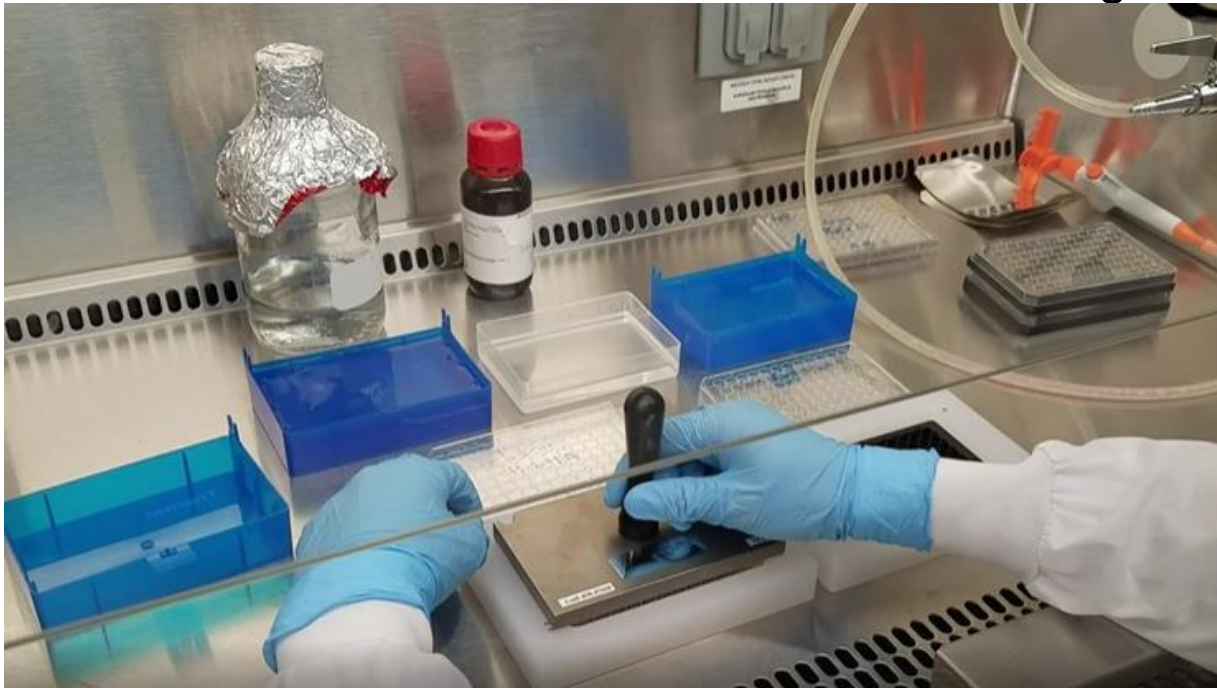


# Pin Transfer Robot for Chemical Screening



*Figure 1 - Manual chemical transfer*

Department of Electrical Engineering and Computer Science  
University of Central  
Dr. Samuel Richie

Florida

Group H

Christopher Clifford  
Brenden Morton  
Dominic Simon  
Yousef Abdelsalam

Electrical Engineering  
Computer Engineering  
Computer Engineering  
Computer Engineering

# Table of Contents

<b>Pin Transfer Robot for Chemical Screening</b>	1
<b>Table of Contents</b>	2
<b>Executive Summary</b>	1
<b>Project Narrative</b>	4
<b>Design Considerations</b>	6
<b>Objectives</b>	8
Motivation	8
Personal Bibliography	8
Related Projects	10
Introduction	10
Building the Library	10
IHC and Imaging	12
Activin A Screen	12
Results	14
Summer 2019	14
Results and Future Work	15
<b>Design Goals</b>	17
<b>Technical Requirements</b>	17
<b>Build Plan</b>	18
<b>Prototype Plan</b>	21
<b>Block Diagrams</b>	23
<b>Estimated Budgeting and Finance</b>	24
<b>Hardware Design</b>	26
CAD Software	26
Eagle	26
Schematic	26
ATMEGA 2560	28
Power	29
Bill of Materials	30

BOM	30
PCB Fabrication	30
Software Design	31
User Interface Algorithm	31
Avoiding OLED Burnout	32
Pin Transfer Step Algorithm	32
Stacking Steps	34
Washing Step Algorithm	35
Microcontroller Connection Functions	37
User Interface Functions	38
Motor Functions	39
Class Diagrams and Data Structures	40
Standards	40
Code of Federal Regulations, Title 40 Part 262	40
PEP 8 and C++ Core Guidelines	42
Robot Operations	42
Input Microplate Stack	42
Workspace Microplate Rail	44
Pin Transfer Tool	45
Washing and Drying the Pin Tool	46
Output Microplate Stack	47
LCD and Keypad	49
Safety Features	50
Cleaning and Maintenance	51
Overview of Design and Technical Architecture	53
Design Constraints	54
Economic and Time Constraints	54
Manufacturing and Sustainability Constraints	55
Societal and Political Constraints	55
Health and Safety Constraints	56
Ethical and Environmental Constraints	56
Facilities and Equipment	57
Personnel	58
Research and Investigations	60
Existing Similar Projects and Products	60
High-throughput Chemical Screening Robot	60
Epson Compound Transfer Robot	60
Seiko Compound Transfer Robot	61

<b>Project Part Selection</b>	62
Pin Transfer Tool	62
Manual Transfer Tool	64
Robotic Transfer Tool	65
Chemical Library	66
Stacking Concepts	67
Microcontroller	70
ATmega328P (Arduino UNO)	72
ATmega2560 (Arduino MEGA 2560)	72
MSP430G2452	72
MSP430FR6989	73
Summary of Boards	73
Design Choice	74
ATMEGA16U2-MU	75
Wireless Connectivity	76
Motors	77
Motor Types	78
Stepper Motors	78
NEMA 17 Stepper Motor	78
NEMA 23 Stepper Motor	79
NEMA 23 High-Torque Stepper Motor	79
NEMA Motor Comparison Table	79
Servo Motors	80
Design Choice	81
Motor Driver	83
TB6600	83
TB6560	84
A4988	84
L298N	84
DRV8825	85
DM542T	85
Code Libraries	86
C++	86
Stepper.h	86
Design Choice	87
Linear Rails	87
Belt-Driven	88
Lead Screw-Driven	89

Part Selection	90
Linear Actuator	90
Fans	92
Solenoid Valve	93
Power Supply	94
Code Base	96
Arduino Wrapper of C/C++	96
Python	97
Design Choice	99
User Interface	99
Keypad	99
Screen	103
LCD	103
OLED	104
Design Choice	105
Part Selection Summary	107
Source Control	108
Github	109
Workspace Base	110
3D Printing	110
Sheet Metal	110
Plexiglass (Acrylic)	111
Prototype Build Procedure	111
Testing Procedures	112
Photometric Measurement	113
Photometric Measurement Procedure	113
Fluorometric Measurement	115
Gravimetric Measurement	115
Consultants, subcontractors, and suppliers	115
<b>Hardware Testing</b>	117
OLED Testing	117
Keypad Testing	117
<b>Administrative Content</b>	118
Milestones	118
Budget Analysis	119
Summary and conclusions	121
Appendix Updated	122

Permissions	122
Adafruit	122
ICCB-Longwood Screening Facility at Harvard Medical School, Boston MA	122
OpenBuilds.com	123
V&P Scientific	123
Pseudo Code for Scanning Procedure	124
Citations	124

# Executive Summary

This document explains many of the design and technical considerations made when building the pin transfer robot. In this executive summary we will summarize what is discussed in this document and the overarching goal of this project. The pin transfer robot is a tool designed to fill a gap in the current product line being offered in small laboratory experimental tools and equipment. The purpose of this robot is to help laboratory technicians to perform a liquid pin transfer process without the inaccuracies or difficulties that come with the manual pin transfer process. The inspiration for this project originally came from a summer internship at the Mayo Clinic where the pin transfer process was used many times in an experiment and the usefulness of this robot became apparent. The motivation for the project and its utility is further discussed in the *Project Narrative* section of the document.

Many different design considerations were made before selecting our current two dimensional gantry design with a workspace rail to mobilize microplates. This is discussed in full detail in the *Design Considerations* section of the document. This project was designed with a specific set of objectives and goals in mind. These are used to validate the project's usefulness for the scientist that will eventually operate as well as to gauge how well we built this robot to fit the niche we originally set for it. The *objectives* section defines first:

1. The motivation for the project and the designers of the project including personal bibliographies and personal statements of design purpose.
2. A chronological narrative about previous research experiences leading up to the project that inspired the construction of this robot. This includes many small molecule and chemical screens and cell culture experiments as well as building a chemical library providing a natural progression into high throughput screening assays.
3. We end the objectives section with an overview on our design goals discussing what the pin transfer robot should be able to accomplish at the end of the project.

The most important part of the *Design Goals* section of the document is the *Technical Requirements*. This part of the document contains a comprehensive list of each part of the project that is necessary to demonstrate and perform in order to be considered a success. This includes technical and design limitations to the robot's dimensions, the number of microplates that can be processed at one time, and what features the Graphical User Interface will include in its operation and functionality. This section defines the overall requirements that guide the entire project and will explain why each design choice was made.

The *Build Plan* takes all of the *Technical Requirements* into design consideration. Here, in this section we begin to discuss how the pin transfer process works manually, and then we begin to start to replace this process step by step with our robot. We discuss how each subsystem will replace its manual equivalent, how this will change the accuracy or repeatability of each step, and how we plan on implementing each subsystem. The actual

implementation of each of these subsystems is elaborated on in greater detail in the *prototype plan* section. Here each functional section of the robot is separated and defined. These subcategories of the robots are divided by their separate design considerations and functionality. These parts of the robot were handled by different people and have very clear functions apart from each other such as input/output microplate stacking and unstacking, moving the plates from input/output stacks in the workspace area, operating and cleaning the pin transfer tool and then drying of the pin transfer tool, and finally the graphical user interface as well as any buttons and switches.

The *Prototype Plan* continues by containing a detailed parts selection table where we define the cost and choice of parts used for each subsystem. We elaborate on why the choices were made and include explanations, CAD drawings, and source material where we deem necessary. A house of qualities and a decision matrix are included to support our decisions and why we are putting our resources into the development of certain aspects of the project we deem the most important. We also discuss project funding here and how the project funding will be split amongst group members as of now our project has no external funding. A large majority of the funding will be provided by Christopher as he has a practical use for the robot and plans on using the robot in the future to conduct scientific experiments. The *prototype plan* section also importantly includes flow charts both for hardware as well as software operations of the project. These flowcharts are important both for the designer and user to better understand the workflow of the robot and how each of the subsystems interact with another.

In *Hardware Design* the project is described in great technical detail. This includes schematics and computer aided design (CAD) drawings using programs such as solidworks for mechanical parts and Eagle for integrated circuit board design. This section describes each part that will be built, purchased or designed. We discuss how to properly use these parts to prevent LED burnout and other such failures.

The *algorithms* are discussed later. These algorithms are both for hardware functions as well as software. The algorithms include the pin transfer process and the stacking/unstacking microplates process. The algorithms are explained by very detailed flowcharts that help elaborate on proper usage of the components as well as give a general understanding as to how the subsystem functions and interacts. These sections are particularly helpful if you are trying to learn how to properly interface using the LED/Keypad and the graphical user interface (GUI).

*Robot Operations* serves as a user operating manual for the robot. It contains the following sections that give a medium depth overview of each of the subsystems and can direct the user to other parts of the document if more information is required. This section also includes maintenance information and safety information.

*Robot Operations* subsections:

1. Input Plate Stack
2. Workspace Microplate Rail



3. Pin Transfer Tool
4. Washing and Drying the Pin Transfer Tool
5. Output Microplate Stack
6. LCD and Keypad
7. Safety Features
8. Cleaning and Maintenance

*Constraints* includes multiple sections that talk about project design constraints that limit the designers ability to include certain or additional features to the project. The design constraints listed in the document are:

1. Economic and Time
2. Manufacturing and Sustainability
3. Societal and Political
4. Health and Safety
5. Ethical and Environmental
6. Facilities and Equipment

*Personnel* portion of the document gives an introduction to the group members that helped design and build this robot. The project team consists of three computer engineers and one electrical engineer. The project is headed by Christopher Clifford and built with the intent to be operated in the Islet Engineering laboratory at the Mayo Clinic Rochester. The *Personnel* section outlines the responsibilities and contributions of each of the authors of this document, what these individuals offer to the team with their expertise and experiences.

*Existing Similar Projects and Products* contains an in-depth exploratory dive into the current commercial industry as well as research equipment with similar purpose or functionality to our pin transfer robot we propose. This includes a procured list of robots ranging from high content screening facility robots used in large pharmaceutical or drug discovery labs to some smaller industrial robots and liquid handling robots. There are no direct comparisons made because our robot will be filling a niche that is currently empty. There are currently no small scale robots made with the intent and purpose of pin transfer operations. The closest thing is liquid handling stations that have adapters fit to enable pin transferring operations however as discussed in this section of the document this impairs the robots ability primarily in the capacity of plates it can hold therefore severely limiting the autonomy of these robots, a design constraint our pin transfer robot doesn't have.

*Project Part Selection* is a very detailed section that deliberates why certain components were picked to be used in our project and their advantages are discussed when compared to alternative parts. Each subsection typically discusses each of the potential parts or materials that were considered, and then each of the candidates benefits and disadvantages are divulged. The importance and usage of each of these parts is mentioned as well in order to give context as to why certain design constraints exist for the parts. The parts discussed in order are:

1. Pin Transfer Tool
2. Manual Transfer Tool
3. Robotic Transfer Tool
4. Chemical Library
5. Stacking Concepts
6. Microcontroller
7. Wireless Connectivity
8. Stepper Motors
9. Servo Motors
10. Motor Driver
11. Code Libraries
12. Linear Rails
13. Fans
14. Solenoid Valve
15. Power Supply
16. Code Base
17. User Interface
18. Keypad
19. Screen
20. Source Control
21. Workspace Base

In the *Testing Procedures* sections the methods used to evaluate the proficiency of the mechanical or software subsystems is expanded on. These sections list protocols, values, and tests used to determine whether a subsystem is working to the design specification standards outlined in the *Technical Requirements* sections. There are testing procedures for the following topics:

1. Photometric Measurement
2. Fluorometric Measurement
3. Gravimetric Measurement
4. LCD testing
5. Keypad testing

Lastly, *Administrative Content* contains many supporting documentation including milestones, budget analysis, project summaries and conclusions. Permissions for images and content provided by external entities is given in this section with all attached communications with the owners of the images, files, or other owned content with the owner's permission to include the content in this document. *Citations* are included at the end of the document to credit the authors of sources used in the research and development of this project.

