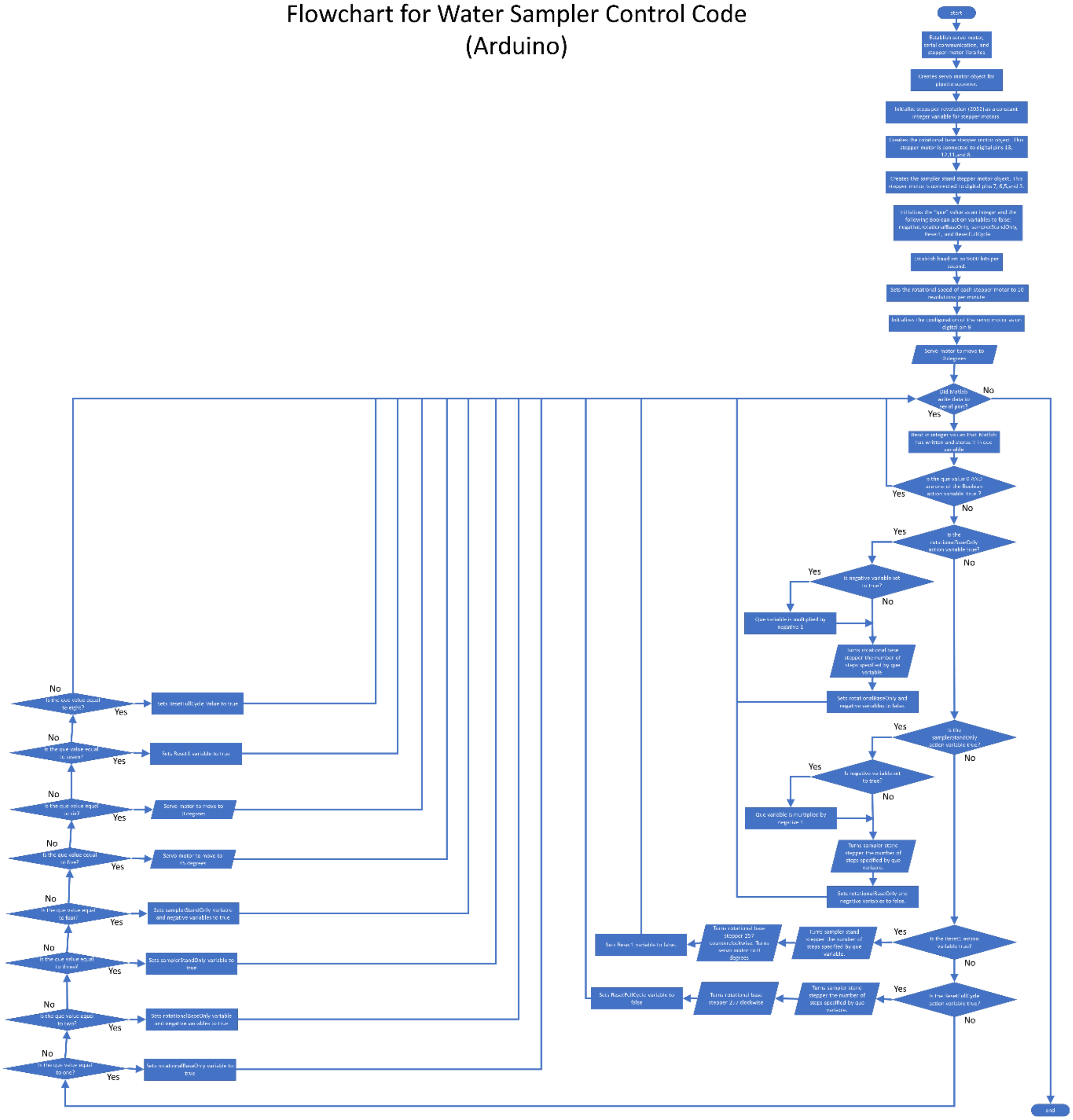


Vision System Flowchart (Revised)

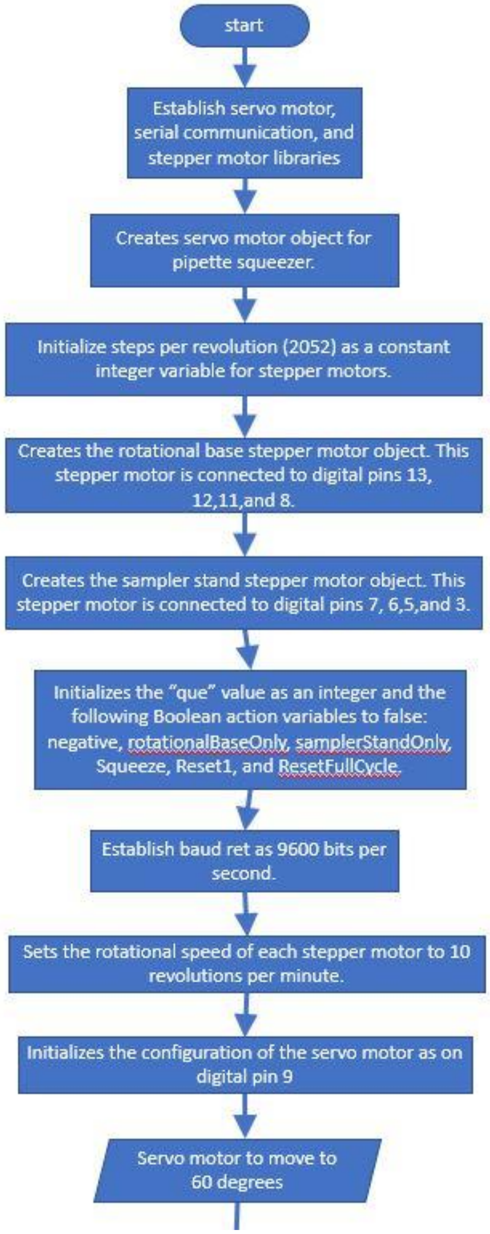


Control Code Flowchart (Arduino)

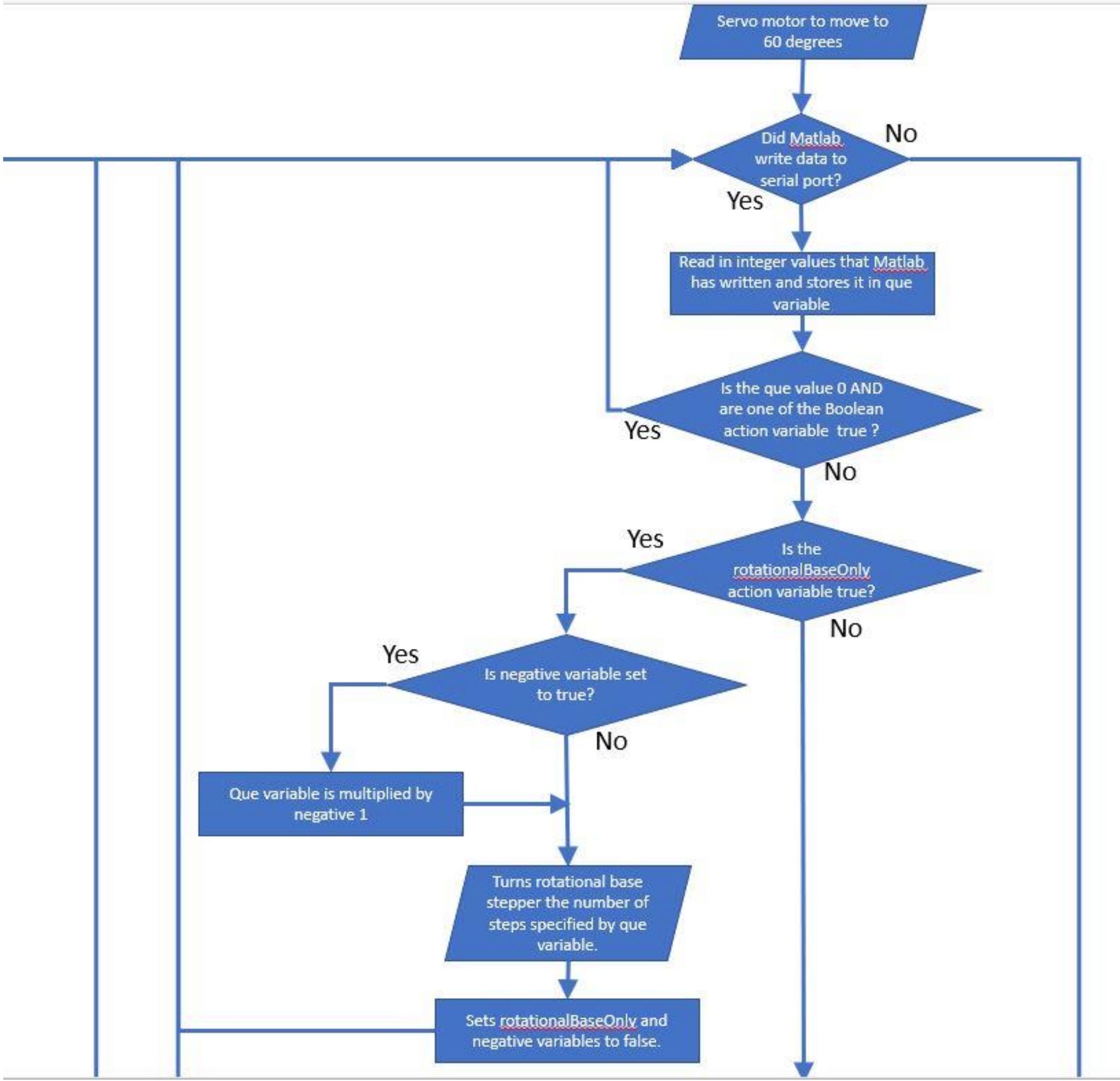
Flowchart for Water Sampler Control Code
(Arduino)



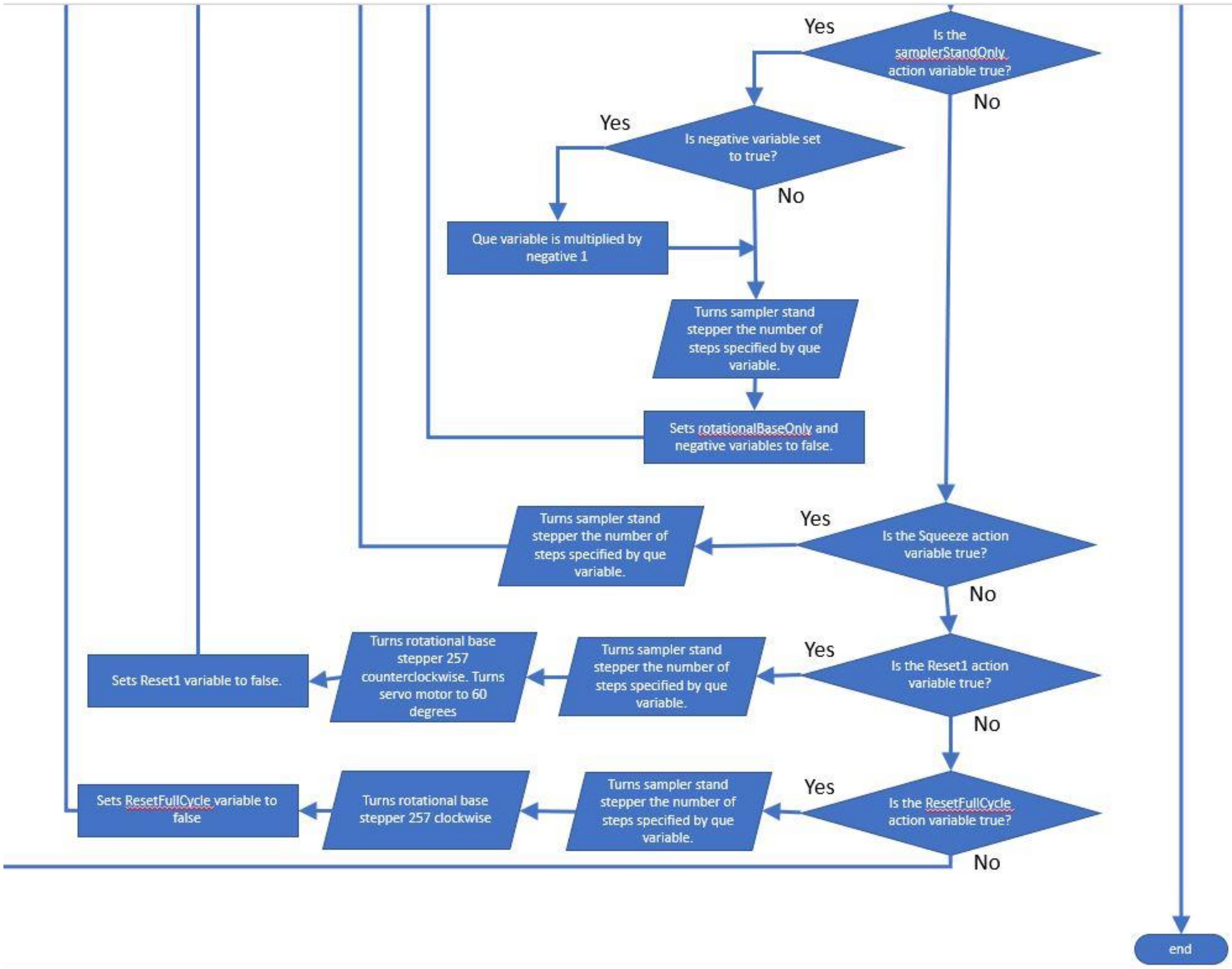
Arduino Flowchart (View 1)