## Project Narrative

As an engineering student working in a regenerative medicine laboratory for the past two summers, I'd thought of many ways to optimize experiments by inventing tools that could automate or assist with parts of the experimental process. Much of my research involved testing an assortment of small molecules and growth factors on differentiating stem cells to determine their influence on the cell's protein expression. This process was normally done by me manually and it is very tedious and any small error or inconsistency can have a massive influence on the outcome and the repeatability of my experiment. For this reason, I became interested in resting a robot that could carry out the chemical screening process for me so that there would be drastically less inconsistencies and time in my experiments. There are currently robots that do what I am describing but as you will see, they will cost anywhere from tens of thousands to millions of dollars. Some labs are completely dedicated to screening chemicals for toxicity and safety, or to find potential anti-cancer drugs. My goal with this project is to create a small robot that could be used by biology labs whose primary focus is not chemical screening and comprises a small part of what the lab does. The benefit of this is it would increase possibilities for experiments in these labs while not having a monetary barrier to entry. I myself would use a robot like this to conduct my experiments in the future and I can personally say that it would greatly increase my productivity. With this robot I could expect orders of magnitude more discoveries based on the quantity of experiments I could conduct.

Some alternatives to our project in the marketplace currently include full scale lab implementation, liquid handling robot adapted to handle both automatic and manual pin transfer. Full scale lab implementation takes up an entire building with incubation chambers, imagers and robotics. The entire chemical treatment, cell culture, and imaging process is automated. For reference, I have included two videos of a full scale lab implementation, one from the [Environmental Protection Agency](#) and another from the [Broad Institute](#). The first real possibility for a smaller lab that wants to get started in chemical screening would be purchasing an adapter for a liquid handling robot. Liquid handling robots are used to dispense and sample liquids from wells or microplates. Some companies such as V&P Scientific sell adapters that can be mounted to the head of liquid handling robots so that a pin transfer tool can be fitted to the robot. This effectively creates a pin transfer robot with some major drawbacks. The biggest problem is that liquid handling robots are designed to only handle one plate a time, which means that if you would like to treat duplicate plates or many different cell plates, then you would need to manually move the plates in and out of the workspace after each program execution. Ideally, our project will handle plate management by placing the plate in and out of the work space. This would strongly differentiate our project from available options on the market today.

The purpose of this document is to act as a report on any and all plans and research regarding the project. The Design Considerations section of this report displays some initial ideas and sketches for the design of the gantry robot. We've gone through several iterations in order to ensure that our design and implementation was as budget friendly as possible while still maintaining high quality pin transfer operations. The Objectives section of this report outlines the motivation, Design goals, and Technical requirements

in quantitative detail. With proper design goals and technical requirements, the expectations of the project become clear. From there, the discussion can move forward. The Build Plan section simply discusses the approach for building the final iteration of the Liquid Handling Robot. The Prototype plan is an independent build plan whose purpose is to provide a build plan that prioritizes certain build steps first to make sure that basic functionality can be fulfilled before building on it. The Personal Bibliography provides some history behind the subject matter expert in our group, Christopher Clifford. There is also a series of related projects from which inspiration was drawn. The hardware and software block diagrams were made to show a top-down overview of the hardware and software setup that we will be using to build the Liquid Handling Robot. The Estimated Budgeting and Finance section of this document outlines a table of all of the parts that we will be using, the quantities in which we will be using them, the total prices of each of the groups of parts, and the total estimated amount that will need to be paid towards the project. The rest of the sections of the project proceed to go into more and more detail about the parts, the testing process, the software design and development specifications and standards, the robot operations, and the design constraints of the project.

## Design Considerations

In the field, smaller research labs tend to have to use some form of pipetting/pin-transferring tool that needs to be properly handled and operated. As stated before, not only can this process be tedious, but it also entails room for human error and cross-contamination. In order to alleviate this, we will attach a pin transfer tool to a robot in such a way so as to automate the pin transfer operation. There would also be a drying fan that would be activated after the pin transfer operation completes. There must also be a mechanism for aligning the pin transfer head with the 96/384 well plate, which must involve the head moving up and down for the pin transfer motion in the least. Ideally, there would also be a mechanism for moving the plates into position from an existing stack or repository of plates.

There are many such possible implementations that do just that. One of the more reasonable and practical implementations is represented in the sketches shown below in Figure 2 and Figure 3:

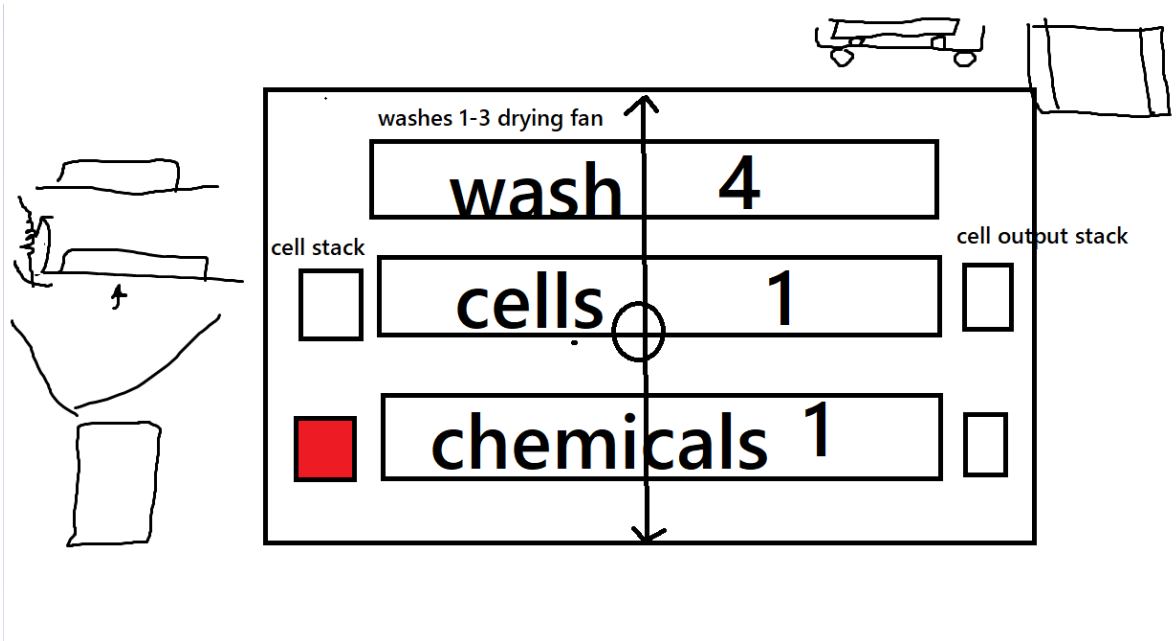


Figure 2: Rough Illustration of Design Idea 1

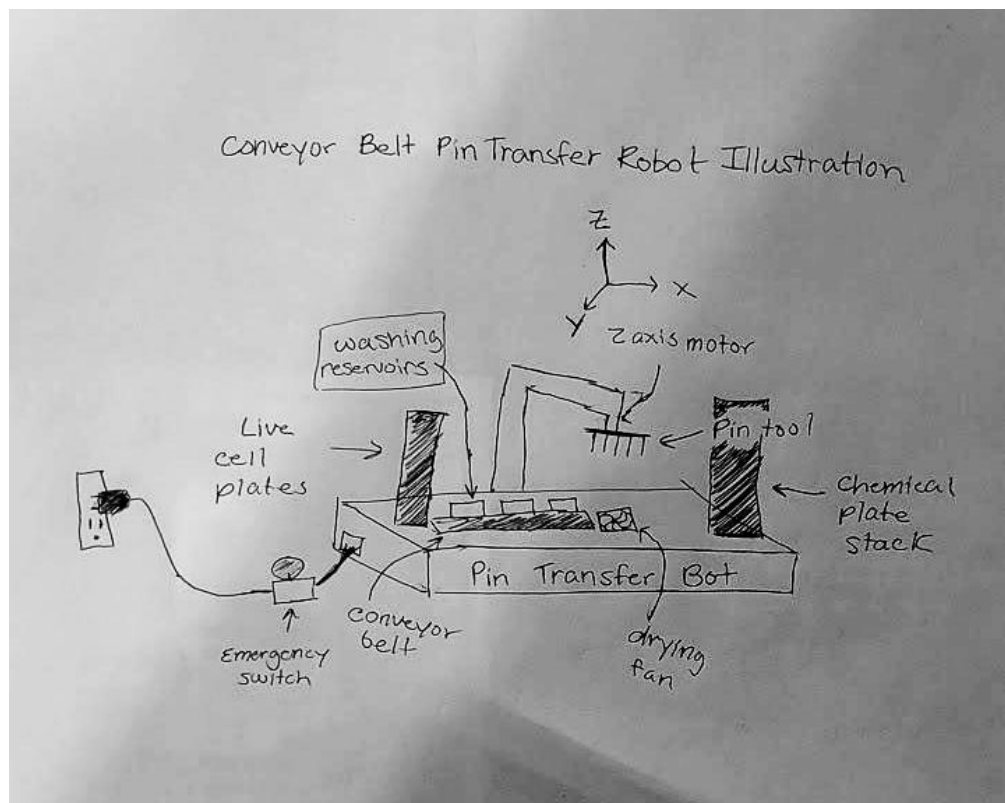


Figure 3: Rough Illustration of Design Idea 2

As you can see, there should be two FIFO structures on the left as per Figure 1, labelled as the white and red squares. There should be a gantry head that has two degrees of

freedom, namely the y and z axes. An example of the 2-axis gantry can be seen in Figure 2. There is no need for a third degree of freedom here since the wash steps, cell plates, and the chemical plates are all moved through their own separate workspace rails into the appropriate position. When the cell and chemical plates are in the appropriate position, the head must drop onto the chemicals and use the pin tool to transfer the chemicals to the cells. Once the pin transfer has succeeded, the pin tool must move to the wash belt in order to be washed in a step by step process. Possible wash steps include combining a cleaning chemical agent with a drying fan or suction. From there, the cell and chemical plates will move to an output stack.

Lastly, there is the possibility of using a gantry robot that can move anywhere within the XYZ coordinate plane provided for it in order to perform the pin transfer operation on well plates that are provided within a grid-like area.

Some nice-to-have ideas that could be implemented would be some kind of barcode scanner that can read information about the time in which the pin transfer operation was implemented along with the cells and chemicals that were used to be read into a database. This is mostly because the FIFO structures that we plan on using are not going to be sorting the plates in any way, and so implementing barcodes would allow researchers to identify the plates and know what reactions took place so that they may document the results of the reaction as needed. Reference the block diagram (Figure 3 and Figure 4 for more details).

## Objectives

## Motivation

The motivation for this project is multipurpose. First and foremost, the motivations of this project are to create a robot that can assist with the chemical screening process used in regenerative medicine research using stem cells. The need for this solution came about from an internship experience of a group member and we felt that we had the capacity to develop a solution that could be beneficial to the scientific community. A secondary motivation for the project is that we as a group of graduating seniors at the University of Central Florida need to demonstrate the knowledge we have gained over the course of our engineering bachelor's degrees. To do this our classes culminate in this senior design course where we collaborate on a capstone project that displays our potential.

## Personal Bibliography

I aspire to obtain a Ph.D. in Biomedical Engineering and to cure Type 1 Diabetes through innovative blends of human physiology and machines. My strong background in both applied mathematics and biology will equip me to achieve solutions based on novel technical insights that bridge these fields. Currently, I am working towards an Electrical

Engineering major and a Bio-Engineering minor at the University of Central Florida. I see these concurrent degree pathways as key steps towards my future goals in diabetes research, because together they are preparing me for research opportunities in regenerative medicine where I can investigate beta cell differentiation and encapsulation. By building a solid foundation in both engineering and biology, I will gain unique perspectives on problems I face in my research career—a unique confluence of disciplines which I will embody as a scientist.

My education is enhanced through undergraduate research. As a first semester freshman, I interviewed for openings in several Bio-Engineering labs and joined the Interventional Robotics Lab. This experience solidified my early interests in research. My excitement for diabetes research inspired me to partake in two summer positions at the Mayo Clinic Rochester working in the islet engineering lab under Dr. Quinn Peterson where I acquired skills in regenerative medicine and tissue engineering. One of my contributions was an in-house chemical library of small molecules and growth factors which I used to screen developing endocrine cells (Figure 5). My ambition is that the research I conduct will benefit diabetics. To study how research translates from bench to bedside, I began volunteering as a researcher at the AdventHealth Translational Research Institute where I assist in designing multiple ongoing clinical trials related to diabetes, metabolism, and early diagnosis. These experiences have been profoundly formative for me as a researcher. I believe my early experiences in research will strengthen my graduate school applications and provide a valuable arena for going beyond the minimal requirements of an undergraduate STEM education.

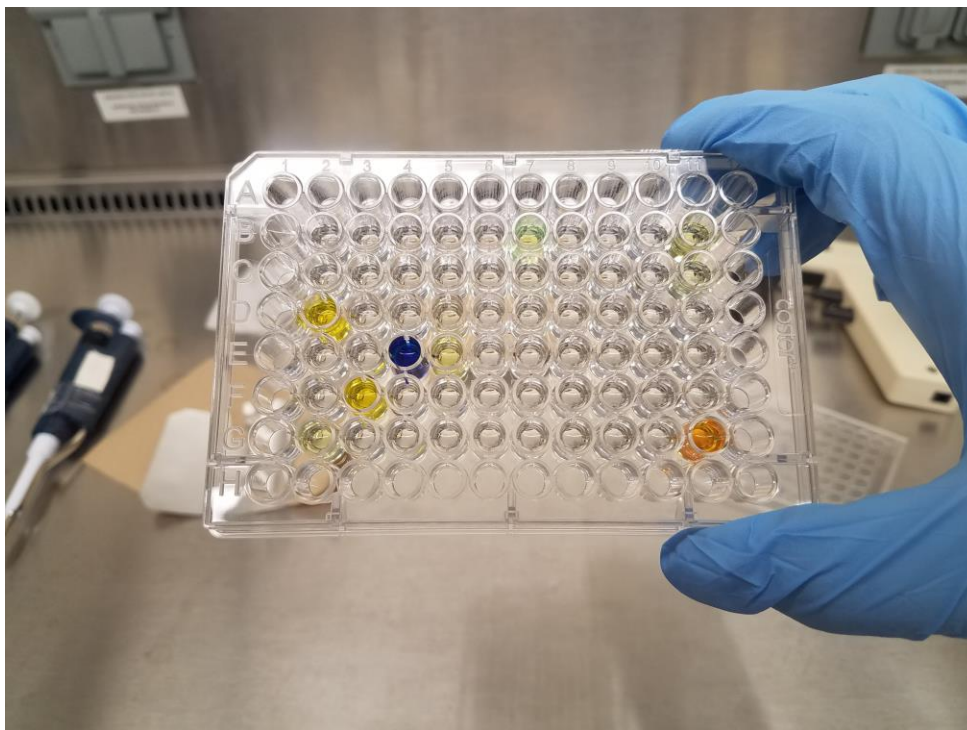


Figure 5 - Example of a chemical library plate used to screen live cells.

While undergraduate studies and research are necessary to achieve my goals, I am also intricately involved with the diabetes community in central Florida. I lead the Type 1 @ UCF student organization where we serve a community of students and professionals with diabetes. The Type 1 @ UCF organization invites speakers from biotech companies, health institutions and the Orlando area to discuss diabetes and how to overcome its associated challenges. We offer free A1C tests, psychological and dietary counseling, as well as on-site medical professionals as resources for our members. The rewarding part of the Type 1 Organization is when a member shares their personal triumphs with managing their diabetes, catalyzed by our club's support. Community involvement through my organization drives my motivation to research diabetes and ultimately find a cure.

Being a 2019 recipient of the prestigious astronaut scholarship, I used this opportunity to support my personal research projects including building an autonomous pin transfer robot which I will use this coming summer in Dr. Peterson's lab to validate exciting discoveries made in my previous internship. I plan to use any additional funding to continue my research on beta cell differentiation from stem cells, and to fund these ongoing projects of mine as they develop and evolve throughout my undergraduate research career.

## Related Projects

### Introduction

In the United States approximately 1.25 million individuals live with Type 1 Diabetes (T1D) [1]. T1D is an autoimmune disease that eliminates the production of insulin in those affected by attacking the insulin-producing beta cells ( $\beta$ -cells) in the pancreas. People with diabetes rely on exogenous insulin to survive.

There are many different methods being studied to replace  $\beta$ -cells destroyed by T1D. One promising area of research is  $\beta$ -cell generation from embryonic stem (ES) cells.  $\beta$ -cells have been successfully generated from ES cells in laboratories in the hope of transplanting the differentiated stem cell-beta cells (SC- $\beta$ ) into T1D patients [2].

Numerous compounds are necessary to differentiate ES cells into SC- $\beta$  cells. These compounds can be prohibitively expensive, and the differentiation process is complicated and not fully elucidated. I spent the past two summers at the Mayo Clinic Islet Regeneration Lab where my project was to create an in-house chemical library, and use it to reduce the cost of "designing" an islet, as well as, more specifically, to optimize the differentiation protocol to make more insulin secreting SC- $\beta$  cells.

### Building the Library



To create our in-house chemical library, I gathered small molecules and growth factors from all stages of our differentiation protocol along with some promising compounds and experimental drugs recommended by my PI or that I had identified in literature.

My first step was to organize a wide assortment of small molecules within a spreadsheet database by their name, respective targets, and effective concentrations. 216 small molecules were selected for screening. 10 mM stock solutions of each chemical were diluted in DMSO (figure 6a) and deposited into four 96-well microplates at three concentrations in triplicate totaling 36 microplates (Figure 6b).

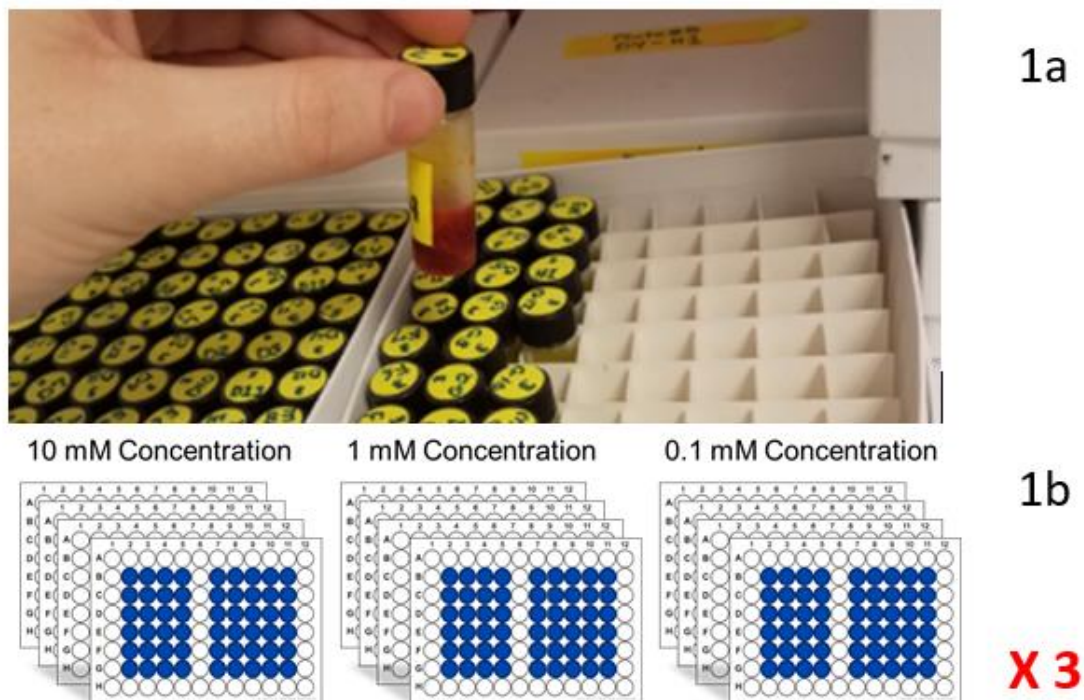
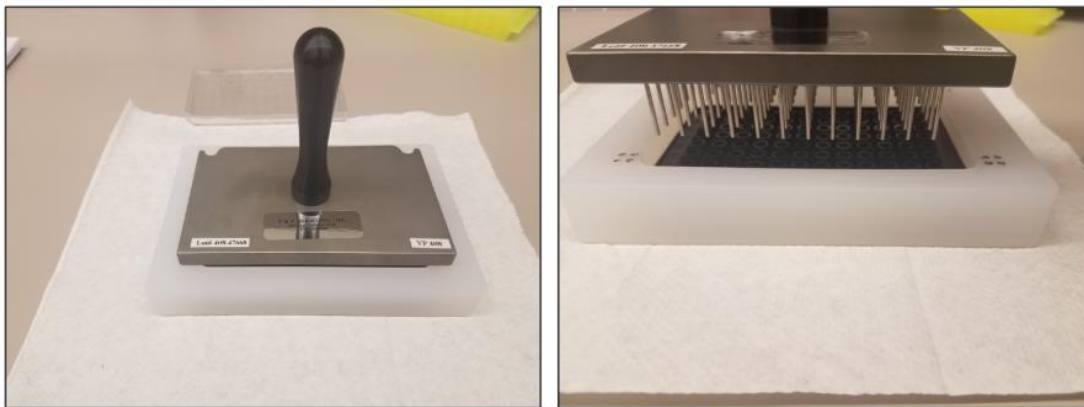


Figure 6: a) 10mM stock solutions b) One of three copies of the compound library

In order to facilitate the transfer of compounds from the library plates to plates with seeded cells, I used a V&P Scientific VP408 Manual Replicator shown in Figure 7 . This tool enabled a higher throughput with increased accuracy.

# Pin Transfer Tool

- Transfers  $\sim 0.2 \mu\text{L}$  per well
- Used for chemical screening



*Figure 7: Pin transfer tool. Transfers  $\sim 0.2 \mu\text{L}$ /well.*

## IHC and Imaging

Treated cells were fixed using 4% PFA and stained with primary and secondary antibodies specific to the cell population screened. Three washes with PBST were performed between each step. After staining, microplates were analyzed using a ThermoFisher Cellinsight CX5 high content imager.

## Activin A Screen

Utilizing the chemical library, I conducted a chemical screen to identify an Activin A (AA) substitute. AA is one of the many factors used in SC- $\beta$  cell differentiation and is especially costly. It coerces pluripotent ES cells into the definitive endoderm stage (figure 8) [2]. My objective was to screen for other small molecule candidates that could replace AA in the SC- $\beta$  differentiation protocol.

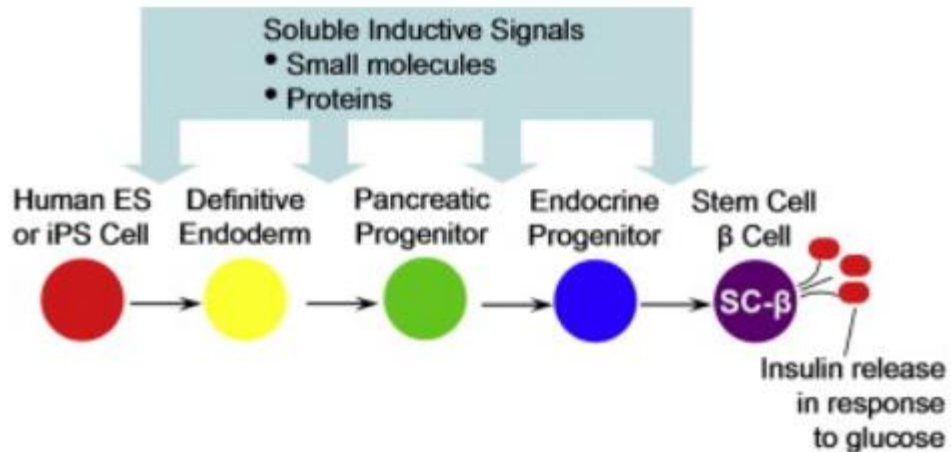


Figure 8: Unique stages in SC- $\beta$  differentiation [2].

Lab protocol routinely combines Activin A and CHIR (a small molecule) to convert ES cells into definitive endoderm cells. By using CHIR alone as a negative control, I could directly compare the effects of AA to other small molecules being screened.

Months were dedicated to optimizing the protocol and antibody conditions to ensure I had a confident assay that could identify influential compounds. During the process of assay development, I learned how to adhere cells to well plates using Matrigel, aspirate cells, wash cells with PBS, 'fix' cells by crosslinking them with paraformaldehyde, and image cells using immunofluorescence staining. OCT4 (an ES cell marker) and SOX17 (a definitive endoderm marker) staining was used to determine the identities of the cell subpopulations. This can be seen in Figure 9 and in Figure 10 below.



Figure 9 - Microscope

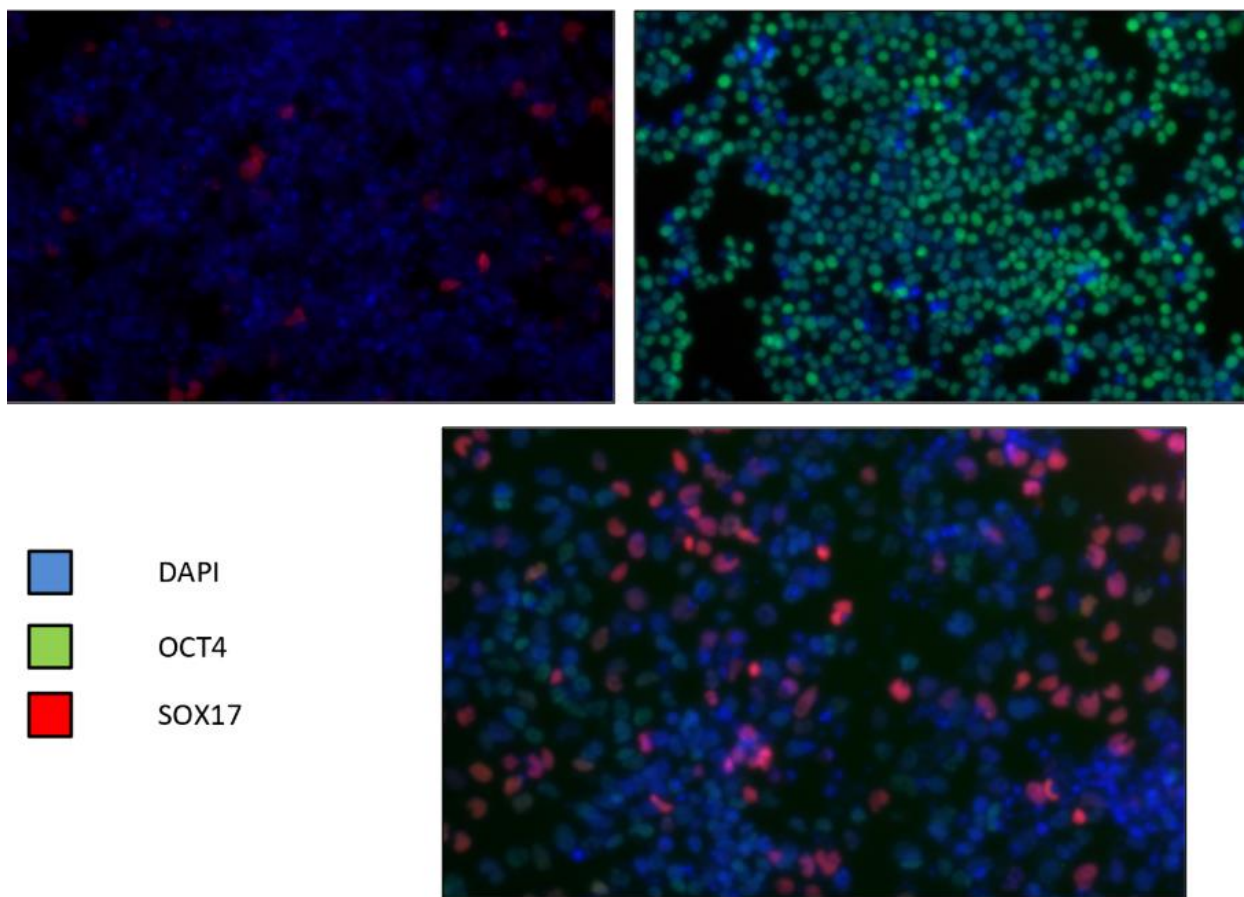


Figure 10: Immunofluorescence staining for SOX17.