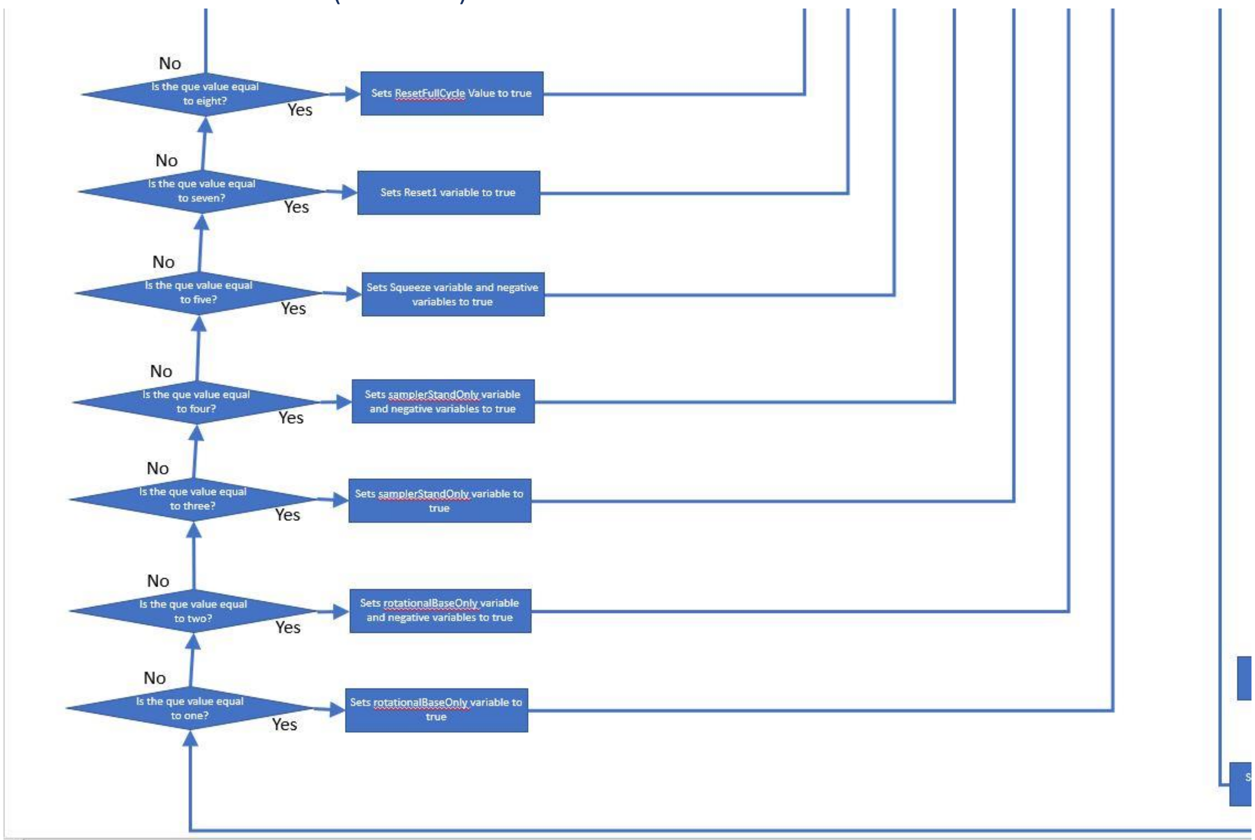
Arduino Flowchart (View 2)



Arduino Flowchart (View 3)

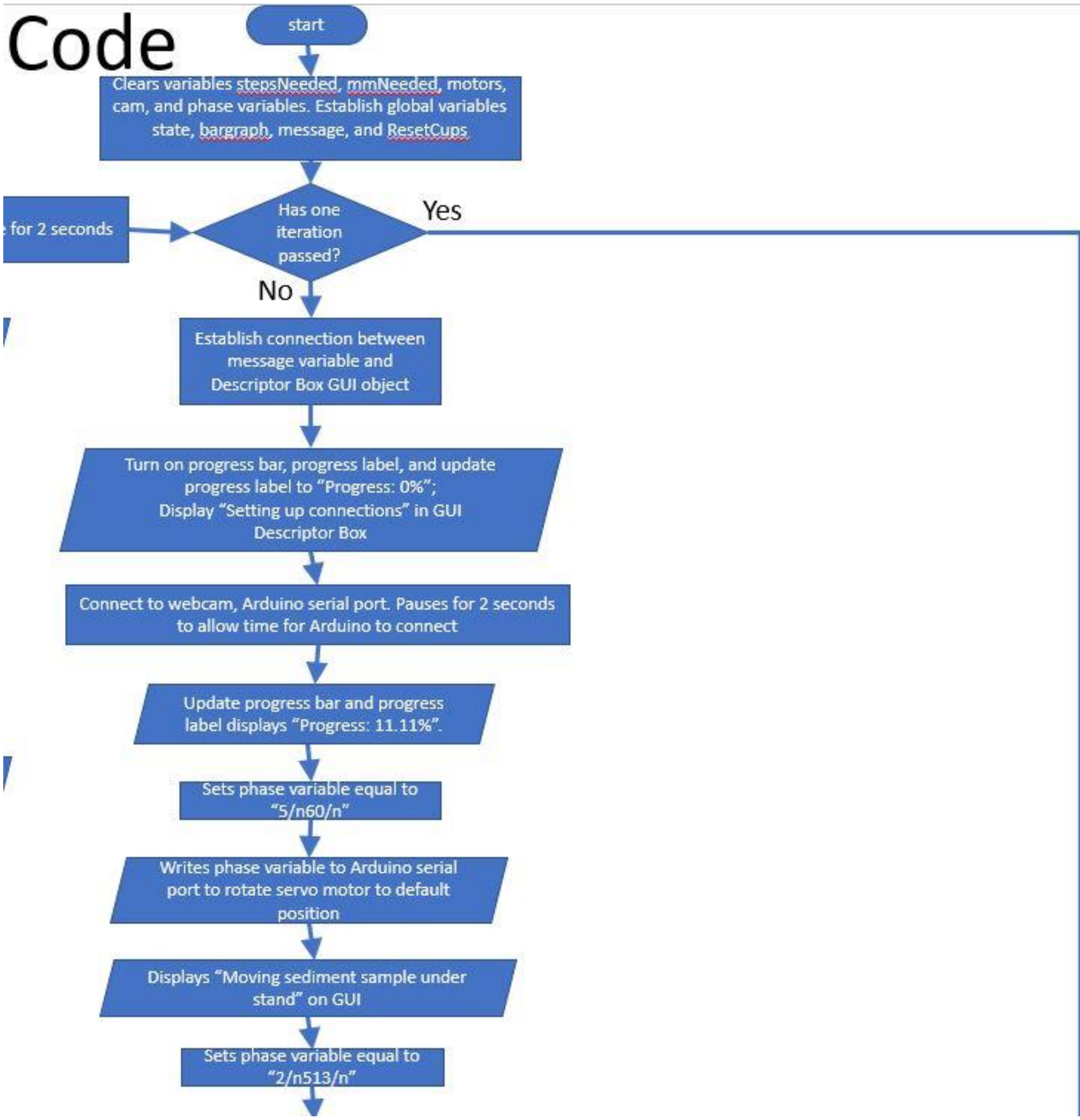


Arduino Flowchart (View 4)

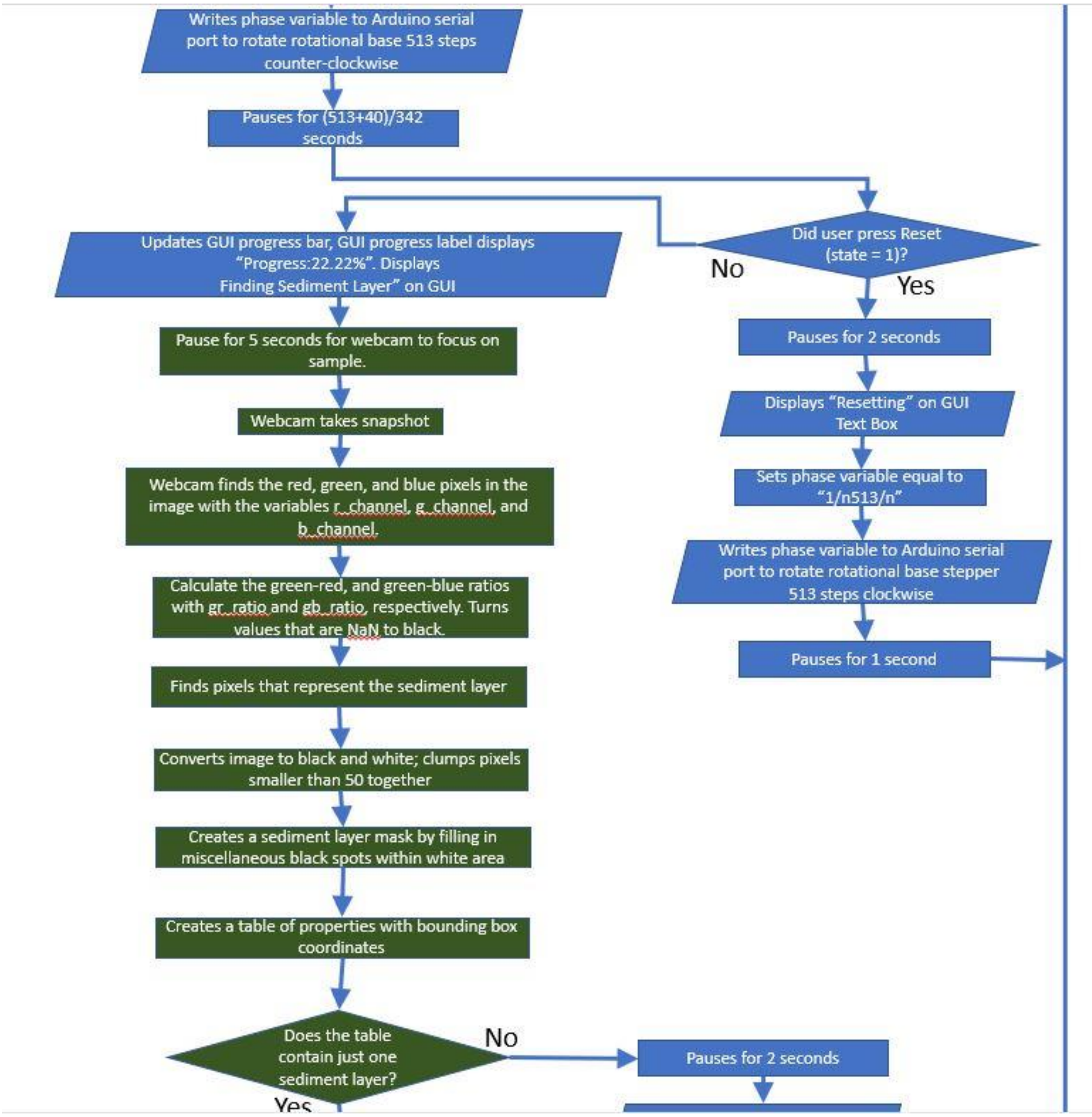


Control Code Flowchart (Matlab)