## Results

Ten or fewer chemicals showed any SOX17 expression. After manually observing each well under a microscope, it appeared that most were falsely positive. This was, of course, somewhat disappointing as it would have been beneficial to find a replacement for Activin A, but it drove me to proceed screening in summer 2019.

## Summer 2019

Now familiar with the lab, I was well prepared to resume my experiments with greater confidence, efficiency, and independence. My plans to continue the AA screen changed when I learned that only up to 60% of our differentiated cells express NKX6.1, a marker of mature  $\beta$ -cells. After discussing with my mentor, I chose to pursue a compound that would induce NKX6.1 expression in our cells.

NKX6.1 expression is first observed in the pancreatic progenitor (PP) stage (figure 3). Consequently, I chose to screen differentiating cells in the PP stage for compounds that significantly upregulate NKX6.1 expression. Only 1mM and 0.1mM library concentrations were selected in duplicate.

## Results and Future Work

After treatment and NKX6.1 immunofluorescent staining, results were analyzed in the high content imager. To my excitement, a special subset of small molecules seemed to improve NKX6.1 induction in pancreatic progenitor cells (figure 11). These were ALK5 inhibitors and the CDK inhibitor BMS-265246 (BMS).

NKX6.1 expression level after different treatment

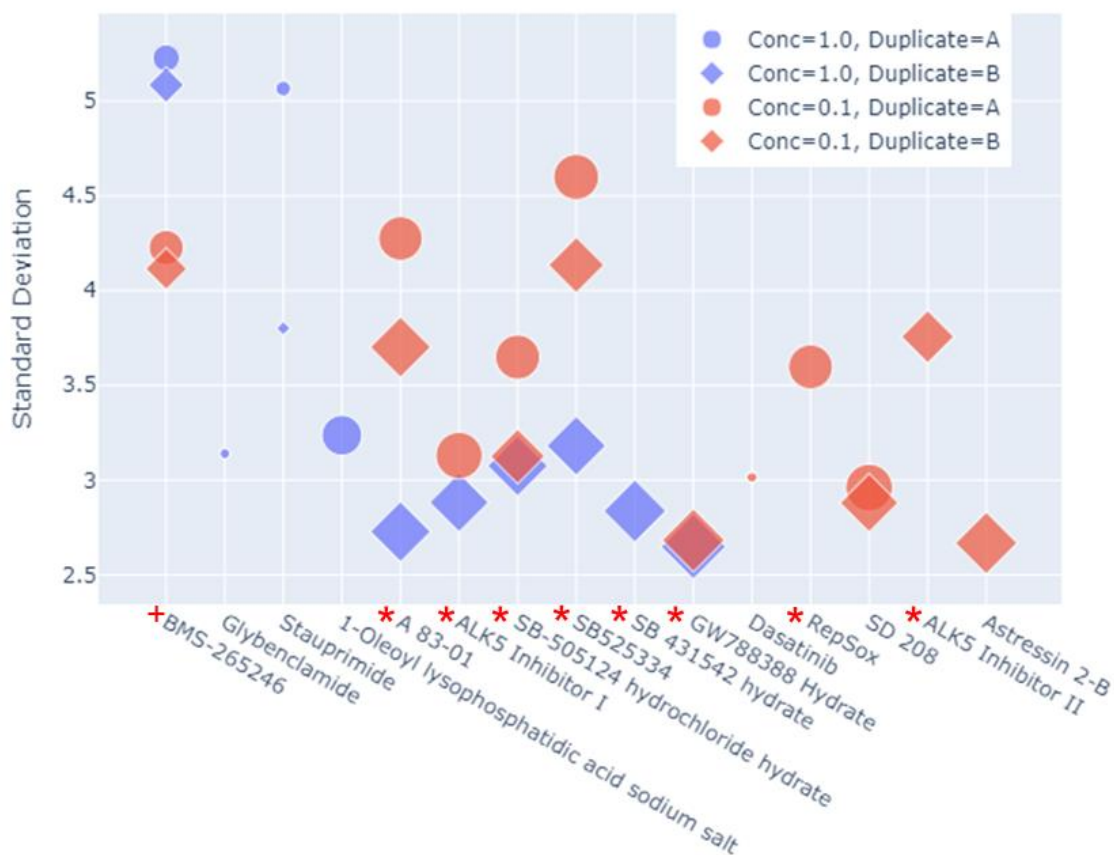


Figure 11: \*ALK inhibitor, + CDK1/2 inhibitor that selectively kills NKX6.1- cells.

My prior experience equipped me to conduct multiple chemical screens and validate hit compounds in secondary assays. Secondary screens confirmed that BMS significantly induced NKX6.1 expression and had the added benefit of selectively killing non SC- $\beta$  cell populations in a dose-dependent manner (figure 12)

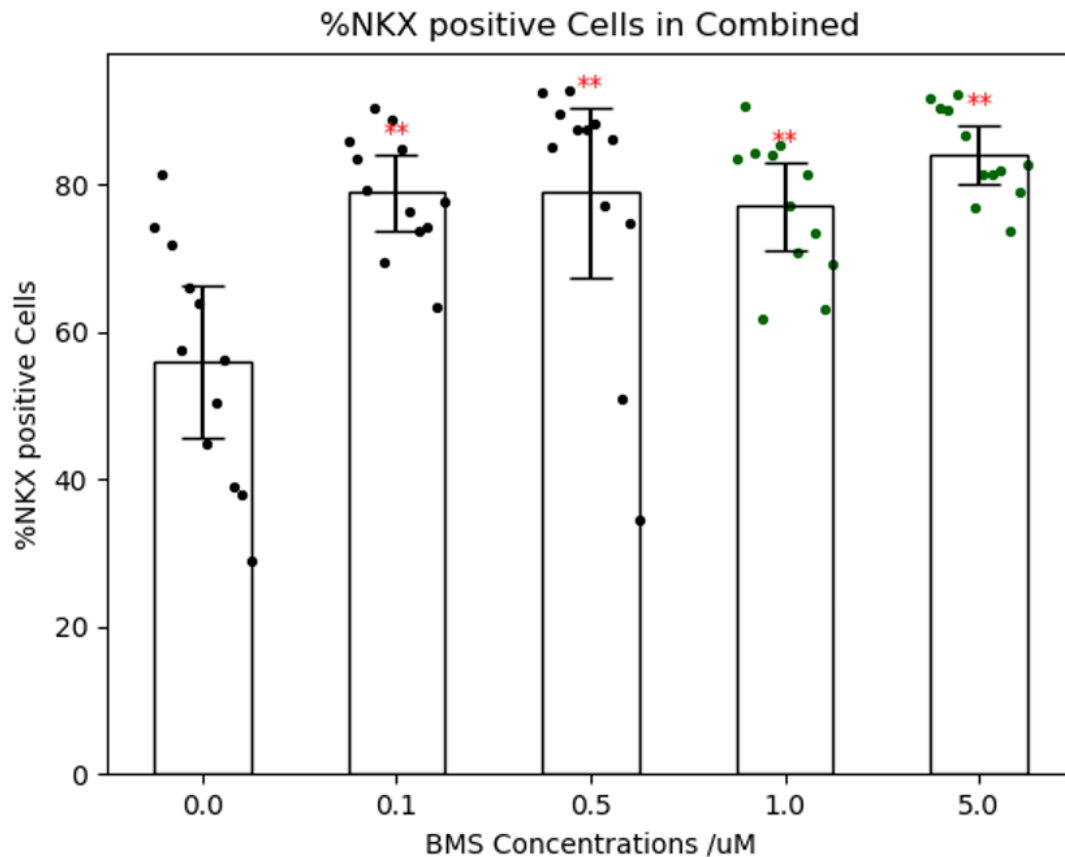


Figure 12: BMS demonstrates a dose-dependent induction of NKX6.1

In the future I plan to treat differentiating PP cells with my identified compounds and observe whether these cells secrete insulin in response to glucose stimulation. This would provide valuable insight into the mechanism(s) for NKX6.1 induction. Leveraging my engineering background, I am also automating the screening process with robotics so that more compounds can be tested with greater precision, efficiency, and resolution.

#### References

- [1] Center for Disease Control, "National Diabetes Statistics Report, 2017 Estimates of Diabetes and Its Burden in the United States Background," 2017.
- [2] F. W. Pagliuca et al., "Generation of Functional Human Pancreatic  $\beta$  Cells In Vitro," *Cell*, vol. 159, pp. 428–439, 2014

# Design Goals

The goal of this project is to semi-automate the pin transfer process in high content screening. This robot is intended for research labs whose purpose is not exclusively chemical screening. Our robot should enable the small-scale implementation of a chemical screening protocol that would permit labs to gain insight on novel interactions of small molecules on the cell types they study without sending samples to external labs or investing in more expensive equipment. The benefit of this is labs would gain more immediate results and spend less money than purchasing equivalent machines.

## Technical Requirements

- Dimensions:
  - The robot shall be no taller than 21 inches
  - The robot shall be no longer 46 inches
  - The robot shall be no deeper than 18 inches
- The system will be powered by a 120v wall outlet
- The robot will have an emergency shut-off button
- The robot will weigh less than 30lbs (light enough to pick up)
- The robot will be able to be sanitized with 70% isopropyl alcohol
- The robot will be able to fit in a biosafety cabinet
- The system will be controlled using a MCU board (Arduino or Texas Instruments)
- GUI supported by a color OLED screen:
  - Asks for Number of Well Plates
  - Allows the user to tell how deep to put the pins in the solution
  - Allows the user to tell how long to leave the pins in the solution
  - Alerts the User when the pin transfer is complete
  - ETA of current process in hh:mm:ss
  - Current step in cycle
  - Robot name
  - Organization logos
- The robot will have the following workspace requirements:
  - Should be able to handle a maximum of (16) Perkin Elmer 96 well plates and a minimum of 8. Extending to 384 well plates should also be a reasonable stretch goal.
  - Plates in tray
  - Plates out tray
  - 3 Wash reservoirs (200mL)
  - Pin drying fan
  - workspace rail to move plates along workspace
- The robot will use economical stepper motors. These will be used for:
  - workspace rail actuation
  - Pin transfer head movement (2-axis)
- All linear actuations will be done using belt driven linear actuators

- Multiple sensors will be used to keep all moving parts within the workspace
- A barcode reader may be implemented if time allows
- Cycle time should be less than 2 minutes (from input stack to output stack)
- The total cost of the system shall not exceed \$3000
- The pin transfer tool should have a failure rate of <1% (0 out of 100 tests)
- Failure happens when:
  - Pin tool touches bottom of well plate
  - Pin tool hits well plate
  - Plate is not correctly loaded onto workspace rail
  - Chemical splashes/spills out of well from shaking
- The MCU and sensors must be integrated into a PCB.
- A VP scientific pin transfer tool must be used for the chemical transfer process.
- The robot should be able to remove microplate lids using a suction cup actuator
- If a barcode is implemented it should be able to transmit the barcode information to other devices such as phones, pc either wirelessly or through USB.
- Cleaning solution reservoirs should exist that gravity feed new solution into the cleaning wells when empty
- Used cleaning solution should be vacuumed out by an external vacuum port that will be connected to the robot, operated by a servo valve.

## Build Plan

The build plan for this project is most importantly to meet all of our design technical objectives. This will ensure that our project will result in a novel design that can aid research labs that would like to implement small scale chemical screening experiments while minimizing resources required to conduct these experiments.

The conceptual thinking on how to start planning our build first is the very basic functions of the robot. The robot's primary objective is to transfer small molecules, growth factors, and other chemicals onto in vitro microplate cell cultures. This process is normally done manually if not done in a large specialized high throughput chemical screening or pharmaceutical lab. The robot will transfer these chemicals using a metal pin transfer tool. A pin transfer tool is a specialized device that uses surface tension of many metal pins to pick up a precise amount of liquid and then transfer that liquid to another container or vessel which is used as a very sensitive dilution step. This is often very necessary because it is more convenient to store stock chemicals in very concentrated library plates and then do a large ratio dilution in order to save solvent, space, and indirectly money.

On a small scale, this process can be done manually by a lab technician. The steps would look something like the following: The lab technician would take two input microstack plates, one stack being the chemical factors and the other stack being the cells to be treated. Each would be in a standardized microplate format. A microplate typically has 6, 12, 24, 48, 96, 384 or 1536 sample wells arranged in a 2:3 rectangular matrix. A manual



pin transfer tool would then be used to transfer the chemicals onto the cells. The operation of the pin transfer tool manually consists of covering the two plates of interest (chemical and cell microplates) with a key plate. This key plate matches a key pin on the pin transfer tool that makes sure that each pin that corresponds to a well in the microplate ends up in its respective well and that the tool is aligned with the microplate. After aligning the pin transfer tool with the microplate, the pin transfer tool is slowly and carefully lowered into the chemical factor microplate.

The technician has to be very careful to consistently dip the pins into the liquid at the same depth otherwise the amount of liquid transferred will change on each cycle of pin transfer which is extremely undesirable when conducting an experiment because you would have little consistency in the doses of chemicals the cells were treated with and would devalue experimental results. The pin tool is then dipped into the cultured live cell plate. In this action the user needs to again align the pin transfer tool with the microplate aligning key and then slowly dip the pin transfer tool into the live cell culture plate. In this step it is very crucial for the handler to not touch the bottom of the microplate with the pin transfer tool. This is because in most 2D cell cultures the cells are adhered to the bottom of the microplate and if the pin were to touch the bottom of the plate it could scrape the cells off and kill them.

After completing one cycle of the pin transfer process the pin transfer tool needs to go through an extensive wash process in order to ensure that there is no cross contamination between chemical or cell culture plates. If there is no wash step in between transfer cycles the media and cells could be deposited in the next chemical or cell plate to be used, or chemicals from a previous microplate could mix with the next chemical factor microplate which would not only spoil the current experiment but it could also spoil future experiments because the chemical factor microplates are frozen and used for many experiments. The wash process can be anywhere from one to four steps in a majority of lab protocols. The amount of wash steps depends on the types of chemicals being used and the solvents they are dissolved in within the chemical factor plates. Some chemicals are only miscible in certain solvents and these solvents have different properties. In some cases one chemical factor plate can have multiple solvent types and for that reason washing solutions may need to have many different chemical properties primarily based on the chemical polarity. For this reason it is common to use Deionized water, Alcohol, Dimethyl Sulfoxide, and other solvents in order to ensure the most chemicals are removed from the pin transfer tool before the next cycle begins. After a fixed number of cycles the washing solutions should be drained and reservoirs refilled to make sure no contaminants build up in the washing vessels, this number will change based on the protocol being used. After washing the pin transfer tool the pin tool needs to be dried before it can be used again in another pin transfer cycle. Manually this is normally done with a blotting paper, or washing with a low evaporation point liquid last such as isopropyl alcohol and waiting for it to evaporate off of the pin transfer tool. This part of the process can waste a lot of time and is a part of the robotic process that can be optimized a great deal.

The build plan is to replicate the overall procedure for manual operation of a pin transfer tool while incorporating the reliability and reproducibility of a robotic system into this

process. The first part of the pin transfer process that needs to be designed is how to store the input plates which consists of the live cell cultures as well as the chemical factor plates. Ideally, according to our technical specifications we would like our robot to be able to handle sixteen (16) chemical factor microplates and sixteen (16) live cell culture microplates maximum. These plates should be able to be accessed by the rest of the robotic system and unloaded onto an actuator that can move the plates to the pin transfer tool. The pin transfer tool will do what a manual pin transfer tool would do; it will dip into the chemical factor plate and then dip into the live cell culture plate. One key difference is that this step will not require a key plate to ensure that the pin transfer tool is in the correct location to ensure the pins each enter their respective well in the microplates. Instead this will be ensured using switches, sensors, and encoders that will let the robot know where the microplates reside in the X, Y, Z cartesian plane. This is one of the major benefits of introducing robotics into this process. Another benefit robotics give is that the pin tool needs to be dipped into the microplates at a very specific depth to ensure that the same amount of liquid is transferred in each transfer process. At the same time the pins cannot scrape the bottom of the microplate to ensure that cells are not knocked loose from the wells. A robot can make sure this never happens by also employing encoders in the linear actuator motors that make sure the pin transfer tool always reaches a fixed depth where it will stop and then raise out of the microplates. Maximum range of motion switches can also be implemented in the system that would prevent the robotic pin tool from hitting the bottom of the microplate in the event that the encoders were incorrect or failed.

The pin transfer tool will then need to move to another region of the robot where the cleaning solutions are located. The cleaning solutions should be in reservoirs that are large enough to hold a volume of liquid that can allow pins to reach a deep enough depth so the whole pin is cleaned of chemical residue. These reservoirs should also have the ability to be drained via vacuum port and refilled through a valve connecting the cleaning reservoirs to a cleaning solvent reserve where additional cleaning solution can be sourced from after each transfer/wash cycle. The pin transfer tool will dip into each of the cleaning solutions (as stated above in the manual pin transfer process one to four cleaning steps can be used) and the pin transfer tool will receive all of the benefits over the manual process listed above in the chemical transfer process in reference to the same cleaning depth being reached in every wash and to ensure that no cleaning step is accidentally missed which could cause contamination in the source chemical factor plates or the next live cell plate that will be operated on.

After the cleaning step the pins will need to be dried. In the manual process as discussed, the pin tool is blotted using blotting paper and usually the alcohol based cleaning solutions are used last to speed up the evaporation process. The downside of this process is that small lint pieces can get onto the pins unless very expensive scientific grade blotting paper is used. The benefit of using a robotic system is that a pressure air washer or a heating element and fan can be used to dry the pin transfer tool after the one to four wash steps. Not only does this allow for a better method to dry the pin transfer tool because it cannot generate lint particles on the pin transfer tool, but it also speeds up the drying process. This translates into much faster transfer cycle times. This is very valuable to

scientists because it will speed up their experiments while reducing workload on the lab technicians.

Finally the treated live cell plates will be transferred from the workplace to an export stack where the used chemical plates can be recovered and stored back in the freezer by the lab technician, and the live cell plates will be recovered and stored back in the incubator by the lab technician. By being able to handle many plates in the input and output plate stacks this will allow the lab technician to spend less time supervising the machine.

We anticipate that it will be very beneficial to automate these steps using robotics. The main benefits of automating these steps will be that it will ensure that the same amount of liquid is transferred in each cycle of the pin transfer process and that the experimental steps are easily repeatable which will validate any results obtained from the experiments using this device to treat cultured cells.

## Prototype Plan

The design for the prototype is split into these distinct parts: an input/output plate stacker, a workspace rail workspace that can handle wash steps and moving the current microplate being used (either chemical or live cell), a pin transfer head that can move up or down the z-axis as well as slide between two workspace rails in order to wash the pin tool and transfer chemicals, a drying element to dry the pin tool, and a structurally strong frame that is disinfectable and supports the pin tool.

As of now the prototype plan for the overall structure of the pin transfer robot is that it will be a gantry design made out of high quality stainless steel. This robotic gantry will support the pin transfer tool head which will be actuated by two encoded stepper motors. The stepper motors will allow the robot to move the pin transfer tool in the up and down directions as well as over two parallel conveyor rails. The plates will be moved along using belt driven linear actuators as seen in Figure 4 below. One rail will house the cleaning reservoirs that are used to clean the pin tool and the other workspace rail will house the live cell plates or chemical factor plates.



*Figure 4. Linear belt driven actuator Solid Works CAD model*

There will be an embedded drying fan in the base of the workspace. This will dry the pin transfer tool faster and make sure that no liquid will contaminate the next pin transfer cycle. The drying fan will be activated by a lever which will be pushed down when the pin transfer tool pushes down on it. The fan will stay activated for a fixed amount of time which will be determined experimentally.

There will be an OLED display panel on the front of the robot that will show the operator many things including the estimated remaining time on the process and will allow the operator to select the amount of microplates being fed into each input tray. The user will also be able to select here the amount of time that the pin transfer tool will be in the liquid solutions and how deep, and therefore how much, the solution will be transferred.

The prototype design will be powered by a microcontroller that could be either Texas Instruments or Arduino. The microcontroller chip is soldered onto a PCB with the GPIO pins connecting to the sensors and switches. This information is computed in the MCU to determine where the pin transfer tool is and what actuations should take place. The PCB will be stored underneath the workspace in the base of the robot.

# Block Diagrams

Below in Figure 13 is the Hardware block diagram that we will be using for this project.

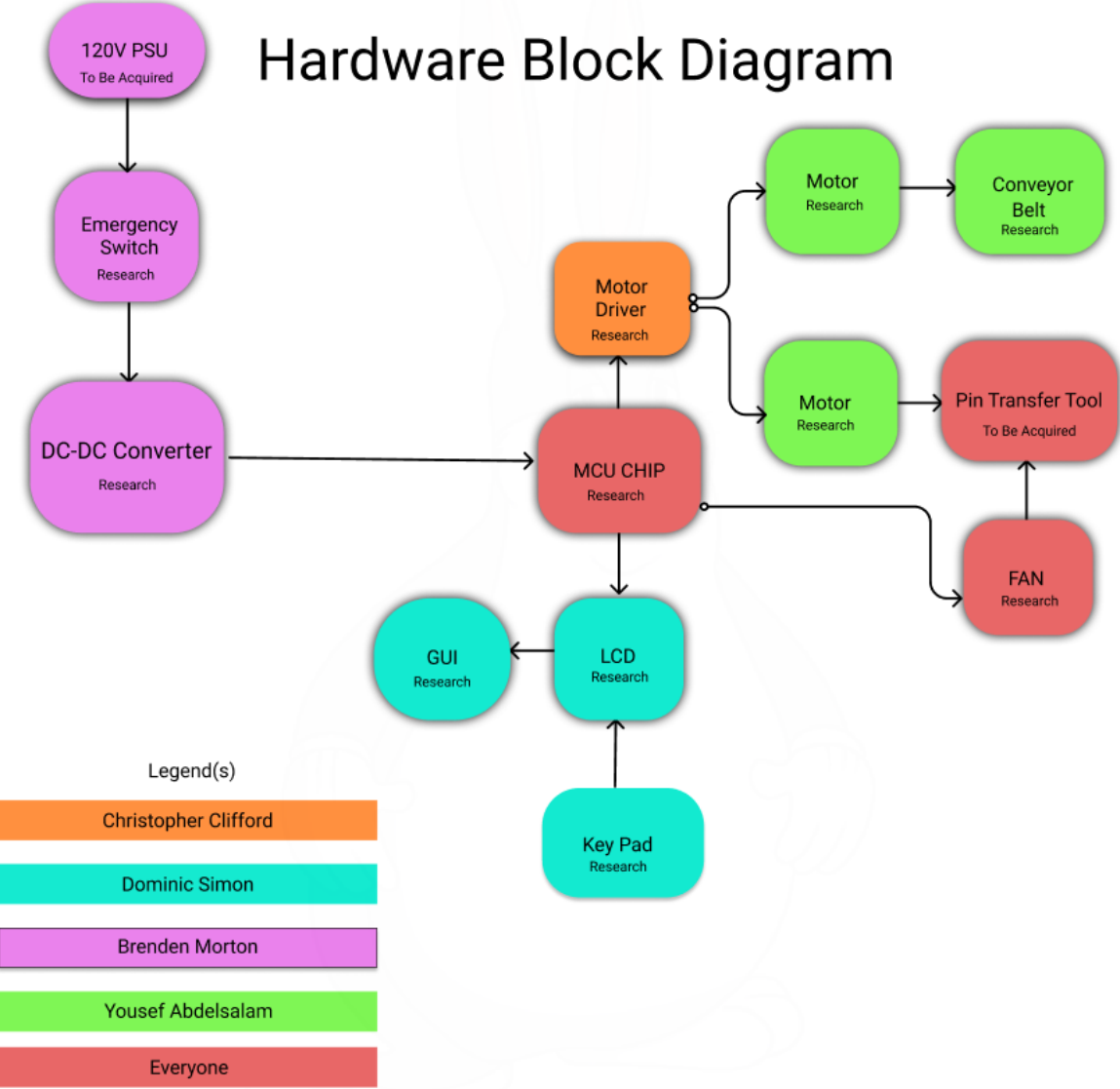


Figure 13: Hardware Block Diagram

Below in Figure 14 is the design and flow of the software for this project.

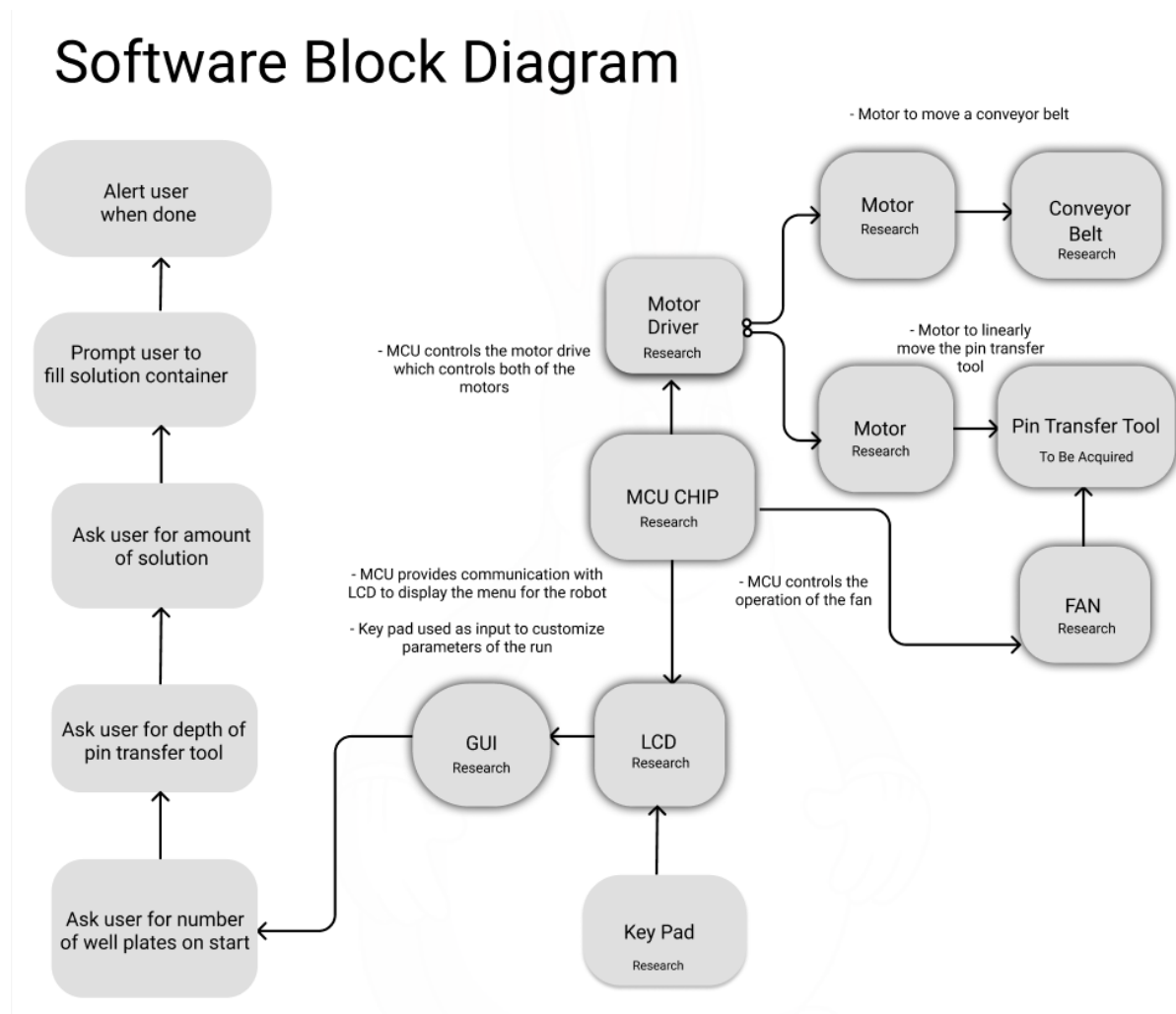


Figure 14: Software Block Diagram

## Estimated Budgeting and Finance

Part	Description	Quantity	Unit Cost	Total Cost
-	Pin Transfer tool	1	\$0 - \$200	\$0 - \$200
Arduino Atmega 2560	MCU	1	\$0 - \$20	\$0 - \$20
OpenBuild	workspace rails	1	\$100 - \$130	\$500
TI	12V Fan	1	\$10	\$10
Youngneer	12V Relay (8 pc)	1	\$11.99	\$11.99
MEANWELL 24V	Power Supply Unit	1	~\$100	~\$100
TI	DC-DC	1	\$3	\$6
JLCPCB	PCB	5	\$20	\$20
BIQU A4988	Motor Driver	2	\$9.50	\$19.00
Usongshin e 17HS4401 S	Motor	2	\$9.97	\$19.94
Any LCD	LCD (16x4)	1	~\$15	~\$15
COM-1466	Key pad	1	\$4.50	\$4.50

**Table 1: Cost per Item**

Number of Team Members	4
Sponsor Contribution	Unknown
Total Cost of Project Based on Current Prices	~\$400
Contribution per team member	~\$100

**Table 2: Cost per Member**

## Hardware Design

### CAD Software

Computer-aided design or CAD is a way of using computers to assist in the development of design, simulation, manufacturing a product or work. CAD software will be used in this project to develop the schematic of the electrical wiring of the PCB. In addition to that, the CAD software will be able to generate the board file from the electrical schematic. This greatly simplifies the process of creating a board file since it is much easier to create the electrical schematic.