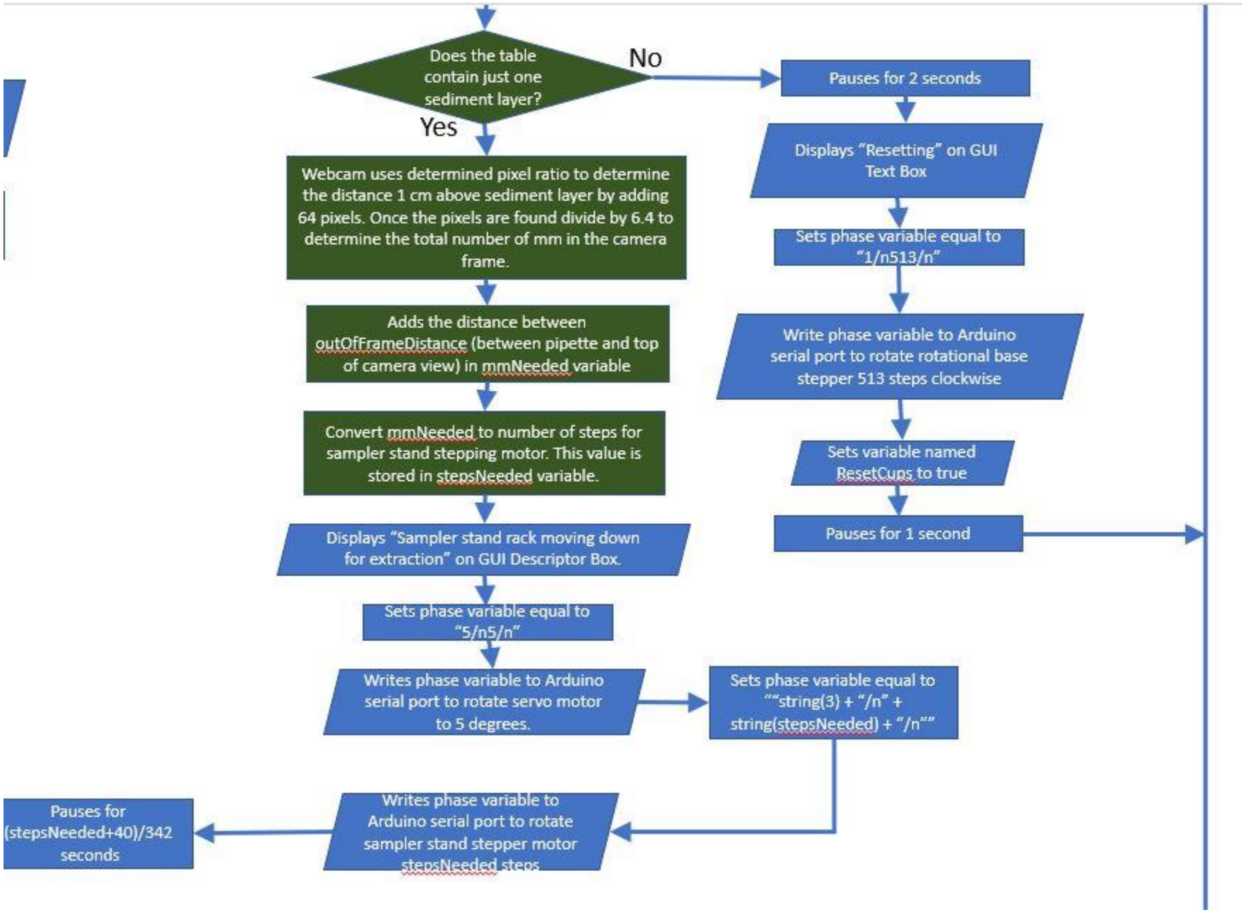
Control Code Flowchart (Matlab View 1)



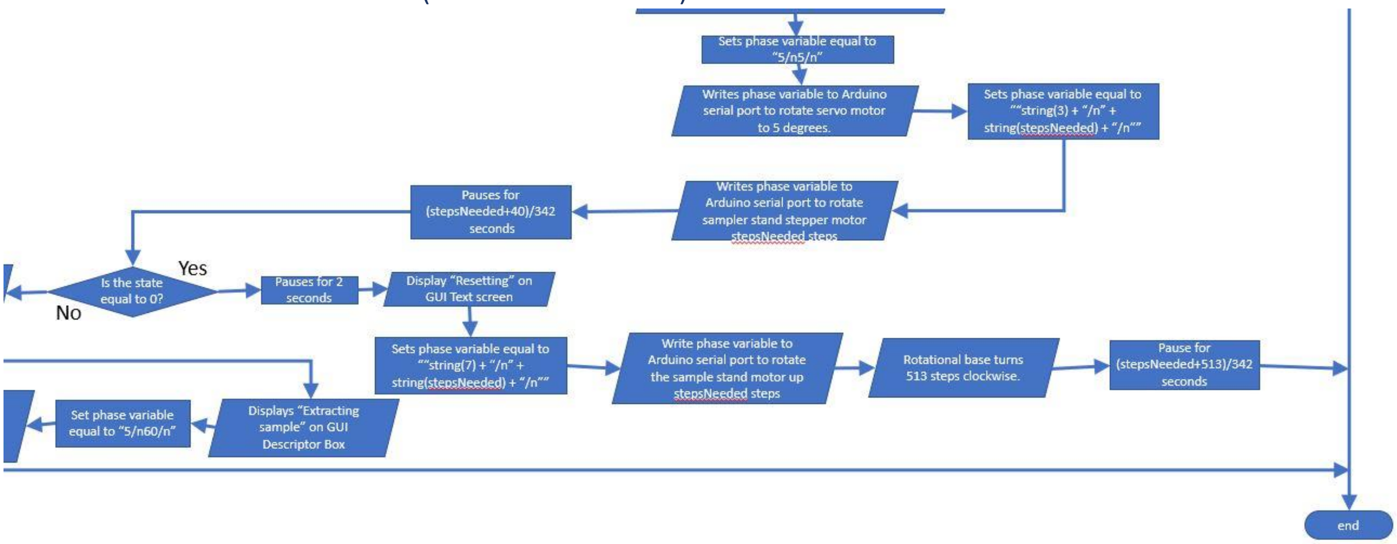
Control Code Flowchart (Matlab View 2)



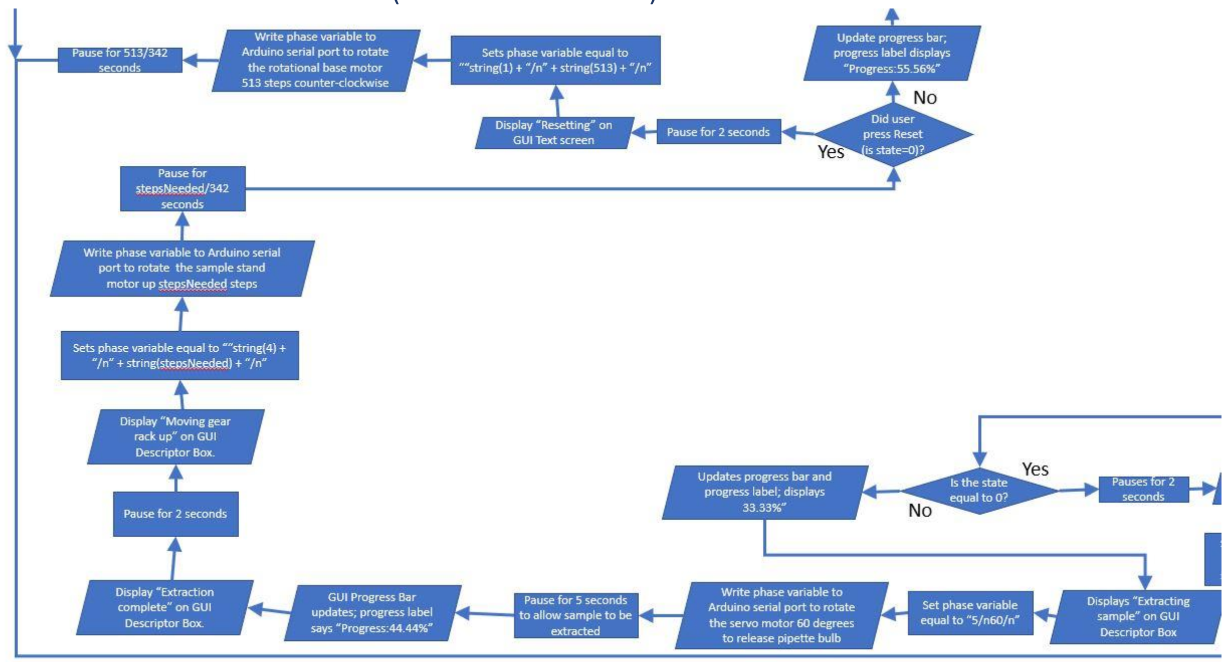
Control Code Flowchart (Matlab View 3)



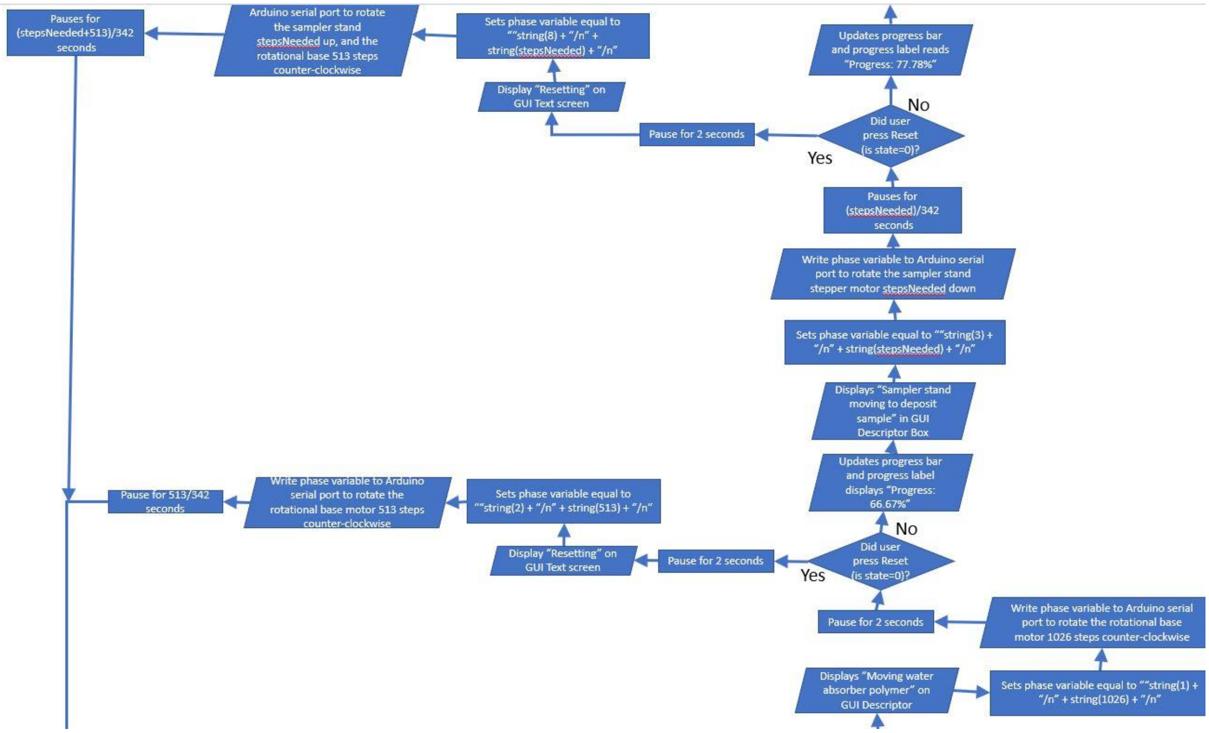
Control Code Flowchart (Matlab View 4)



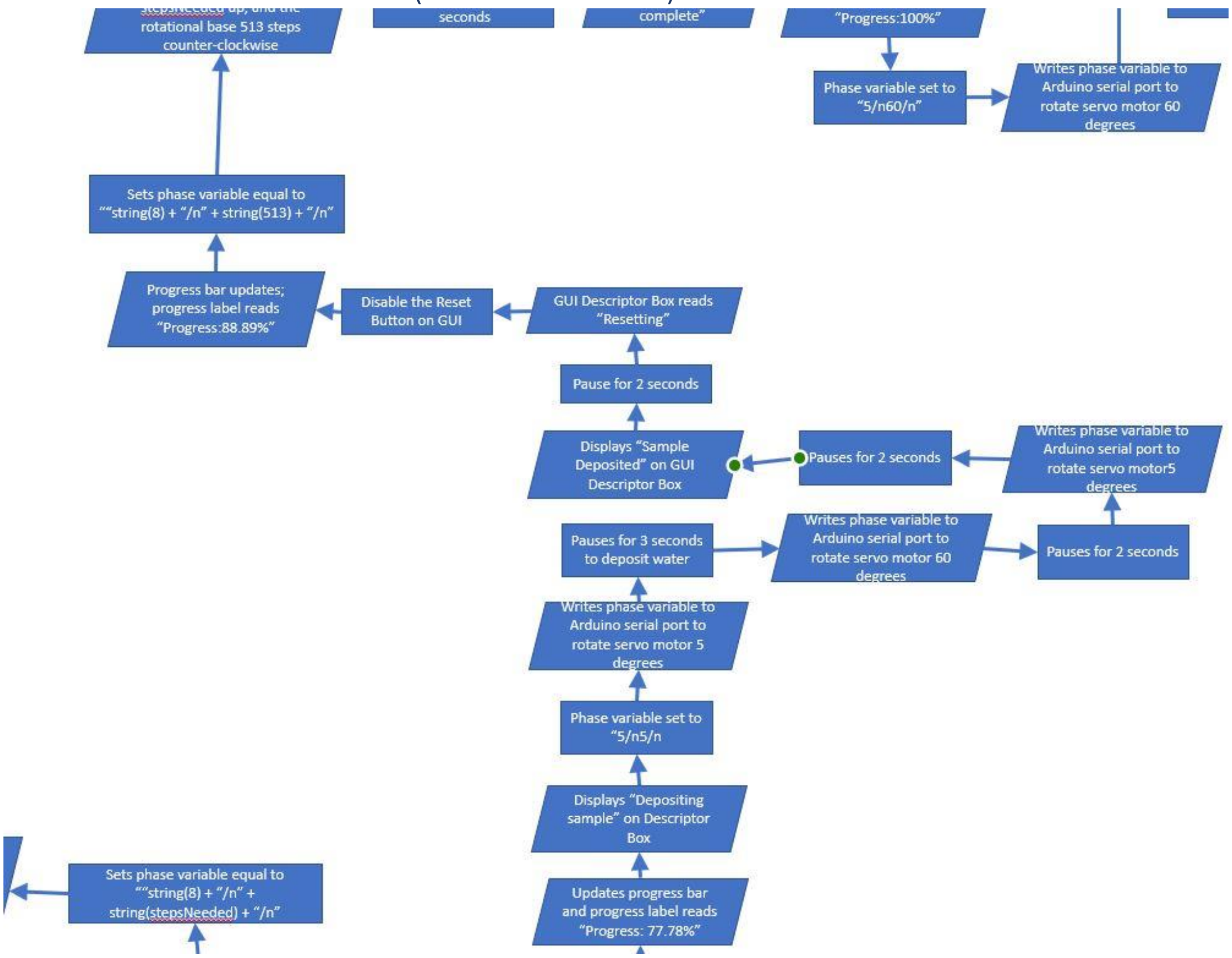
Control Code Flowchart (Matlab View 5)