### Eagle

We chose to use Autodesk's Eagle CAD PCB software to develop the circuit schematic as well as the PCB for this project. Other than having prior experience with this CAD software, Eagle is a well-rounded and feature-rich software that has many useful tools for creating schematics and easily converting those schematics to board files for PCB fabrication. The Eagle PCB design software allows the user to first design the electrical schematic of the desired PCB and then convert the schematic to a board file. In doing so, the circuit designer does not have to worry about initially making all of the wiring connections on the board file of the PCB. Instead, Eagle allows for the user to first design the electrical schematic and then automatically creates the board file. After that, the circuit designer can route the components together properly.

### Schematic



Using Eagle, the electrical circuit schematic was developed by first placing the main components of the board such as the ATMEGA 2560 microcontroller IC and the secondary microcontroller IC ATMEGA16U2-MU chip. One by one, we added more components to the circuit such as voltage regulators for maintaining a constant 5V or 3.3V for the logic on the ICs. Following some of the recommended datasheets **[ATMEGA2560 DATASHEET]** **[ATMEGA16U2 DATASHEET]** for the ATMEGA 2560 as well as the ATMEGA16U2, we created the remainder of the circuit connections. Some of the other features that we used in this schematic include the net ports. Net ports allow wires to be virtually connected, such that in the schematic they are not physically connected through the junction of two adjacent wires but by a labelling scheme that is provided by Eagle. These net ports still allow for the electrical connections to be present in the board file for the schematic. The main purpose of this feature is to allow for the wiring diagram to appear cleaner.

Shown below in Figure 16 is the electrical schematic of the microcontroller IC (ATMEGA2560) which is the main IC of the machine.

This will control all of the GPIOs of the machine including some of the following:

- 4 Stepper motor drivers
- TFT LCD screen
- Keypad
- Linear Actuator

Later in this document, we will discuss the reasoning for selecting the ATMEGA2560 IC for this project, but one of the main reasons is because of the amount of accessible GPIO pins.

# ATMEGA 2560

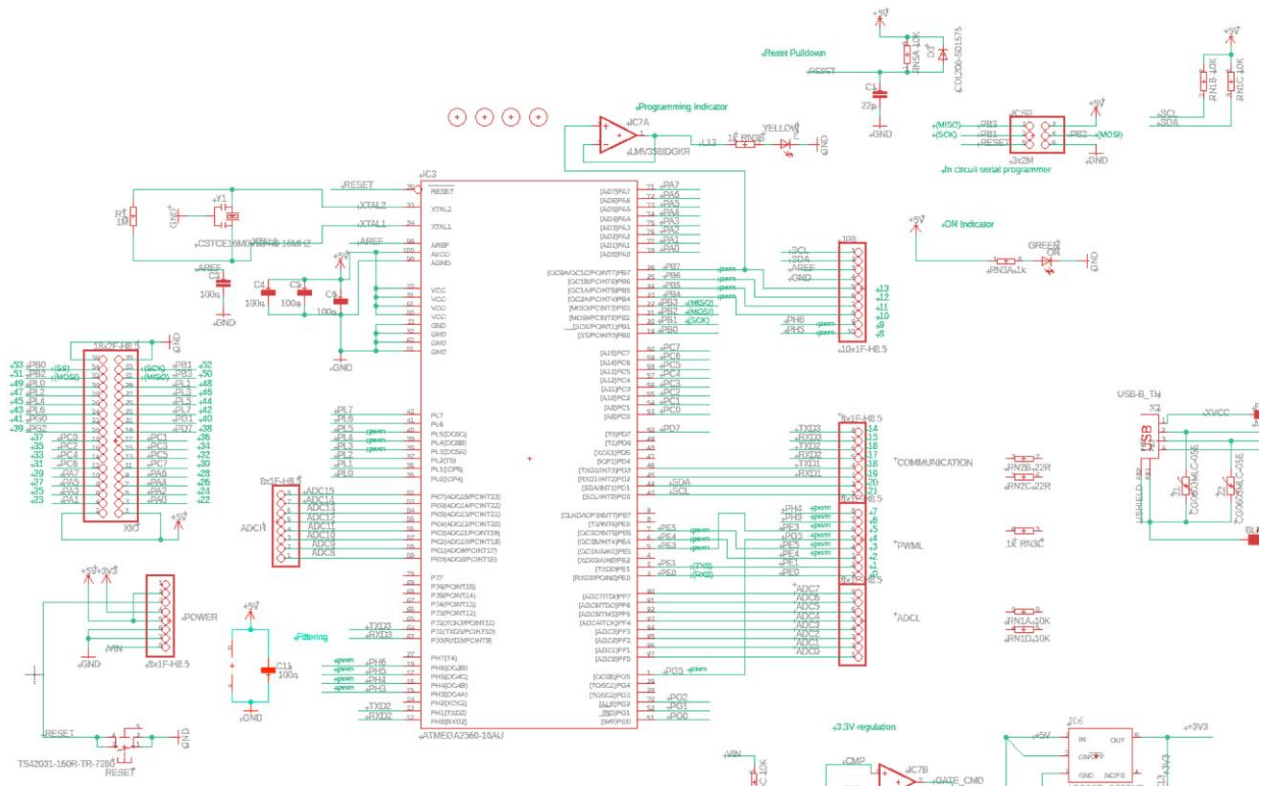


Figure 16 - Schematic of ATMEGA 2560 electrical wiring

Shown below in Figure 17 is the schematic for the power regulation for the logic of the microcontroller IC. The microcontroller requires both 5V and 3.3V. As such, there is a regulated 3.3V and a regulated 5V provided by separate 5V and 3.3V DC voltage regulators. The schematic below represents the electrical wiring for these two voltage regulators. These designs were modeled after the designs from the ATMEGA 2560 datasheet [ATMEGA2560 DATASHEET] since these are tried and tested designs that work well with this specific microcontroller IC.

# Power

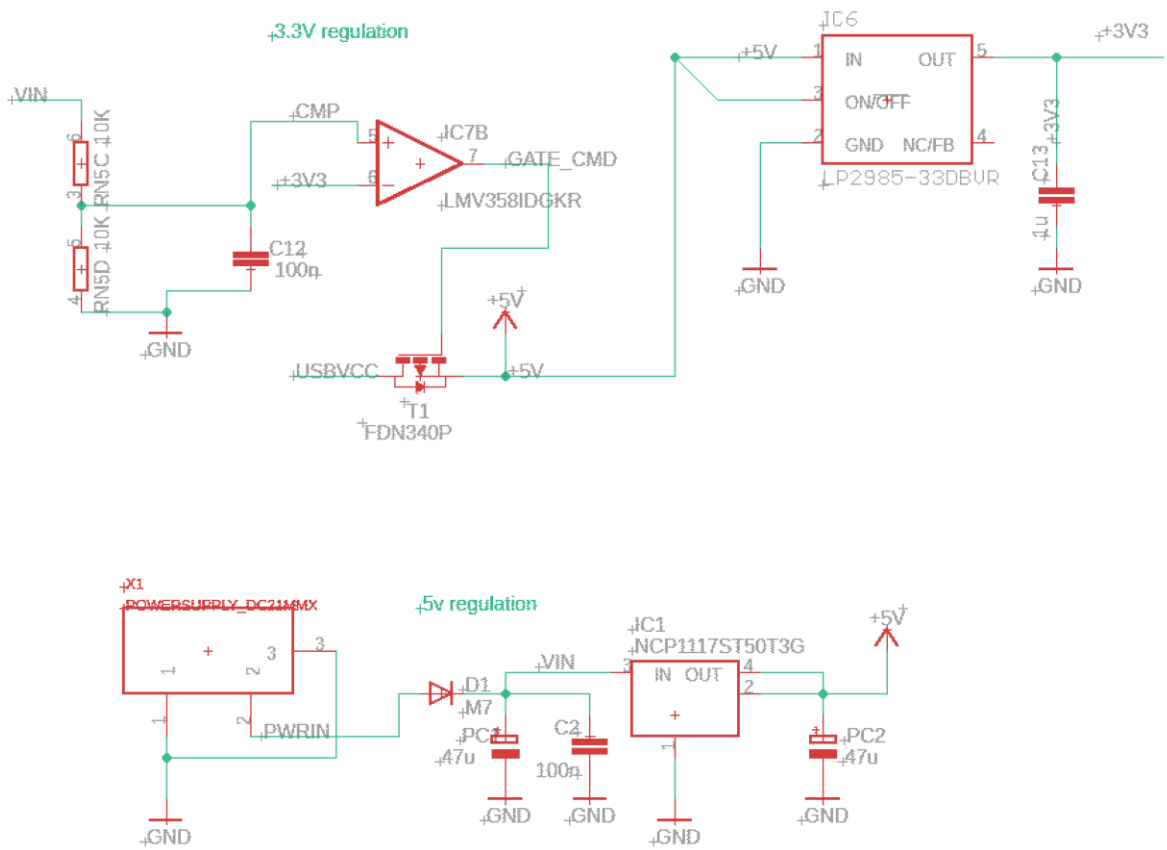


Figure 17 - Schematic of the power regulation for the PCB

# Bill of Materials

## BOM

Name	Footprint	Quantity	Price
22p	C0603-ROUND	3	0.21
100n	C0603-ROUND	10	0.263
1u	C0603-ROUND	2	0.28
M7	SMB	1	0.55
CD1206-S01575	MINIMELF	2	0.45
MF-MSMF050-2 500mA	L1812	1	0.32
NCP1117ST50T3G	SOT223	1	0.48
ATMEGA2560-16AU	TQFP100	1	13.68
ATMEGA16U2-MU	MLF32	1	2.83
LP2985-33DBVR	SOT23-DBV	1	0.53
LMV358IDGKR	MSOP08	1	0.50
BLM21	805	1	0.63
GREEN	CHIP-LED0805	1	0.26
47u	PANASONIC_D	2	0.44
8x1F-H8.5	1X08	5	0.69
1M	R0603-ROUND	2	0.55
TS42031-160R-TR-726	TS42	1	0.36
SJ	SJ	2	0.54
22R	CAY16	1	0.23
1k	CAY16	2	0.44
10K	CAT16	2	0.66
YELLOW	CHIP-LED0805	3	0.27
FDN340P	SOT-23	1	0.69
POWERSUPPLY_DC21 MMX	POWERSUPPLY_DC- 21MM	1	0.96
USB-B_TH	PN61729	1	0.88
18x2F-H8.5	2X18	1	5
16MHz	QS	1	0.75
CG0603MLC-05E	CT/CN0603	2	0.49

Table 4 - BOM for PCB

## PCB Fabrication

After creating the PCB from the electrical schematic, the board design can be fabricated from one of the many PCB fabrication companies. We chose to use JLCPCB since they

provide a cheap solution for printing out PCBs. Additionally, JLCPCB can send newly printed PCBs to the final destination within a few weeks.

## Software Design

The following sections will be about the software of the Pin Transfer Tool. The explanation of the code will be broken into two parts: one for the user interface and one for pin transfer process. However, the actual code will have these parts spread out between more functions/methods to have better readability and follow the principle of abstraction.

### User Interface Algorithm

The user interface will turn on as soon as the Pin Transfer Tool is powered on. A greeting message will briefly be displayed. After the greeting disappears, a new message reminding the user to follow proper lab safety procedures will appear with a prompt for the user to press the '\*' button on the keypad. This message will continue to be displayed until the user presses the '\*' button.

Once the button is pressed, a new message prompting the user to enter the number of well plates to be used will be displayed. There will be two additional messages under the prompt: one to tell the user that the '\*' button confirms their number of well plates and another to tell the user that the '#' button clears their number of well plates. Each time the user presses a button on the keypad to enter a number, it will be displayed at the bottom of the screen. The user will be limited to 2 numbers displayed on the screen. After two numbers are displayed, no more numbers will be available for display and the user will have to clear the numbers entered or confirm their selection. If the number entered exceeds the maximum processable well plate amount or is empty, a new message will be displayed informing the user that their selection is invalid and the reason why it is invalid. Below that message, there will be a prompt telling the user to press '\*' to return to selecting the amount of well plates they want processed.

If a valid number of well plates are selected for processing, the number of well plates selected will be prominently displayed on the screen with a new message asking the user to confirm that the displayed number is the correct number by pressing the '\*' button or, if the number is incorrect, to reselect the number of well plates by pressing the '#' button. If the '#' button is pressed, the message prompting the user to input the number of well plates they want processed will reappear. If the '\*' button is pressed, the Pin Transfer Tool will begin operation.