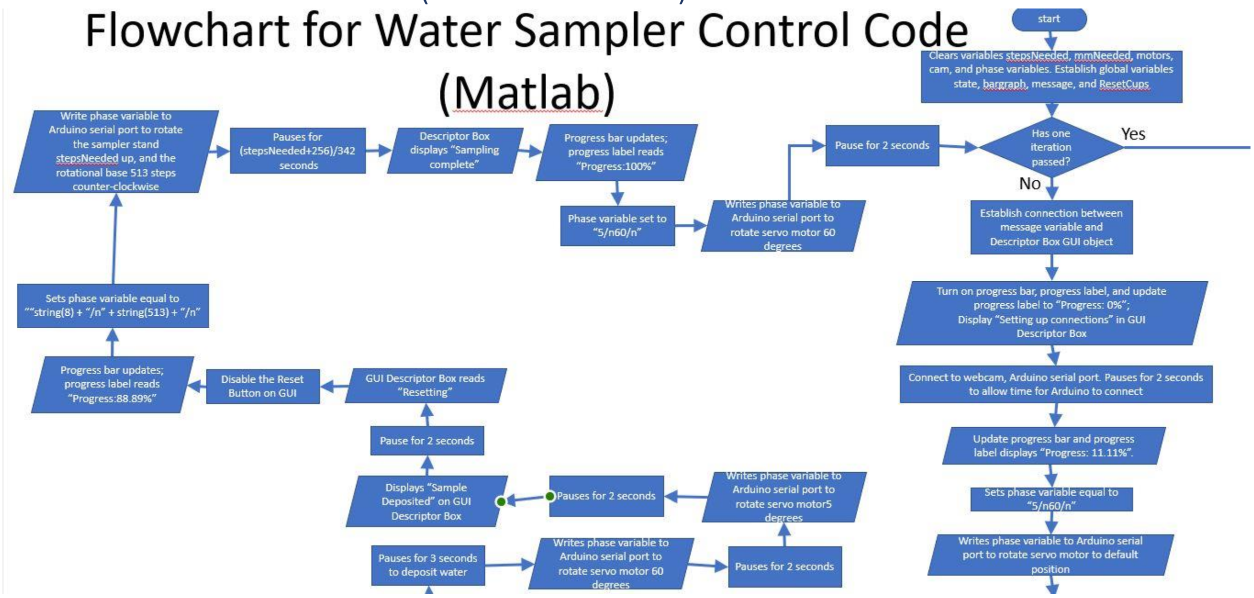
Control Code Flowchart (Matlab View 6)



Control Code Flowchart (Matlab View 7)



Control Code Flowchart (Matlab View 8)



Code

Vision System Code (Original)

```
%Garrett Matheny
%EGR 103
%03/12/21

originalIm = imread('Capture.jpg'); % This represents where
the snapshot will be taken
original = originalIm %This ensures that the original image
that is read will not be altered

g_channel = original(:,:,2); %finds green
r_channel = original(:,:,1); %finds red
b_channel = original(:,:,3); %finds blue

gr_ratio = double(g_channel)./ double(r_channel);
%calculates green/red ratio
gb_ratio = double(g_channel)./ double(b_channel);
%calculates green/blue ratio

%sets pixels that result as undefined to zero
gr_ratio(isnan(gr_ratio)) = 0;
gb_ratio(isnan(gb_ratio)) = 0;

%Detemines where the sediment layer and marker is.
%Note: The constraints on each bin need to be changed.
sediment_bin = ((gr_ratio <= .75) &(r_channel <= 60) &
(g_channel <= 40));
marker_bin = ((gr_ratio <= .45) &(r_channel <= 137) &
(g_channel <= 50));

%creates black and white images for sediment layer and
marker
blackAndWhite = bwareaopen(sediment_bin,50);
blackAndWhite2 = bwareaopen(marker_bin,50);

%Fills in any miscellaneous pixels that are white
```

```
sedimentMask = blackAndWhite >= 100;
sedimentMask = imfill(blackAndWhite,'holes');
markerMask = blackAndWhite2 >= 100;
markerMask = imfill(blackAndWhite2,'holes');

%Covers marker region with red color overlay
red_overlay = imoverlay(original,markerMask,[1,0,0]);

%covers sediment layer with green color overlay
layers = imoverlay(red_overlay,sedimentMask,[0,1,0]);

%creates a figure that shows where the sediment layers are
located.
figure(1)
imshow(layers)


%I didn't know how to get the values into one table, so I
found separate
%values for each mask and vertically concatenated them into
one.
TableOfPropSediment =
regionprops('table',sedimentMask,'BoundingBox');
TableOfPropMarker =
regionprops('table',markerMask,'BoundingBox');
TableOfObj = vertcat(TableOfPropSediment,TableOfPropMarker)

%convert the combined table into an array
usefulTable = TableOfObj{:,:};

%find the top left edge of the sediment layer
minValue = usefulTable(1,2)

%find the lowest part of the marker.  I added the location
of the top part
%of the marker to the height of the marker region.
maxValue = usefulTable(2,2) + usefulTable(2,4)

%Draws a line around the top of the sediment layer
j = round(minValue);
     original(j,:,2) = 1;
     original(j,:,3) = 1;

%Draws a line around the bottom of the marker
```

```
k = round(maxValue);
original(k,:,2) = 1;
original(k,:,3) = 1;

%displays the two lines and the original image.
figure(2)
imshow(original)
```

Vision System Code (Revised)

```
%{
VisionSystem - Determines the location of the sediment
layer so that the
pipette can be position 1 cm above sediment layer
Primary Author: Grant Eyer
Secondary Authors: Garrett Matheny, Dorian Lewis, Christian
Gratz, and
Joshua Hagan.
Assignment: EGR 103-015 Water Sampling Project (Control
Code)
Date created: 30 March 2021.
Notes:
*Important*: This code is dependent to the Water Sampler
GUI interface.
This script is not referenced directly within the Water
Sampler Control
function, but the essential functions of this script are
within the control
code.
Changes:
14 April 2021 - Deleted the while loop functionality, and
decided that
detection of the pipette tip was unnecessary.
Purpose: Helps detect the top of the sediment layer, this
is so that the
sampler stand stepper motor can determine the approximate
amount of steps
needed so that the pipette tip is approximately 1 cm above
the sedmient
layer.
24 April 2021- deleted lines pertaining to cup angle. Use
pixel/mm ratio to
autocalculate 1 cm above sediment layer.

%}

%connects with USB webcam named 'USB2.0 PC Camera'. This
would be done at
```

```
%the beginning of the WaterSamplerControl function
cam = webcam(2);


%Begin Vison System Process
originalIm = snapshot(cam);% This represents where the
snapshot will be taken
original = originalIm; %This ensures that the original
image that is read will not be altered

g_channel = original(:,:,2); %finds green
r_channel = original(:,:,1); %finds red
b_channel = original(:,:,3); %finds blue

gr_ratio = double(g_channel)./ double(r_channel);
%calculates green/red ratio
gb_ratio = double(g_channel)./ double(b_channel);
%calculates green/blue ratio

%sets pixels that result as NaN to zero
gr_ratio(isnan(gr_ratio)) = 0;
gb_ratio(isnan(gb_ratio)) = 0;

%Detemines where the sediment layer and marker is on the
basis of RGB
%analysis.
sediment_bin = (r_channel <= 65) & (gr_ratio <= .85);

%creates black and white images for sediment layer
blackAndWhite = bwareaopen(sediment_bin,50);

imshow(blackAndWhite)
%Fills in any miscellaneous pixels that are white
sedimentMask = blackAndWhite >= 100;
sedimentMask = imfill(blackAndWhite,'holes');

% Convert to pixels
pixelPerMM = 6.4;

%Stores the information about the sediment layer detected
into a table
TableOfPropSediment =
regionprops('table',sedimentMask,'BoundingBox');
```

```
%convert the table into an array
usefulTable = TableOfPropSediment{:,:};

%Otherwise, the webcam inds the highest part of the
sediment layer plus
%64 pixels (a centimeter)
minValue = usefulTable(1,2) + 64

%determines how the distance within the camera view in mm.
mmNeeded = minValue/pixelsPerMM;

out
```

Control Code (GUI setup and Callbacks)

```
classdef WaterSamplerGUI < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure              matlab.ui.Figure
        PauseButton           matlab.ui.control.Button
        Lamp                  matlab.ui.control.Lamp
        LampLabel             matlab.ui.control.Label
        WaterSamplerGUILabel  matlab.ui.control.Label
        Label                 matlab.ui.control.Label
        DescriptorBox         matlab.ui.control.EditField
        ResetButton           matlab.ui.control.Button
        StartButton           matlab.ui.control.Button
        UIAxes                matlab.ui.control.UIAxes
    end




    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            clear serialport arduino

            global bargraph pausebutton ResetCups %allows bargraph to be shared
between interface callbacks and control code.

            ResetCups = false;%sets reset cup message to false

            app.DescriptorBox.Value = 'Ready for sampling';  %Message that
displays when interface is open.

            set(app.ResetButton,'Enable','off'); %disables the stop button
            set(app.PauseButton,'Enable','off'); %disables the stop button

            %Sets the background of the box to yellow
            %disables user editability.
            set(app.DescriptorBox,'BackgroundColor','Yellow');
            set(app.DescriptorBox,'Editable','off');
```

```
            set(app.DescriptorBox,'FontWeight','bold');

            %Turns the toolbar visibility, Interactivity, visibility of
            %UIAxes off.
            set(app.UIAxes.Toolbar,'Visible','off');
            set(app.UIAxes,'Interactions',[]);
             app.UIAxes.Visible = 'off';

            %Lines 43 through 47 create a bar graph that looks like a
            %progress bar. Makes bar graph invisible. Sets the body color
            %of each of the bars to white, and the edges of each of the
            %bars a turquoise color.
            y = ones(1,9);
            bargraph =
bar(app.UIAxes,y,1,'FaceColor','flat','BarWidth',1,'PickableParts',"none");
            bargraph.CData(:,:) = ones(9,3);
            bargraph.EdgeColor = [0,.8,.8];
            bargraph.Visible = 'off';

            %Makes Progress Label initially invisible
            set(app.Label,'Visible','off');

            %Sets lamp label and color to let user know that they can close
            %the program.
            set(app.LampLabel,'Text','Can close GUI');
            set(app.Lamp,'Color','g');

            pausebutton = 1;

        end


% Button pushed function: StartButton
function StartButtonPushed(app, event)
global bargraph pausebutton state ResetCups %global variable that connect the
progress bar, state, and pausebutton
%variables to control code.
ResetCups = false;
%set state equal to one so that break statement are not engaged
state = 1;
%Changes label and color to let uder know that they can not close
%the GUI while running
set(app.LampLabel,'Text',"Can't close GUI")
set(app.Lamp,'Color','r')
%Disables the start button and enables the stop and pause buttons once user
%presses the start button.
set(app.StartButton,'Enable','off');
set(app.ResetButton,'Enable','on');
set(app.PauseButton,'Enable','on');
%executes Water Sampler Control Code.
```

```
WaterSamplerControl(app)
%disables the stop button and the enables the start button once
%the water sampler control code is exited.
set(app.ResetButton,'Enable','off');
set(app.StartButton,'Enable','on') ;
%Turns the body color and the visiblity of the bars off.
bargraph.CData(:,:) = ones(9,3);
bargraph.Visible = 'off';
%Turns the progress label off
set(app.Label,'Visible','off');
%Changes lamp label and color to let user know they can close
%the GUI.
set(app.LampLabel,'Text',"Can exit GUI");
set(app.Lamp,'Color','g');
%initializes pause button value
pausebutton = 1;
%sets the enability of the pause button off.
set(app.PauseButton,'Enable','off');
%if ResetCups is true, it tells user that they need to load cups in the correct
location.
%Otherwise,the default text is set to 'Ready for
%Sampling'
if ResetCups == true
app.DescriptorBox.Value = 'Please load samples in designated locations';
ResetCups = false;
else
app.DescriptorBox.Value = 'Ready for Sampling';
end
end


% Button pushed function: ResetButton
function ResetButtonPushed(app, event)
global state bargraph
%Once the value is triggered, the state variable is set to 0,
%meaning that the program will eventually stop.
state = 0;
%sets the enability of the stop button off
set(app.ResetButton,'Enable', 'off')
%Turns the Descriptor box message to 'Please wait'
app.DescriptorBox.Value = 'Please wait';
%clears global variable message
clear global message
%turns the progress bar and label off
set(bargraph,'Visible','off')
set(app.Label,'Visible','off')
end


% Close request function: UIFigure
```

```
function UIFigureCloseRequest(app, event)
global pausebutton
% if the pause button is pressed or when the control code is
% not running, the user can't exit the GUI.
if strcmpi(app.DescriptorBox.Value,'Ready for Sampling') | (mod(pausebutton,2) ==
0) | strcmpi(app.DescriptorBox.Value,'Please load samples in designated
locations')
%deletes app and clears associated global variables
clear global message
clear global bargraph
clear global pausebutton
delete(app)
else
return %return to Start Button function
end
end


% Button pushed function: PauseButton
function PauseButtonPushed(app, event)
global pausebutton message bargraph
%lets user know program is paused
message.Value = 'Program paused:Press Again to Continue';
%Changes lamp label and color to let user know they can close
%the GUI.
set(app.LampLabel,'Text',"Can exit GUI");
set(app.Lamp,'Color','g');
%turns the progress bar and label off
set(bargraph,'Visible','off')
set(app.Label,'Visible','off')
%sets the enability of the stop button off
set(app.ResetButton,'Enable', 'off')
%adds 1 to original pausebutton value
pausebutton = pausebutton + 1;
%while the pause button value is even, this while loop pauses
%the program.
while (mod(pausebutton,2) == 0)
pause(1) %pauses for one second
%pausebutton;
end
%turns the progress bar and label off
set(bargraph,'Visible','on')
set(app.Label,'Visible','on')
%sets the enability of the stop button off
set(app.ResetButton,'Enable', 'on')
end
end


% Component initialization
```

```
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 804 543];
app.UIFigure.Name = 'MATLAB App';
app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest,
true);

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
app.UIAxes.XColor = 'none';
app.UIAxes.XTick = [];
app.UIAxes.YColor = 'none';
app.UIAxes.ZColor = 'none';
app.UIAxes.Position = [229 355 348 67];

% Create StartButton
app.StartButton = uibutton(app.UIFigure, 'push');
app.StartButton.ButtonPushedFcn = createCallbackFcn(app, @StartButtonPushed,
true);
app.StartButton.BackgroundColor = [0 1 0];
app.StartButton.FontSize = 30;
app.StartButton.FontWeight = 'bold';
app.StartButton.Position = [229 180 121 51];
app.StartButton.Text = 'Start';

% Create ResetButton
app.ResetButton = uibutton(app.UIFigure, 'push');
app.ResetButton.ButtonPushedFcn = createCallbackFcn(app, @ResetButtonPushed,
true);
app.ResetButton.BackgroundColor = [1 0 0];
app.ResetButton.FontSize = 30;
app.ResetButton.FontWeight = 'bold';
app.ResetButton.Position = [455.5 180 122 51];
app.ResetButton.Text = 'Reset';

% Create DescriptorBox
app.DescriptorBox = uieditfield(app.UIFigure, 'text');
app.DescriptorBox.HorizontalAlignment = 'center';
app.DescriptorBox.FontSize = 30;
```

```
app.DescriptorBox.FontWeight = 'bold';
app.DescriptorBox.BackgroundColor = [1 1 0];
app.DescriptorBox.Position = [61 270 688 74];


% Create Label
app.Label = uilabel(app.UIFigure);
app.Label.HorizontalAlignment = 'center';
app.Label.FontSize = 16;
app.Label.Position = [248 421 317 37];


% Create WaterSamplerGUILabel
app.WaterSamplerGUILabel = uilabel(app.UIFigure);
app.WaterSamplerGUILabel.HorizontalAlignment = 'center';
app.WaterSamplerGUILabel.FontSize = 40;
app.WaterSamplerGUILabel.FontWeight = 'bold';
app.WaterSamplerGUILabel.Position = [86 494 640 50];
app.WaterSamplerGUILabel.Text = 'Water Sampler GUI';


% Create LampLabel
app.LampLabel = uilabel(app.UIFigure);
app.LampLabel.HorizontalAlignment = 'center';
app.LampLabel.Position = [638 468 94 22];
app.LampLabel.Text = 'Lamp';


% Create Lamp
app.Lamp = uilamp(app.UIFigure);
app.Lamp.Position = [740 465 28 28];


% Create PauseButton
app.PauseButton = uibutton(app.UIFigure, 'push');
app.PauseButton.ButtonPushedFcn = createCallbackFcn(app, @PauseButtonPushed,
true);
app.PauseButton.BackgroundColor = [1 0.4118 0.1608];
app.PauseButton.FontSize = 30;
app.PauseButton.FontWeight = 'bold';
app.PauseButton.Position = [349 98 108 58];
app.PauseButton.Text = 'Pause';


% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end
```

```
% App creation and deletion
methods (Access = public)

% Construct app
function app = WaterSamplerGUI

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)

if nargout == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

Control Code (Matlab Script)

```
%{
WaterSamplerControl - Extracts and deposits sewage water
samples for
COVID-19 analysis.
Primary Author: Grant Eyer
Secondary Authors: Garrett Matheny, Dorian Lewis, Christian
Gratz, and
Joshua Hagan.
Assignment: EGR 103-015 Water Sampling Project (Control
Code)
Date created: 3 April 2021.
Notes:
*Important*: This code is dependent to the Water Sampler
GUI interface.
Changes:
On 3 April 2021 - linked interface with control code; began
implementing
how program will influnece the appearance of the Water
Sampler GUI.
On 15 April 2021 - Added in the functions that related to
connecting the
arduino serial ports, servo motor, webcam, and stepper
motors.
On 16 April 2021- After realizing that the motors were
acting
asynchronously, the matlab servo object was deleted and was
placed along with
the stepper motor objects in the Arduino IDE file labeled
"WaterSamplerMotor2".
On 18 April 2021 - Added in a break statement before
sediment layer
analysis (lines 88-95)
On 20 April 2021 - motor steps were changed from 342 to
257.
Purpose: User places a sample containing water contaminated
with COVID-19
```

```
and an empty water-absorbing polymer in their designated
locations as
listed on the Water Sampler machine.  The program rotates
the container
with the contaminated water and extracts about 1 mL of
water. The program
uses a webcam to ensure that the extraction takes place
about 1 cm above
the sediment layer.  Once extracted, the rotational base
moves the water
absorbing polymer to deposit the water sample for further
analysis. The
program then resets the structure for any other
contaminated samples that
need to be tested.
On April 24, 2021 - changed vision system portion of
control code (no longer
(determines angle tolerance).
%}
function WaterSamplerControl(app)
%clears rotational base stepper motor, sampler stand
stepper motor,webcam,
%, servo motor, mmNeeded,and stepsNeeded variables
clear stepsNeeded mmNeeded motors cam phase
global state bargraph message ResetCups %connects vaiables
between WaterSampler GUI and control code.

%the iteration is set to one so that the process executes
only once.
for iteration = 1

%establish a connection with the app Descriptor Box
message = app.DescriptorBox;

%Progress bar and progress label are made visible on
WaterSampler GUI.
    bargraph.Visible = 'on';
    app.Label.Visible = 'on';
    app.Label.Text = 'Progress: 0%';

%Lets user know that program is connecting to arduino and
its components
    message.Value = 'Setting up connections';
```

```
 %Connects to arduino interface, as well as stepper motors,
servo
 %motor, and webcam.

cam = webcam(2);
motors = serialport('COM4',9600);

%pauses for 2 seconds to allow serial port to connect.
pause(2)

%Updates GUI progress bar and label
    bargraph.CData(1,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 11.11%';

    %sets servo motor to default position at 60 degrees
    phase = "5/n60/n";
    write(motors,phase,"string");

    %Rotational base moves clockwise to position the
sediment sample under
    %the stand.
    message.Value = 'Moving sediment sample under stand';

    %indicates that que 1 is required, followed by the
steps needed for the
    %rotational base to move
     phase = "2/n513/n";
     write(motors,phase,"string");

     %pauses so that arduino can catch up with Matlab
visuals
     pause(513/342);

%if user presses the reset button on the GUI, initially
pause for two seconds
%let user know that the machine is resetting, then the
rotational base moves
%clockwise to set back to default position. Pauses again so
that arduino
%can catch up with Matlab.
    if state == 0
       pause(2);
       app.DescriptorBox.Value = "Resetting";
       phase = string(1) + "/n" + string(513)+ "/n";
```

```
        write(motors,phase,"string");
        pause(513/342);
         break;
    end

%Update the progress bar and label once sampler stand has
moved pipette the
%specified number of steps.
    bargraph.CData(2,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 22.22%';

 %Begin Vison System Process

 message.Value = "Finding sediment layer";
 %gives camera time to focus on sample
 pause(5);

%Take a snapshot of the sediment layer cup
originalIm = snapshot(cam);

g_channel = originalIm(:,:,2); %finds green
r_channel = originalIm(:,:,1); %finds red
b_channel = originalIm(:,:,3); %finds blue

gr_ratio = double(g_channel)./ double(r_channel);
%calculates green/red ratio
gb_ratio = double(g_channel)./ double(b_channel);
%calculates green/blue ratio

%sets pixels that result as NaN to zero
gr_ratio(isnan(gr_ratio)) = 0;
gb_ratio(isnan(gb_ratio)) = 0;

%Detemines where the sediment layer and marker is on the
basis of RGB
%analysis.
sediment_bin = (r_channel <= 65) & (gr_ratio <= .85);

%creates black and white images for sediment layer
blackAndWhite = bwareaopen(sediment_bin,50);

%Fills in any miscellaneous pixels that are white
sedimentMask = blackAndWhite >= 100;
sedimentMask = imfill(blackAndWhite,'holes');
```

```
%pixels per mm
pixelPerMM = 6.4;

%Stores the information about the sediment layer detected
into a table
TableOfPropSediment =
regionprops('table',sedimentMask,'BoundingBox');

%convert the table into an array
usefulTable = TableOfPropSediment{:,:};

%if the table is empty or if it detected multiple "sediment
layers", initially
%pause for two seconds. Then,the GUI displays "resetting"
so that the user
%understands that the program is resetting. Turn rotational
base clockwise
%to its default position and pause again so that user knows
when the
%control code execution ends.
if isempty(usefulTable) | (numel(usefulTable) > 4)
    pause(2);
    app.DescriptorBox.Value = "Resetting";
    phase = "1/n513/n";
    write(motors,phase,'string');
    ResetCups = true;
    pause(513/342);
    break; %would get out of the for loop
end

%Otherwise, the webcam inds the highest part of the
sediment layer plus
%pixel tolerance.
pixels = usefulTable(1,2) - 64;

%determines the in-frame height in mm.
mmNeeded = pixels/pixelPerMM;
%Vision system ends

%represents distance from pipette tip to the top of the
camera field of
%view
outOfFrameDistance = 30;
```

```
%determines the total distance needed to travel in mm.
mmNeeded = mmNeeded + outOfFrameDistance;

%converts mm into steps for the sampler stand stepper
motor.
stepsNeeded = round(4.2859 * mmNeeded);

%lets user know that the sampling stand is moving pipette
down.
message.Value = 'Sampler stand rack moving down for
extraction';

%tells arduino to rotate the servo motor 15 degrees
 phase = "5/n5/n";
 write(motors,phase,"string");

%rotate stand stepper motor as specified by stepsNeeded.
phase = string(4) + "/n" + string(stepsNeeded) + "/n";
write(motors,phase,"string");

pausetime = (stepsNeeded+40)/342;
pause(pausetime);

%if user presses the reset button on water sampler GUI,
initially pause for
%two seconds. Displays "Resetting" on GUI to let user know
sampler machine is
%resetting. The sampler stand and rotational base stepper
motors, and the
%stepper motor reset. Program pauses to ensure that Matlab
visuals match
%with actions of the Water Sampler.
    if state == 0
        pause(2);
        app.DescriptorBox.Value = "Resetting";
        phase = string(7) + "/n" + string(stepsNeeded) +
"/n";
        write(motors,phase,"string");
        pausetime = (stepsNeeded+513)/342;
        pause(pausetime);
        break;
    end
```

```
 %Progress bar and label updates
    bargraph.CData(3,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 33.33%';

%Lets user know that extraction of water is underway
message.Value = 'Extracting Sample';

%servo motor releases to allow extraction
 phase = "5/n60/n";
 write(motors,phase,"string");

pause(5); %pause for five seconds to ensure that sample is
extracted.