During operation, a message telling the user that the well plates are being processed will be displayed. Once all plates have been processed, a new message will be displayed that lets the user know that the operation has ended. The user will also be prompted to press the '\*' button. When the '\*' button is pressed, the user will be asked if they would like to do another batch by pressing '\*' or power off by pressing '#.' If '\*' is pressed, number



The pin transfer step begins once the user has finished interacting with the user interface or after the end of the previous wash step. On the first pin transfer process of the batch, the pin tool rail and actuator will fully retract. Normally, this will be done at the end of the wash step, but it is also done at the beginning of the first process of the batch to ensure that it is starting from the correct position. First, a flag will be checked to see if this is the first process of the batch. If it is not, move on to the input stacking algorithm. If it is, the pin tool rail and actuator need to be reset. First, the pin tool linear actuator will begin retracting. The code will pause execution temporarily. When code execution resumes, the actuator will stop retracting. The actuator will be full retracted. Next, the pin tool rail will begin retracting. Code execution will be delayed. When the delay ends, the rail will stop retracting. The pin tool rail will be fully retracted.

The input stacking algorithm will be executed. Finally, the pin transfer process can begin. The pin tool rail will begin extending. Code execution will be delayed. Once code execution resumes, the rail will stop extending. The pin tool is now directly over the chemical workspace rail and the chemical plate. The pin tool actuator begins extending. Code execution temporarily stops. When it resumes, the extending stops. The pin tool is now in the chemical plate. Another code execution delay will occur. This is to ensure that the pin tool successfully takes up the chemicals it needs. Once the delay is finished, the pin tool actuator will begin retracting. Code execution will temporarily stop. When the execution resumes, the actuator stops retracting. The pin tool actuator is now fully retracted. The pin tool has all of the chemicals it needs.

Now, the pin tool needs to deposit the chemicals into the cell well plate. The pin tool rail will begin extending. A code execution delay will occur. When the delay ends, the rail will stop extending. The pin tool will now be directly over the cell workspace rail and the cell plate. The pin tool actuator will begin extending. A code execution delay will occur. When it resumes, the actuator will stop extending. The pin tool is now putting the chemicals into the cell well plate. The pin tool actuator begins retracting. Code execution temporarily stops. When it resumes, the actuator stops retracting. If this is the last pin transfer process of the batch, then reset the first process check flag. The next part is the washing step. Below in Figure 19 depicts the flow control for the pin tool.

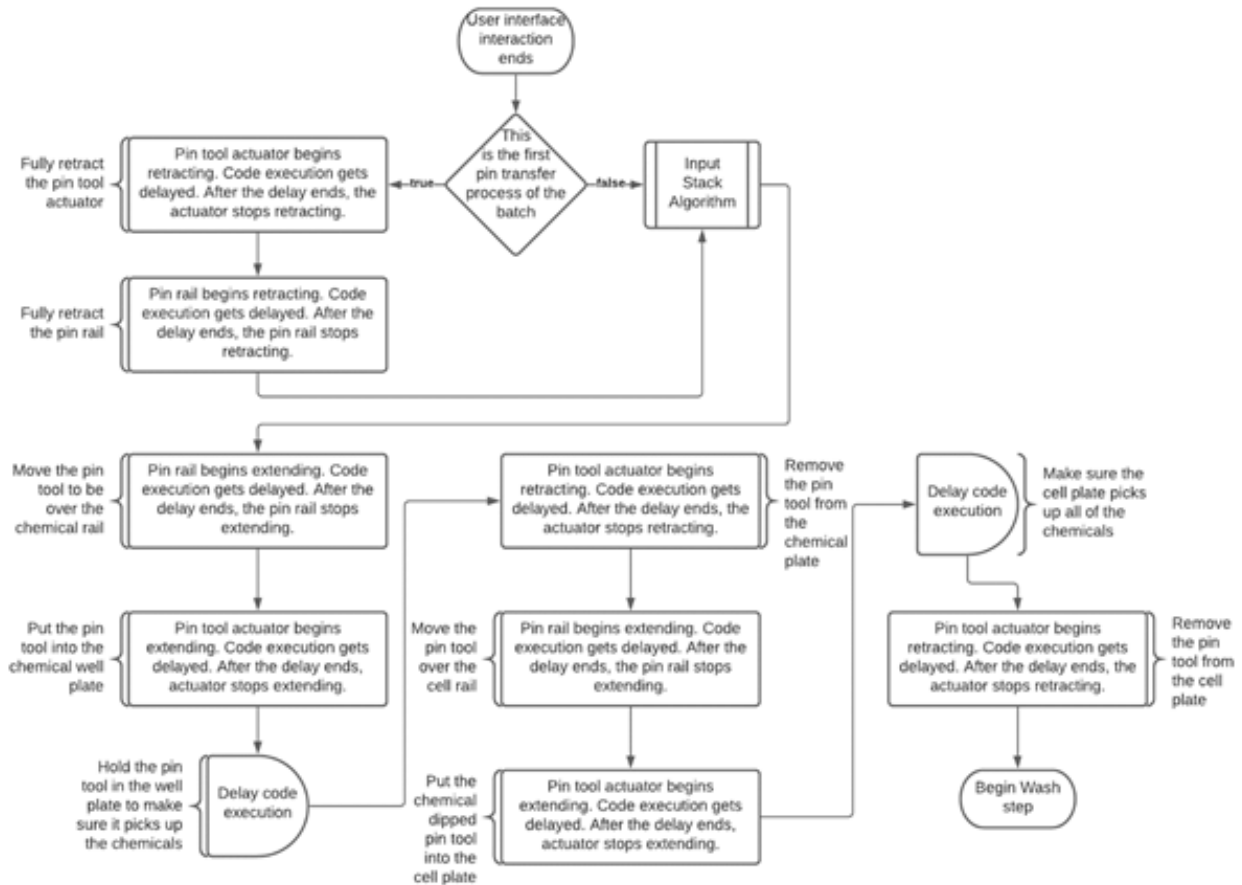


Figure 19 - Pin transfer step algorithm flowchart

## Stacking Steps

This step starts after the pin rail and pin tool actuator resets, if they need to. Like the rail and actuator, a flag will be checked to see if this is the first pin transfer process of the batch to avoid any parts being in the wrong place. If it is the first process, then one of the stacks will begin retracting. A delay in code execution will occur. Once code execution resumes, the stack will stop retracting. That stack will be fully retracted so that the topmost shelf is level with the workspace rail. This will repeat for each stack. If this is not the first pin transfer process of the batch, then each stack will need to move to the next well plate. One stack will begin extending. A code execution delay will occur. Once the delay ends, the stack will stop extending. The well plate below the previously used well plate will be level with the workspace rail. Both stacks will do this.

Both stacks now have a well plate flush with the rails, regardless of whether or not this is the first pin transfer process of the batch. The well plates need to be moved under the pin tool for the pin transfer process. A pin is activated behind the current well plate on the stack to push the well plate on to the workspace rail. This happens for both stacks. Both the chemical and cell well plates are now on their respective rails. One of the rails starts extending. The code execution temporarily stops. Once it resumes, the rail stops





The washing step begins once the linear actuator that the pin tool is attached to fully retracts after the chemicals on the pin tool have been transferred to the cell plate. The pin tool rail needs to be positioned over the cleaning rail and the cleaning rail needs to have the first washing reservoir lined up with the pin tool. The rail that the pin tool is attached to will then begin extending. Further code execution stops temporarily. This will allow for the pin tool to be directly over the wash rail. Once code execution resumes, the pin tool will stop its extension. The wash rail will then begin retracting. Code execution will stop temporarily so that the wash rail has enough time to fully retract. Once code execution resumes, the wash rail will stop retracting. The purpose of this is to reset the position of the wash rail from the previous cycle. It will then begin extending. Again, code execution will stop temporarily. Once the delay has ended, the wash rail will stop extension. At this point, the first wash reservoir should be directly under the pin tool.

The pin tool needs to now be cleaned in the first reservoir. The linear actuator the pin tool is attached to will begin extending. Code execution will stop temporarily. The actuator will stop extending once code execution resumes. Code execution will again be delayed so that the pin tool will be able to fully soak in the cleaning solution from the reservoir. Once code execution resumes, the pin tool linear actuator will begin retracting. Another code delay will occur so that the actuator can fully retract. When code execution resumes, the actuator will stop retracting. The wash rail will then begin retracting. Code execution will be delayed. When code execution resumes, the wash rail will stop moving. This is so that the next reservoir will be placed under the pin tool.

The process of the pin tool dipping into the cleaning solution in a reservoir and moving to the next reservoir will occur for each of the three reservoirs. After the third reservoir has been completed, the pin tool will be over nothing on the wash rail. There will be a fan mounted on the base of the Pin Transfer Tool at the opposite end of the wash rail than where the reservoirs are. The pin tool actuator will extend, delay code executions for a short time, and then stop extending. This will put the pin tool on level with the fan. The fan will then receive power and code execution will be delayed. This will allow the pin tool to be fully dried from all of the cleaning solutions it was put in.

Once code execution resumes, the fan will lose power. The pin tool actuator will then begin retracting. Code execution will then be delayed. When the delay ends, the actuator will stop retracting. The pin tool actuator will be fully retracted. The pin tool rail will then begin retracting. A code execution delay will occur. Once the delay ends, the rail will stop retracting. The pin tool will now be directly over the chemical workspace rail. This ends the washing step. Shown below is the Pin transfer flow control in Figure 21:

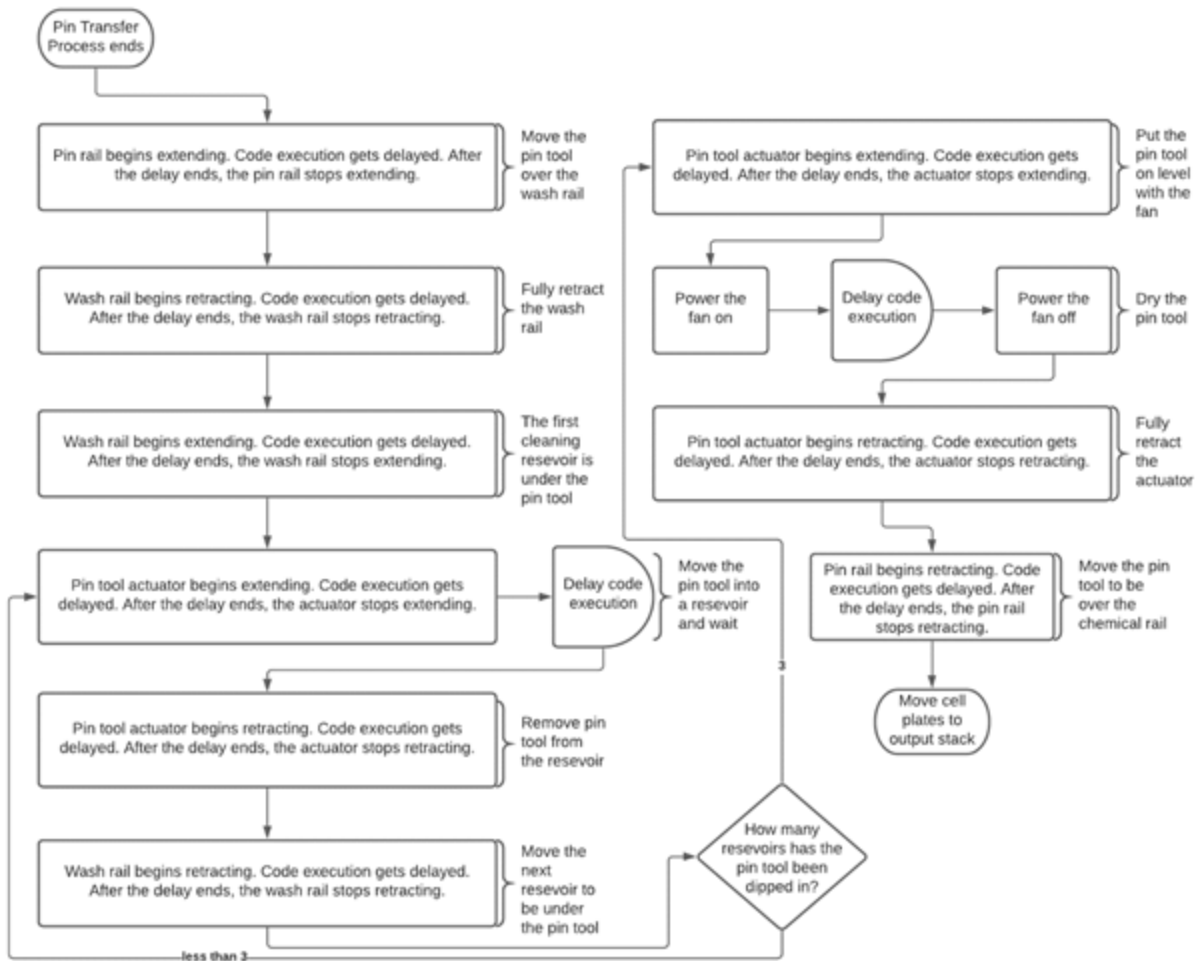


Figure 21 - Washing Algorithm Flowchart

## Microcontroller Connection Functions

Functions from the Wire library will be used to connect to the microcontroller. From this library, the following functions will be used: **[PYSERIAL]**

Library	Function Signature	Functionality
Wire	Serial.begin(int dataRate)	Open a connection to the microcontroller using I <sup>2</sup> C
Wire	Serial.close()	Close the previously opened connection to the microcontroller

Table 5 - Wire API

## User Interface Functions

The OLED will be written to using the Adafruit\_GFX and OLED specific libraries. From these libraries, the following functions will be used: **[OLED FUNCTIONS]**

Library	Function Signature	Functionality
OLED Library	display(int width, int height, connection reference, int GPIO)	Initialize the display with its height, width, I <sup>2</sup> C connection, and GPIO number
Adafruit_GFX	setTextSize(int size)	Set the size of the font that will be written
Adafruit_GFX	setCursor(int x, int y)	Set the starting pixel position of the text
Adafruit_GFX	print(string message)	Prints the specified message at the previously specified cursor point
Adafruit_GFX	display()	Pushes all the changes made up to this function call to the OLED
Adafruit_GFX	clearDisplay()	Deactivates all pixels
Adafruit_GFX	startscrollleft(int startRow, int stopRow)	Scroll the text left across the screen. Used to avoid burnout
Adafruit_GFX	startscrollright(int startRow, int stopRow)	Scroll the text right across the screen. Used to avoid burnout

Adafruit_GFX	startscrollleft(int startRow, int stopRow)	Scroll the text diagonal and left across the screen. Used to avoid burnout
Adafruit_GFX	startscrollright(int startRow, int stopRow)	Scroll the text diagonal and right across the screen. Used to avoid burnout

*Table 6 - OLED API*

The keypad will take in input using the Keypad library. The following functions will be used to take input from the keypad:

Library	Function Signature	Functionality
Keypad	makeKeymap(char keys[])	Turn the 2D array of key names into a keymap
Keypad	Keypad(keymap, row pins, column pins, int numRows, int numColumns)	Constructor to create a keymap
Keypad	getKey()	Detects if a key has been pressed and returns the key symbol corresponding to the one that has been pressed

*Table 7 - Keypad API*

## Motor Functions

Motors and motor drivers will be used in many different areas. Each of them will be controlled using built-in Arduino libraries. The following functions will be used:

Library	Function Signature	Functionality
<b>Built-in</b>	digitalWrite(int pinNum, HIGH/LOW)	Depending on which pin gets HIGH and which get LOW, the actuator will extend or retract

<b>Built-in</b>	analogWrite(int pinNum, int speed)	The actuator will extend or retract at the specified speed. The pin specified HIGH by digitalWrite must receive a nonzero speed.
-----------------	------------------------------------	--

*Table 8 - Motor API*

## Class Diagrams and Data Structures

No code has been written so far, but based on how the functions have been explained, there should not be any need for complex data structures. The data storage needed should be covered by primitive variables. Also, based on prior knowledge of how Arduino works, no classes should be necessary. A function will be written for each step of the process and those functions will be put in Arduino's loop function, which is continuously called while the Arduino chip has power.