%progress bar and label updates
    bargraph.CData(4,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 44.44%';

 %Lets user know extraction is complete on GUI
message.Value = 'Extraction Complete';

%pauses so that user can note that extraction is complete
 pause(2);

%text that displays on Water Sampler GUI updates
message.Value = 'Moving gear rack up';

%sample stand goes up
 phase = string(3) + "/n" + string(stepsNeeded) + "/n";
 write(motors,phase,"string");
 pausetime = stepsNeeded/342;
 pause(pausetime);

%if user presses the reset button on water sampler GUI,
initially pause for
%two seconds. Let user know water sampler machine is
resetting and move the
%rotational base to its default position.
    if state == 0
        pause(2);
        app.DescriptorBox.Value = "Resetting";
       phase = string(1)+ "/n" + string(513)+"/n";
       write(motors,phase,"string");
      pause(513/342);
```

```
        break;
    end

%progress bar and label update
    bargraph.CData(5,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 55.56%';


%Lets user know that the rotational base will rotate the
water absorbing
%polymer under the stand).
message.Value = 'Moving water abosrbing polymer under
stand';

%rotational base moves counter-clockwise
   phase = string(2) + "/n" + string(1026)+"/n";
   write(motors,phase,"string");

%pauses so that arduino actions can line up with matlab
visuals
   pause(1026/342);

%if user presses the reset button on the GUI, initially
pause for two seconds.
%lets user know that the machine is resetting. The
rotational base moves
%counter-clockwise to set back to default position. Pauses
to ensure that arduino
%action matches Matlab visuals
    if state == 0
        pause(2);
        app.DescriptorBox.Value = "Resetting";
       phase = string(2) + "/n" + string(513)+ "/n";
       write(motors,phase,"string");
       pause(513/342);
        break;
    end

%updates progress bar and progress label on Water Sampler
GUI
    bargraph.CData(6,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 66.67%';
```

```
%Lets user know that the stand is moving down to deposit
sample
message.Value = 'Sampler stand moving to deposit sample';

    %sampler stand moves down stepsNeeded distance
    phase = string(4) + "/n" + string(stepsNeeded)+"/n";
    write(motors,phase,"string");

    %pauses so that arduino can catch up with matlab
visuals
   pausetime = stepsNeeded/342;
   pause(pausetime);

%if user presses the reset button on the GUI, let user know
that the machine
%is resetting. The sampler stand moves back up stepsNeeded,
and after two
%seconds, the rotational base moves counter-clockwise to
set back to default position
    if state == 0
        pause(2);
        app.DescriptorBox.Value = "Resetting";
        phase = string(8) + "/n" + string(stepsNeeded) +
"/n" ;
        write(motors,phase,"string");
        pausetime = (stepsNeeded+513)/342;
        pause(pausetime);
        break;
    end

% updates progress bar and label on Water Sampler GUI.
    bargraph.CData(7,:) = [0,.8,.8];
    app.Label.Text = 'Progress: 77.78%';

%Lets user know that the pipette is depositing sample on
Water Sampler GUI
message.Value = 'Depositing sample';

%servo turns 15 degrees to squeeze pipette to deposit
sample. The sample
%pauses for three seconds.
phase = "5/n5/n";
write(motors,phase,"string");
```

```
pause(2); %pauses two seconds

%turns servo motor to 60 degrees
phase = "5/n60/n";
write(motors,phase,"string");

pause(2);%pauses two seconds

%turns servo motor to 5 degrees to squeeze any additonal
liquid
phase = "5/n5/n";
write(motors,phase,"string");

pause(3);
%lets user know that the sample has been completely
deposited on Water Sampler
%GUI.
message.Value = 'Sample deposited';

%pauses for two seconds to confirm that user saw deposit
sample
%confirmation message.
 pause(2)

%Lets user know that stand is resetting for the next
sampling
message.Value = 'Resetting';

%Disables the stop button so that the user can not press
stop. At this
%point, the user can not reset the program, as by this part
of the program
%forward automatically resets.
set(app.ResetButton,'Enable','off');

%updates progress bar and progress label on Water Sampler
GUI.
 bargraph.CData(8,:) = [0,.8,.8];
 app.Label.Text = 'Progress: 88.89%';

%sampler stand moves back up number of steps specified by
stepsNeeded
phase = string(8) + "/n" + string(stepsNeeded)+"/n";
write(motors,phase,"string");
```

```
%pause so that arduino actions line up with Matlab visuals
pausetime = (stepsNeeded+513)/342;
pause(pausetime);

%GUI DescriptorBox is updated with this message
message.Value = 'Sampling complete';

%progress bar and progress label updates on GUI
app.Label.Text = 'Progress: 100%';
bargraph.CData(9,:) = [0,.8,.8];

%turns servo motor to 60 degrees
phase = "5/n60/n";
write(motors,phase,"string");

%pauses for 2 seconds so that the user is aware that the
process has
%completed.
pause(2)

end

 end
```

Control Code (Original Arduino Sketch)

```
#include <Stepper.h> //stepper motor library


const int stepsPerRevolution = 2052; //represents 2052 steps/revolution for stepper motor

int rotate = 0; //intializes rotational variable


//establishes stepper motor object.On Line Driver, the pin numbers are in the following order: 1,3,2,4

Stepper myStepper = Stepper(stepsPerRevolution, 13,11,12,8);


void setup() {

  Serial.begin(9600); //establishes baud rate

   myStepper.setSpeed(10); //sets speed of stepper motor to 10 revolutions per minute (rpm)


}

void loop() {

//read information Matlab has written in serial port

while (Serial.available()) {


  rotate = Serial.parseInt();//reads in values that are integers

  myStepper.step(rotate); //rotates the stepper motor the number of steps specified in the "rotate" variable

  rotate = 0;//resets the rotate variable once the motor completes turning

  }
```

Control Code (Final Arduino Sketch)

```
#include <Servo.h> //Servo motor library
#include <stdlib.h>//Serial communication library
#include <Stepper.h>//Stepper motor library

Servo pipetteSqueezer; //creates servo motor object

/*Creates stepper motor objects for the rotational base and sampler stand. The line driver pins
for each of the stepper motor objects are
are in the following order: 1,3,2,4.*/
const int stepsPerRevolution = 2052; //2052 steps/revolution for sampler stand


Stepper rotationalBase = Stepper(stepsPerRevolution,13,11,12,8); //digital pins 13, 11, 12, and
8
Stepper samplerStand = Stepper(stepsPerRevolution,7,5,6,3);//digital pins 7,5,6,and 3


int que; //initializes que action value
boolean Squeeze = false;
boolean negative = false; //variable that tells arduino whether a value is negative
boolean rotationalBaseOnly = false; //variable that tells only the rotational base needs to move
boolean samplerStandOnly = false; //variable that tells that only the sampler stand should
move
boolean Reset1 = false; // indicates that the first reset option is desired
boolean ResetFullCycle = false; // variable that tells both sampler stand and rotational base to
move back to default positions, assuming no interruptions

/* setup - this action occurs when the arduino is intially plugged in*/
void setup()
{
  Serial.begin(9600);// Set up serial communications

//Sets speed of rotational base and sample Stand stepper motors to 10 revolutions per minute
(rpm).
  rotationalBase.setSpeed(10);
```

```
  samplerStand.setSpeed(10);

//specifies that servo motor is connected to digital pin 9, intializes servo motor to 0 degrees
  pipetteSqueezer.attach(9);
 pipetteSqueezer.write(60);

}
/* Setup - end*/


/* Main Loop - Begin*/
void loop()
{
// void loop is empty

}

/* Main Loop - End */


/* Communication Functions - Begin*
/*SerialEvent()is called after each loop if there is serial data
  available.  It is not necessary to include the call in the loop()
*/
void serialEvent() {
 while(Serial.available()> 0) //if there is data in the serial port, this function reads it and enters
into a while loop.
 {
    // get the new byte:

   que = Serial.parseInt(); //reads in positive integer values

  /*This statement is set up to ensure that the cues register properly.  Ensures that no
miscellaneous "0" values enter into
  the system */
      if ((que == 0) & (rotationalBaseOnly | samplerStandOnly | Reset1 | ResetFullCycle))
```

```
      {
      break;
      }

//once the cue is registered, enter the function processInput() to assess the que value and its meaning.
      else
      {
     processInput();
      }
   }
}


void processInput() {//processes que action values values

//if the only the rotational base is supposed to move, execute the following
  if (rotationalBaseOnly)

    {//if value was found to be negative, convert read in integer to negative value
      if (negative)
      {
        que = que * -1;
      }

    //rotates the stepper motor the specified number of steps
    rotationalBase.step(que);//

  //once rotational base is complete
    rotationalBaseOnly = false;//means do not execute motor function for next number
    negative = false;//turns the negative value to false
    }


    else if (samplerStandOnly)
    {
```

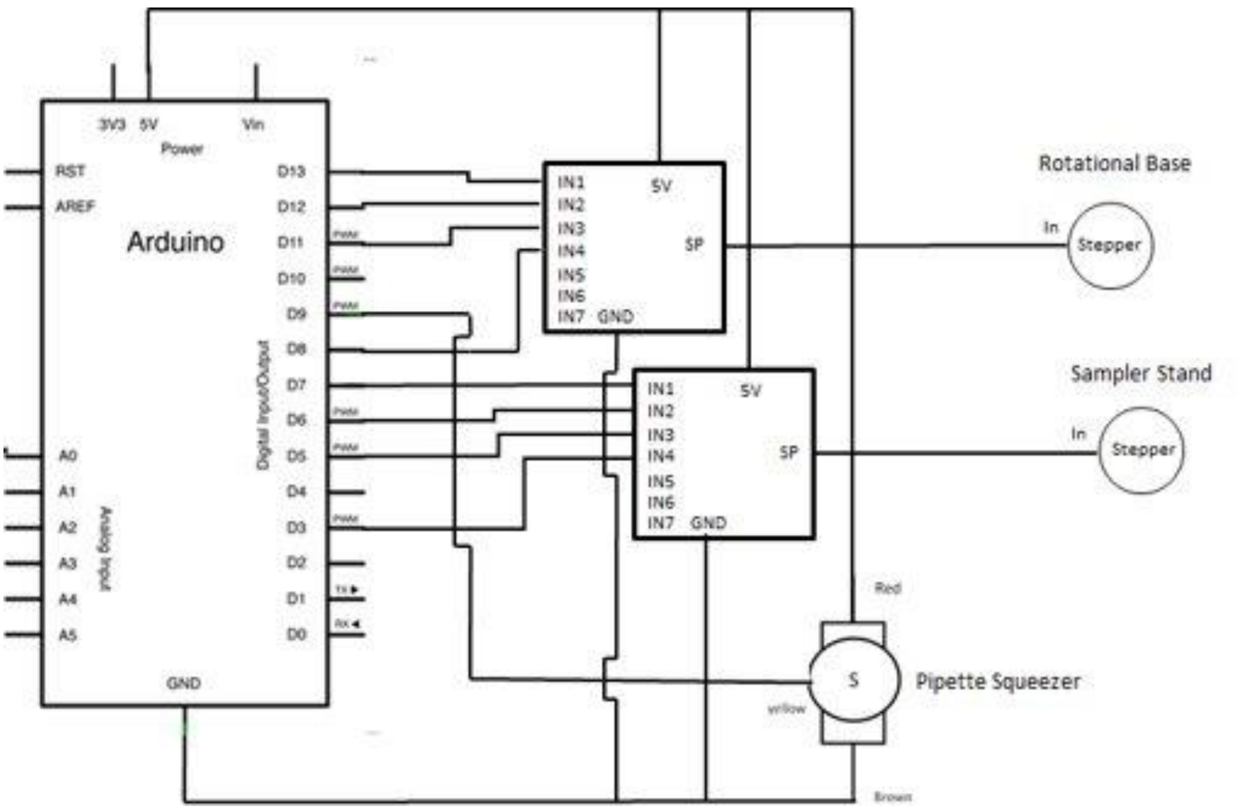
```
   if (negative)
   {//if value was found to be negative, convert read in integer to negative value
     que = que * -1;
   }
  samplerStand.step(que);
  samplerStandOnly = false;//means do not execute motor function for next number
  negative = false;//turns the negative value to false
 }

 else if (Squeeze)
{
 pipetteSqueezer.write(que);
 Squeeze = false;
}


else if (Reset1)
 {

 samplerStand.step(que); //sampler stand goes back up
 rotationalBase.step(513); //rotational base rotates counterclockwise 342 steps
 pipetteSqueezer.write(60); //sets servo motor to default position
 Reset1 = false;//sers Reset1 to false so that it does not accidently run again
 }


 else if (ResetFullCycle)
 {

 samplerStand.step(que); //sampler stand goes back up
 rotationalBase.step(-513); //rotational base rotates counter-clockwise 342
 ResetFullCycle = false;//sets ResetFullCycle option to false so that it does not accidently run
again.
 }
```

```
//these below statements execute commands, as well as interpret what to do with successive
information from the serial port.


   else if (que == 1) //cues action from the rotational base only (movement counter-clockwise);
   {
    rotationalBaseOnly = true;
   }

   else if (que == 2)//rotational base only; turns rotational base clockwise (negative steps)
   {
   rotationalBaseOnly = true;
   negative = true;//tells arduino that incoming number is negative
   }

   else if (que == 3)
   {
   samplerStandOnly = true; //sample stand needs to move down
   }


   else if (que == 4)  //indicate the sampler stand need to move down, indicates that the
number of steps will be negative
  {
  samplerStandOnly = true;
  negative = true;//tells arduino that incoming number is negative
  }

  else if (que == 5)
  {
  Squeeze = true;   //Servo motor turns 45 degrees to deposit sample, or to squeeze pipette
before extraction
  }


 else if (que == 7)  //resets cycle half-way
```

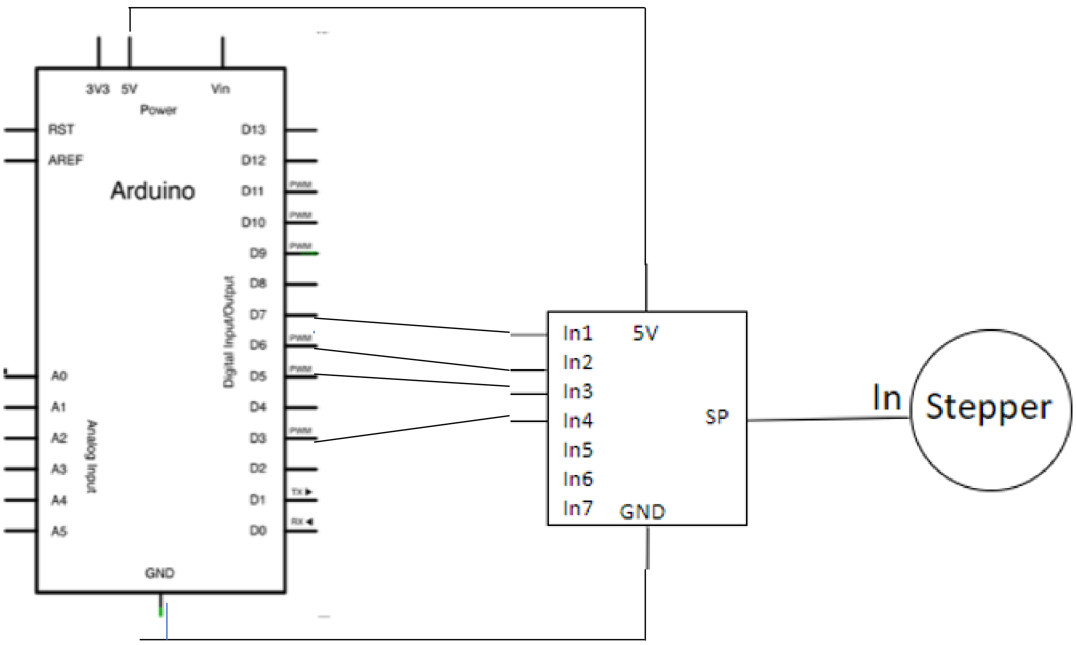
```
 {
 Reset1 = true;
 }

 else if (que == 8)   //Completely resets cycle
 {
ResetFullCycle = true;
 }

}
```

Electrical Schematics

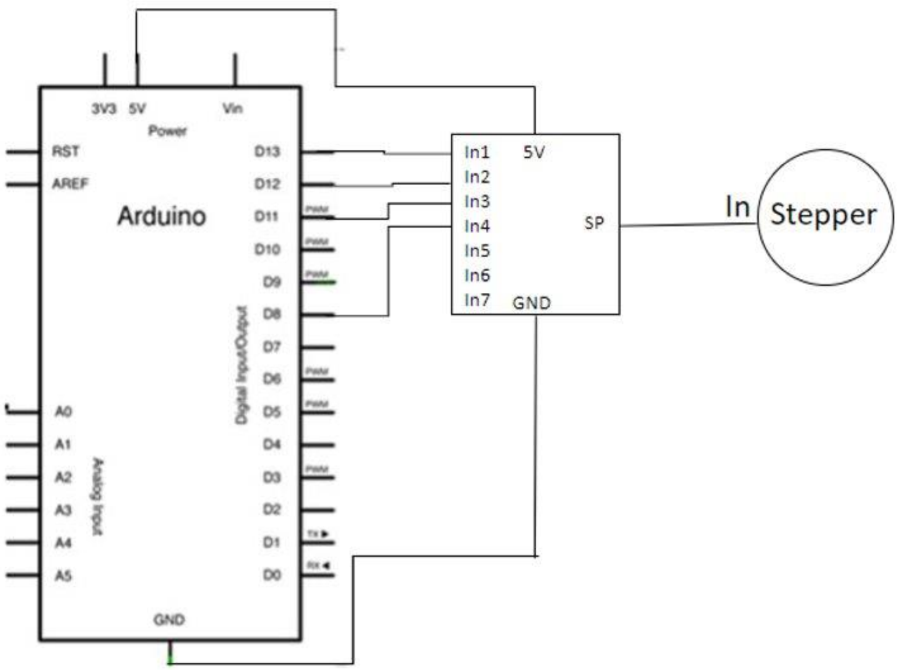
Overall Schematic



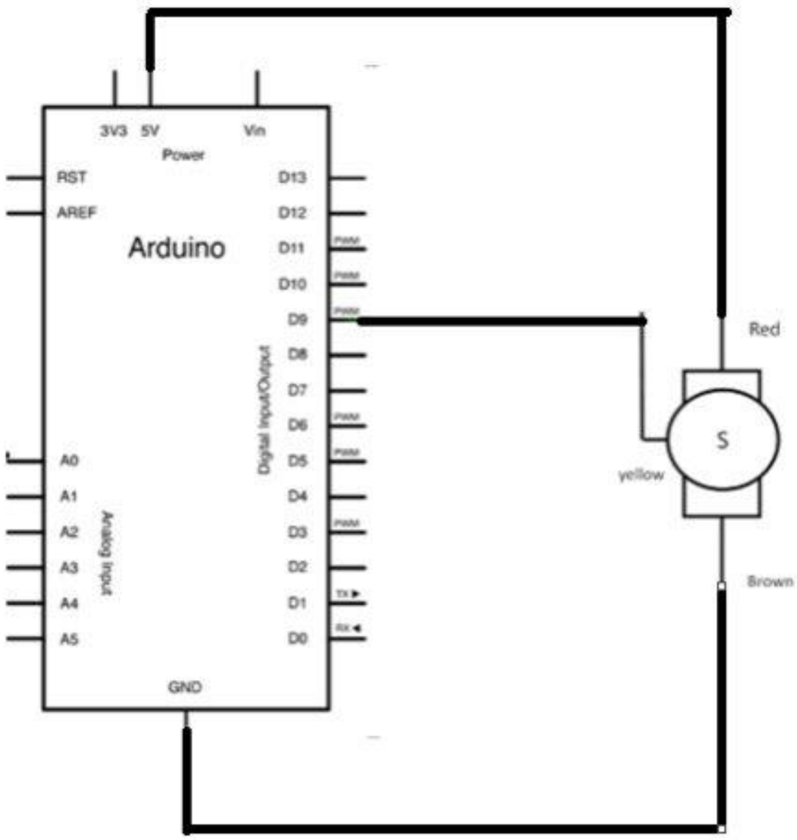
Sampler Stand Schematic



Rotational Base Schematic



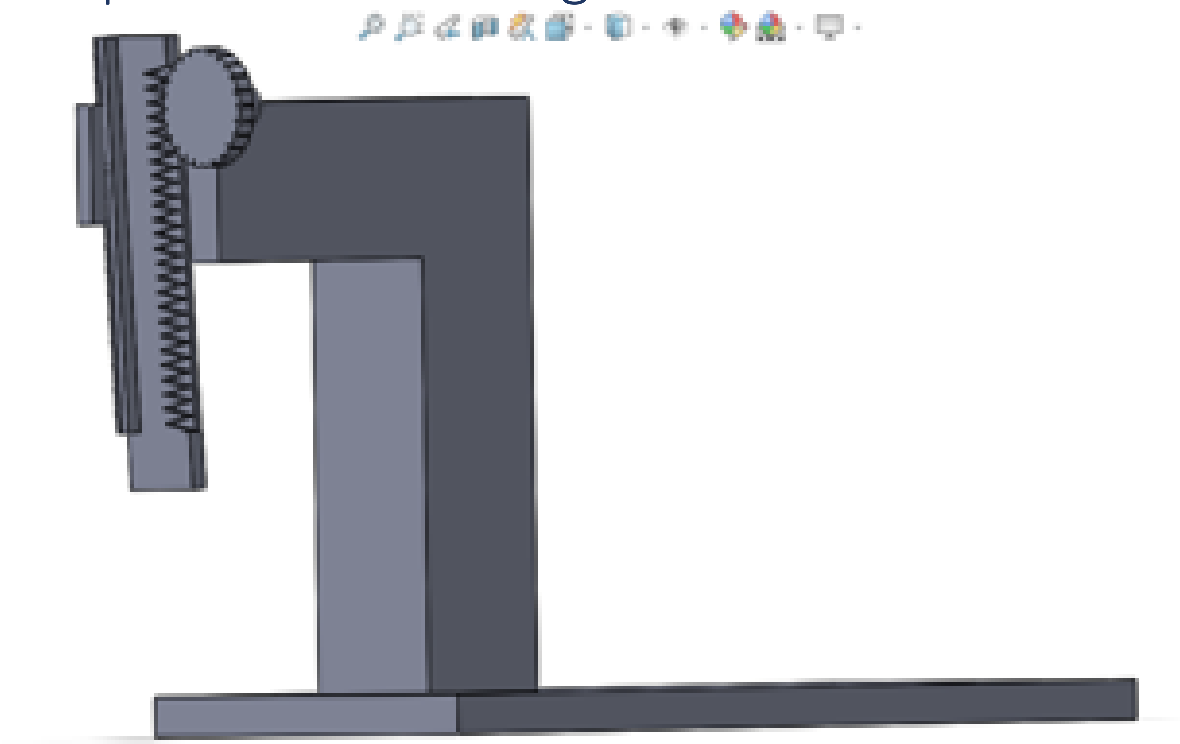
Pipette Squeezer Schematic



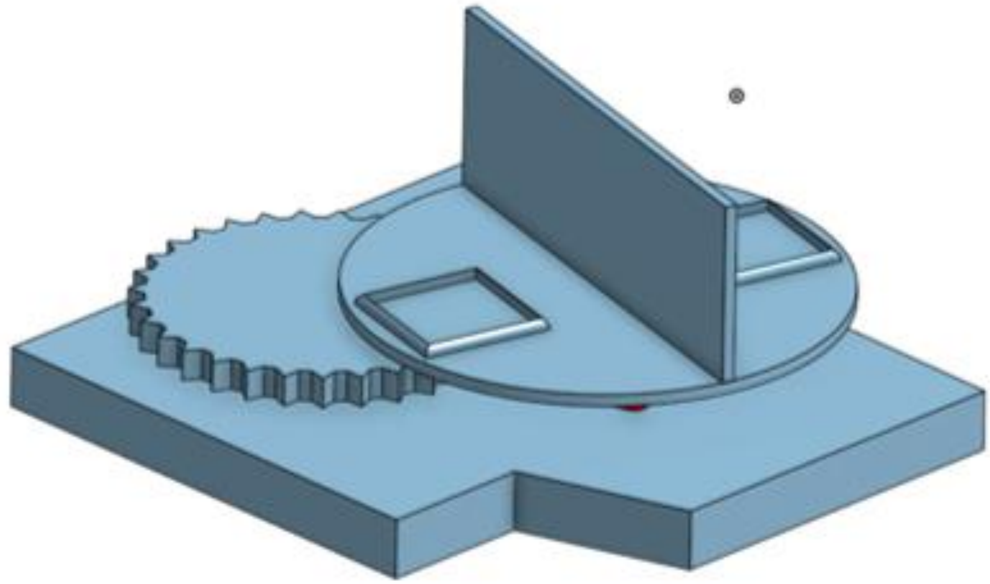
Drawings

Assembled Water Sampler Drawing

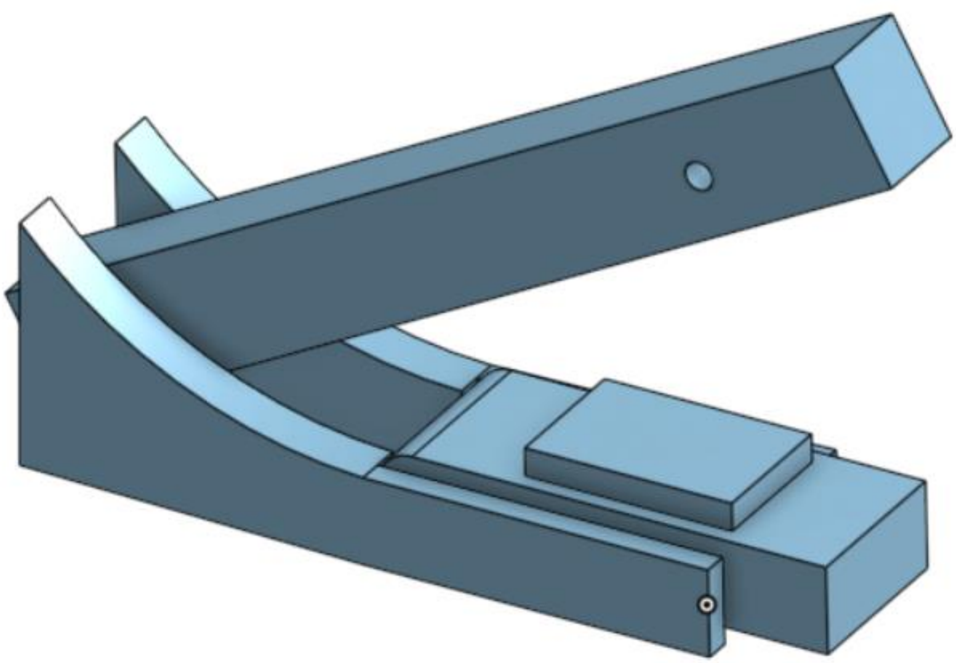
Sampler Stand Drawing



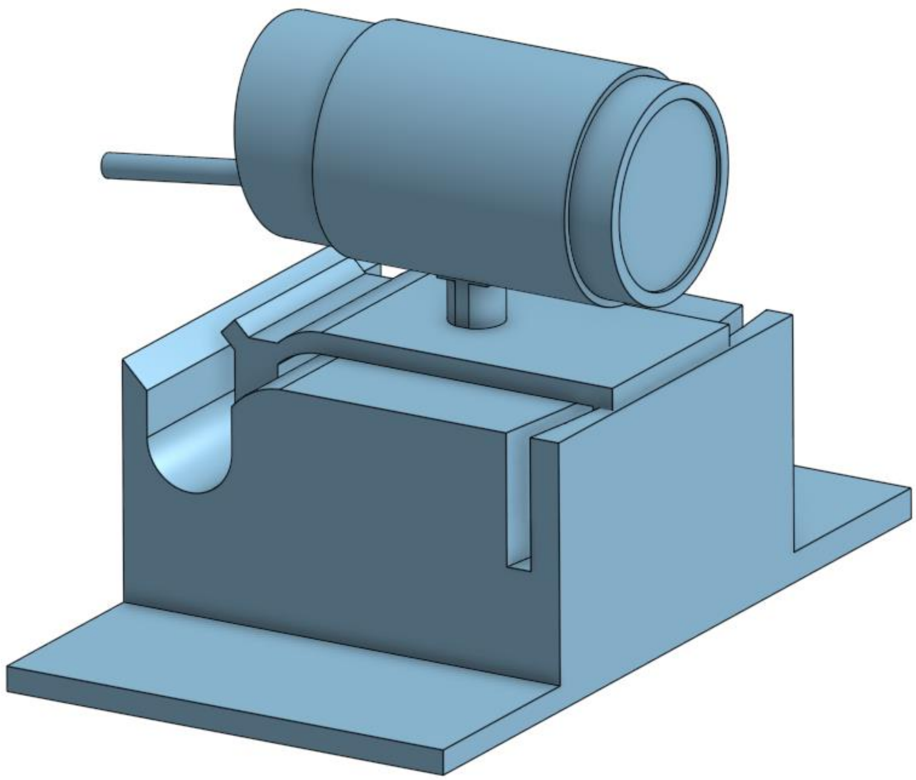
Rotational Base Drawing



Pipette Squeezer Drawing

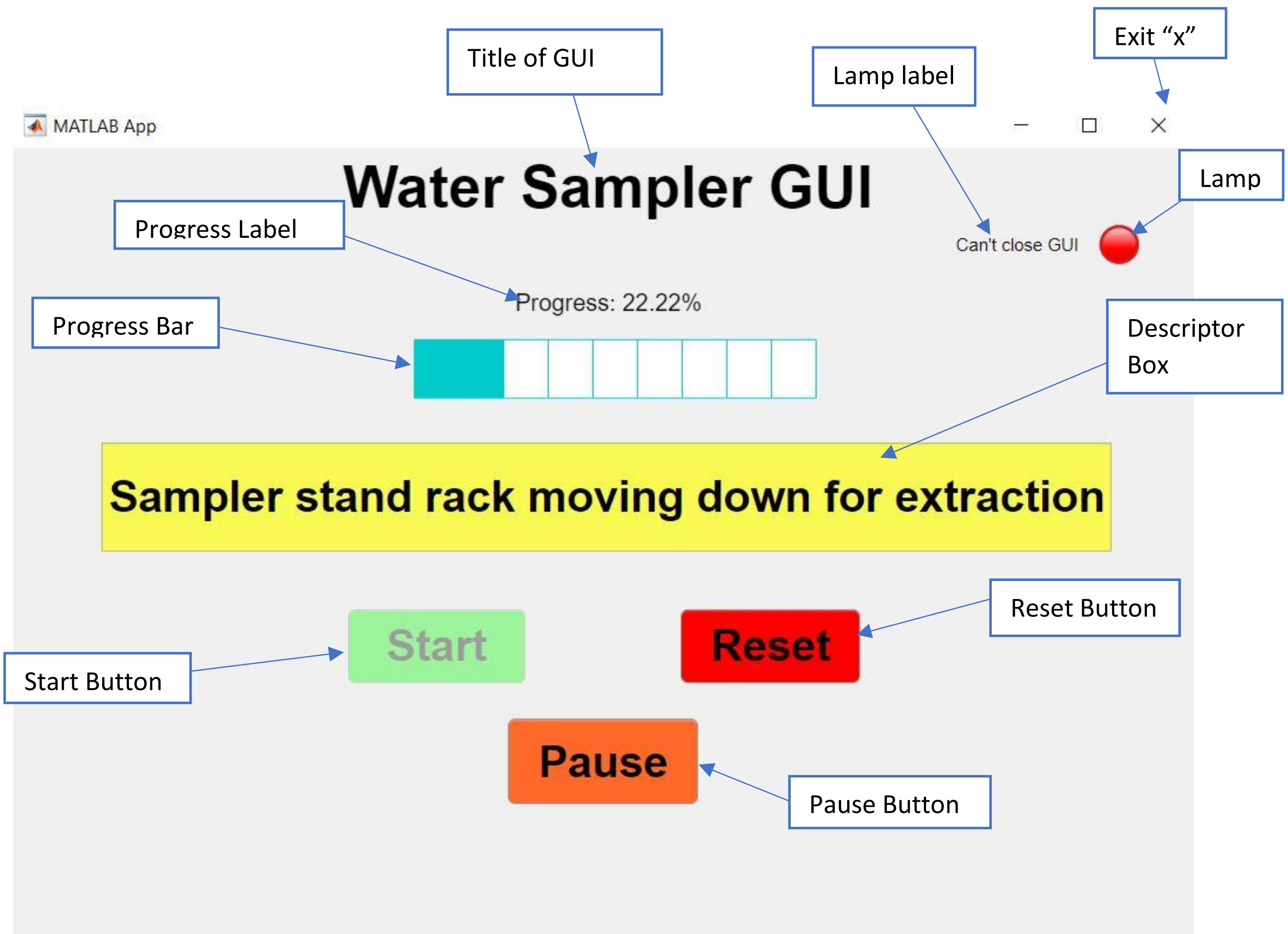


Vision System Stand Drawing



Other contributors: Garrett Matheny, Dorian Lewis, Christian Gratz, and Joshua Hagan

Water Sampler GUI Design



A Note About Functionality

This GUI is meant to solely control the Water Sampler Device and its sampling cycle, and will not run properly if the user attempts to use this application for purposes that it was not designed for. The user needs to place their samples in their respective locations as the labels on the Water Sampler apparatus indicate. The user also is required to download Matlab software, preferably Matlab R2021a, to ensure the program does not run differently than intended. Save the Matlab control code program called WaterSamplerControl in the Matlab directory.

GUI Components

This manual discusses each of the GUI components in two sections. The first section discusses the Interactive features, or the features that user can directly interact with, and features that are non-interactive

. More information about each component is discussed from here onward.

Interactive (User-Controlled) Features

Start Button

Description: When the user first opens the program or begins a new sampling cycle, this option will be the only accessible option for the user. To begin sampling, just press this button. Once the button is pressed, it executes the embedded control code that operates the Water Sampler. This functionality will be disabled until after the Water Sampler resets (the button will appear greyed-out).

Reset Button

Description: When the user first opens the program, this button is disabled (this button will appear greyed-out). Once the user presses the start button, this button becomes enabled. The main purpose of this feature is to allow the user to stop the sampling cycle from fully executing and to reset the Water Sampler. It is important to note that sampling will not stop instantly due to limitations with the control code structure, as well as concerns about potential machine disruption if the user were to stop anywhere they desire. Once the user presses this button, the Descriptor Box will display a "Please Wait" message to let user know their button press was registered. Once the machine resets, this button is then disabled again until user hits the Start Button.

Pause Button

Description: Like the Reset Button, the Pause Button will also be disabled (or greyed-out) until the user presses the Start Button. Once the sampling cycle commences, the user can access this button at any time. The main purpose of the design button is to instantly halt the execution of the control code. This functionality should be used if the Water Sampler machine malfunctions during the sampling cycle. Once this button is pressed, the Descriptor Box displays "Program paused: press again to continue". As the message tells the user, you can resume the program by pressing the button again. While the user is utilizing the pause button, they have the option to exit the GUI. If the Water Sampler Machine resets, the pause button will become disabled again.

Exit “x”

Description: This option is located in the top-right corner of the GUI and allows the user to close the GUI. The user needs to pay attention to the Lamp and Lamp Label (see more on page 38). If the user attempts to close the program when the Lamp and Lamp Label indicate that the user can’t close the GUI, the water sampling cycle will continue uninterrupted. The GUI was designed to ensure that no unexpected errors were to occur if the user closed the GUI prematurely. The user can exit the program only when one of the following occurs:

1. When the Descriptor Box displays “Ready for Sampling”
2. If the Pause Button is pressed (see page 34).

Non-interactive Features

Descriptor Box

Description: This yellow box will be one of the features that becomes visible when the user first opens the GUI. The main purpose of the Descriptor Box is to let the user know what is going on during the water sampling cycle, as well as to let the user know that specific commands were registered. When the program is ready for sampling, it will display the message “Ready for sampling”. This is when the user can place their sample container and water absorbing polymer in their designated locations. Once the user starts the program, the Descriptor Box will update with what the control code is doing throughout the duration of the sampling cycle. The Descriptor Box corresponding updates when the user presses either the Reset Button or Pause Button (see pages 33 and 34). Once the water sampler resets, the Descriptor Box will display “Ready for Sampling” to let the user know they can continue sampling at that point. The user will not be able to manipulate the text in any manner.

Progress Bar and Label

Description: When the user first opens the program, these options will be entirely invisible. Once the user presses the Start Button, both of these features become visible. The main purpose of the Progress Bar and Label is to give the user a relative idea about how far along the water sampler program has gone. After a key phase of the sampling cycle has completed, the progress bar will gradually fill up with a turquoise-color. The Progress Label updates along with the Progress Bar showing the percentage of the sampling cycle that is completed. The Progress Bar and Label will be reset and become invisible either when the user presses the Reset Button or when the sampling cycle runs to completion.

Lamp and Lamp Label

Description: When the user first opens the GUI. Both the Lamp and Lamp Label will be visible. These two features let the user know when it is okay to press the “x” box on the top-right corner of the GUI. There are two scenarios in which the user will be able to close the program:

1. When the Descriptor Box displays “Ready for Sampling”
2. When the Pause Button is used.

In these two cases, the user will know that they can exit or close the GUI once the lamp turns green and the lamp label displays “Can close GUI”. Otherwise, the lamp turns red and the lamp label will display “Can’t close GUI”.

Performance Analysis

Overall, our final design could not execute every function that it was designed for. The sampler stand was able to move the pipette squeezer up and down, but the motion was limited due to slight misalignment between the sampler stand driving gear and gear rack. The rotational base was able to turn, but was largely unbalanced when the two containers were placed on it, so a level had to be used. The pipette squeezer was largely able to squeeze the pipette bulb, but when squeezed the pipette itself did not remain completely aligned with the containers beneath it. The vision system was able to find the sediment layer, but the design of the vision system stand allowed for it to be rotated or angled on accident by the user. The control code was able to coordinate the sampler stand and rotational base to some extent, but was unable to completely reset the subsystem once the sampling cycle had completed. The control code also had trouble when operating the pipette squeezer, as communication between the Arduino serial port and servo motor was somehow interrupted after the other stepper motors moved.

There were several aspects that could improve the overall functionality of this device. For example, one means of improving our device would be to add 4 ball bearings that are underneath the rotational base. Such an addition would ensure that the rotational base would be relatively balanced, regardless of the weight of the sediment layer sample placed on it. This would also help with the vision systems' overall performance, as the cups would be located in a similar position every time the webcam takes a snapshot. The sampler stand driving gear and its gear rack system could be improved upon by increasing the teeth thickness size to ensure that they properly mesh. The pipette squeezer and the issue with straightening the pipette could be resolved by placing additional rings to more fully secure the pipette and restrict its movement. The vision system stand could be resolved with a rigid, clamping mechanism that would ensure that the user could not rotate or angle the camera once it is under the sampler stand. As the control code is dependent on all the other subcomponents properly functioning, experimentation is needed to ensure that it properly integrates all the other subsystems. Future development of the control code will need to make sure to figure out how to properly connect the servo motor and stepper motors connections to the Arduino serial port.

While our device may not have been entirely successful, it is important to note that each of the ideas for the subsystems were relatively creative and plausible to implement. The biggest obstacle for such implantation, though, was that there were various schedule conflicts with the majority of the group members. This meant that there was not as much time to develop and modify each of the subsystems as we would have liked. The development of the water sampler has taught all of us several important aspects if teamwork, such as how to communicate ideas with one another, how to recognize and work around each others' differences in approaching problems, and the importance of time management. Each of the team members of this group will carry these lessons we have learned from this experience as we keep moving forward into the future.