## Standards

Engineering standards are best-practice guidelines created by engineering organizations and companies to aid engineers in the design of their projects. Standards are meant to increase the quality of design aspects of a project, such as efficiency and safety. The following sections will explain each standard that was followed to create the Pin Transfer Tool as well as why each standard is relevant to the Pin Transfer Tool.

## Code of Federal Regulations, Title 40 Part 262

This is a federal standard for hazardous waste storage and transportation for generators of hazardous waste. The specifics of what is and is not a hazardous waste can be found in the previous part – part 261. In essence, a hazardous waste is any consumer or industrial grade byproduct that could cause substantial harm to humans or the environment. **[WASTE][PART 261]**

The first important information of this standard is the classification of hazardous waste generators – any entity that produces hazardous waste. There are three different categories that a generator can fall into: very small quantity, small quantity, and large quantity. These designations can change month to month since they are based on the generator's monthly production of hazardous waste. Very small quantity generators are those that produce less than 1 kilogram of acute hazardous waste, less than 100 kilograms of non-acute hazardous waste, and less than 100 kilograms of acute hazardous waste cleanup byproducts in one month. Small quantity generators are those that produce less than 1 kilogram of acute hazardous waste, greater than 100 kilograms and less than 1000 kilograms of non-acute hazardous waste, and less than 100 kilograms of acute hazardous waste cleanup byproducts in one month. Large quantity generators are those

that produce more than 1 kilogram of acute hazardous waste, greater than 1000 kilograms of non-acute hazardous waste, or more than 100 kilograms of acute hazardous waste cleanup byproducts in one month. These designations are important since each different generator types are occasionally subject to different rules. **[PART 262]**

The next set of important information of this standard is instructions on how to store hazardous waste. Hazardous wastes should be stored in completely sealed containers. If a container is in poor condition, the hazardous waste should be moved to a container that is not in poor condition. Hazardous wastes that could react together should not be stored together and all containers should be thoroughly washed once they are emptied so that no new waste reacts with any residuals of the previously stored waste. Additionally, any possibly reactive wastes should not be stored near each other. All containers must have a label clearly marking it as a hazardous waste container as well as the type of hazardous waste stored within. Any time a generator has more than 55 gallons of non-acute hazardous waste, 1 quart of acute liquid hazardous waste, or 1 kilogram of acute solid waste, they must mark each of the excess waste containers with the date they started to be store and dispose of the excess hazardous waste within 3 days of the start of the excess buildup. **[PART 262]**

The next important information of this standard is for any generators that wish to transport the hazardous waste to offsite waste storage or disposal sites themselves. These generators must send an application to the Environmental Protection Agency (EPA) to do so. Once the generator has gained permission from the EPA to do so, they must fill out an EPA manifest every time hazardous waste is transported and notify the EPA by sending them a copy of that manifest. **[PART 262]**

Hazardous waste that is being transported is subject to rules and regulations put forth by the Department of Transportation (DoT), which can be found in the Code of Federal Regulations, Title 49. Part 173 of that title explains how hazardous materials, including hazardous wastes, should be transported. It lists a number of different options, such as salvage drums. The previously mentioned manifest must also be signed before the waste can be transported. Finally, each hazardous waste container must be marked with a number of identifying information, such as the generator's name, address, and EPA identification number. **[PART 172][PART 173]**

The final set of important information is about recordkeeping. All generators must keep all EPA manifests from the transportation of hazardous waste as well as a signed copy of the manifest from the offsite facility for three years. Any generator that is considered a large scale generator is required to complete a form on their hazardous waste disposal every other year. **[PART 262]**

While the design of the Pin Transfer Tool does not require any materials that will need to be treated as hazardous waste, users of the Pin Transfer Tool will be inputting chemicals that could have byproducts that are considered hazardous waste. Anyone using chemicals that could produce hazardous waste should know of the proper disposal procedures, but our design – the Pin Transfer Tool – is enabling people to possibly create

hazardous waste. Therefore, we believe it is our ethical responsibility to inform users how to dispose of hazardous waste.

## PEP 8 and C++ Core Guidelines

The Python Enhancement Proposal (PEP) 8 is a set of coding standards for Python written by Python developers. The C++ Core Guidelines is an ongoing project by Bjarne Stroustrup, the creator of C++, to create a complete set of C++ standards. Both sets of standards outline best practices for readability, efficiency, and use for their respective languages. The standards cover a wide array of issues such as proper indentation, avoiding redundant code, and arithmetic rules. These two standards were chosen because they were both created by the developers of the languages. It is safe to assume that the developer of a language has some of the most intimate knowledge of that language, so any guidelines they produce for said language would be advisable to follow.

**[C++ Core][PEP 8]**

The main part of these standards that apply to this team is the standard syntax style. People sometimes have their own quirks when writing code or learned to code with a different set of standards, so reading others code can be more difficult than reading one's own code. If everyone is following the same standards, less time will be wasted trying to understand what others wrote, increasing work efficiency. Also, having code with uniform syntax will increase the readability for anyone outside of the design group that attempts to understand the code. Having standards for efficiency and use are still helpful. Writing efficient code is just a good practice and also helps with readability. Understanding certain quirks of the language, such as how different number types interact, will help avoid mistakes that would have been made otherwise.

## Robot Operations

This robot is designed with the intention of speeding the pin transfer process commonly used in chemical screening experiments. The operation of this robot will vary greatly with the requirements of your lab's standard operating procedures. The robot is made with this flexibility in mind. The following sections will detail the operation of the robot's subsystems and how they can adapt to the needs of the operator. There will be a brief description of the subsystem followed by a step by step operation of the subsystem and a description of the electrical or schematics of the system to give an understanding of how the system functions internally.

## Input Microplate Stack

The pin transfer robot is designed to work with standard form factor microplates. The robot can be modified to work with many different dimensions by contacting the manufacturers of the Pin Transfer Robot, but this particular prototype is designed around the dimension 11.6 mm x 127.75 mm x 85.34 mm (height x length x width). A microplate



typically has 6, 12, 24, 48, 96, 384 or 1536 sample wells arranged in a 2:3 rectangular matrix (Perkin Elmer).

The dimensions of the microplate is important to match the dimensions considered when building your pin transfer robot for multiple reasons. It is important that the pin transfer robot knows the exact coordinates of the microplates at any given time in the pin transfer process. This is because if the robot is incorrect in its assumption of the microplate's position the pin transfer head could be damaged by being crushed into the microplate and the microplate and its contents could also be damaged. The dimensions are also important to ensure that the pin transfer tool head properly lowers into the microplate at the correct depth. Incorrect microplate dimensions could lead to the pin transfer tool either bottoming out on the microplate and damaging the pin transfer tool or it could lead to a shallow dip into the liquid contents of the microplate which would transfer an incorrect amount of liquid contents without notifying the operator to the error.

The microplate dimensions being used are also extremely important to ensure that the microplates fit into the input and output microplate stacks. The input microplate stack will be a shelf-like unit that is built to house all of the chemicals and incubated cells residing in microplates. If the microplates are an incompatible size then they may not fit in the shelf or may fall out of the shelf or be crushed when the shelving unit moves up or down on its linear rail to load or unload a microplate. For this reason only the correct dimension microplate should be used when operating the pin transfer robot to avoid damaging and part of the robot or any of the microplates and to avoid biological or chemical spills.

There will be two input stacks on the pin transfer robots workspace when the robot is configured for a eight microplate max handling capacity. One input microplate stack is to hold up to eight microplates which contain the chemicals to be transferred. These microplates will have any assortment of chemicals all within the wells of the microplates. All of these chemicals will be transferred onto the live cell plates at once with the pin transfer tool so it is necessary that the well layout of these chemical microplates exactly matches the desired experimental task. The chemical microplates will be housed in the microplate stack or "pantry." There are eight shelves in the stack and any shelf can be accessed by the robot non-sequentially so the order the microplates are inserted into the stack is not relevant. The user will have to use the graphical user interface to designate which chemical plates will be sampled and deposited into what cell microplates. The chemical microplates will be assigned their own list in code, called a *parent*, which will store in memory what cell plates will be treated by the respective plate, called the *children* of the parent chemical microplate. This process will be explained in greater detail in the LCD and Keypad subsection of this Robot Operations section.

The input stacks will be located on the periphery of the workspace of the left side of the workspace when looking at the front of the robot. The input stacks are geometrically rectangular prisms made out of stainless steel sheet metal. The shelves will have sheet metal bases that will divide the whole stack into eight cubbys each holding one microplate each. The stacks are supported by two linear actuators on each side of the stack. These linear actuators allow the input stacks to move up and down in the z-axis. There is a cutout beneath the stack that allows the stack to drop down into the base of the robot.

The workspace of the robot is raised from the base of the robot to allow room for the stack to drop below the workspace. This enables the stack to align any of the shelves with the workspace where the microplates can then be slid out of the shelving unit onto the workspace for processing. Since the top of the shelf needs to be accessed the workspace is raised higher than the height of the input and output stacks. This ensures the stacks can drop far enough for the top shelf to be unloaded.

The input stack has a cutout on the side of each shelf. This shelf will be used to push the microplates out of the stack so that it can begin to slide along the workspace and be processed by the pin transfer tool in the main section of the workspace. A flat plate is located on the back of the shelf unit. The plate has a pin that sticks out toward the front of the robot and the pin slides along a track the length of the cubby. Once a shelf is selected to be unloaded the stack lowers so that the respective shelf is level with the workspace. The pin is then engaged by a small actuator mounted to the workspace. This actuator slides the pin forward pushing the flat plate behind the microplate. This forces the microplate out of the shelf unit and onto the workspace table. The pin will be spring-loaded so that it returns to its starting position once disengaged from the actuator used to unload the microplate. From here the plate will begin to be processed by the pin tool and the sliding pin will be disengaged by the actuator. The input stack will then move to the next desired position determined by the software.

## Workspace Microplate Rail

The main workspace of the pin transfer robot houses three microplate rails. These rails allow for the movement of reservoirs and microplates along the length of the robot's workspace. The three rails have distinct uses. From front to back, the first rail contains live cell culture microplates that are fed from the live cell culture microplate stack. This microplate rail runs the length of the workspace between the input and output live cell culture microplate stacks. The second rail contains chemical factor microplates that are fed from the chemical factor microplate stack. This microplate rail runs the length of the workspace between the input and output chemical factor microplate stacks. The third rail contains pin transfer tool washing reservoirs. There are three reservoirs in total all on the third microplate rail. These reservoirs each contain a different cleaning solution used to remove chemicals and contaminants on the pin transfer tool after each wash cycle to prevent cross contamination between different microplates. The reservoirs are all tethered with vacuum and liquid transfer hoses to allow the automated draining and refilling process to take place between a user-defined number of cycles. This wash reservoir rail runs the length of the workspace. The absence of input and output stacks allows more horizontal space for the wash reservoirs to slide along the workspace. This is important because the pin transfer tool is fixed in the center of the workspace and cannot traverse the length of the workspace so when it interacts with any of the microplates or reservoirs occupying the workspace rails it must be centered in the workspace. The pin transfer tool only has two degrees of freedom: one is to raise vertically up or down, and the other is to move widthwise to select which of the three workspace rails will be operated on.

The rails used in this robot are v-slot rails and are belt driven, powered by NEMA motors. The motors are mounted at the end of the rail beneath the workspace. The belt is fed through the v-rail and is fastened to the microplate carrier that moves the microplates and reservoirs across the workspace. There should be no sag in the belt it should be tightly fastened so that there is no slack in the belt. The NEMA motor is electrically connected to a motor driver that ensures the motor is accurate and places the microplate in the precise desired location.

## Pin Transfer Tool

The pin transfer tool is specially designed to transfer extremely small amounts of liquid in a precise manner. Conventionally the pin transfer tool can be done manually or robotically. For manual operation the pin transfer tool is held and slowly lowered onto a source microplate. The contents of the source microplate will be transferred to the destination microplate. The transfer tool is typically aligned using a key plate that allows for key pins on the manual pin transfer tool to be aligned in order to ensure that each pin on the pin transfer tool enters its respective well. This robot pin transfer tool eliminates the need for a key plate by using very accurate NEMA motors on a linear belt driven actuator platform. The linear actuator is calibrated in a manner that ensures the pin transfer tool is always inserted correctly into the microplate it is operating on.

The pin transfer tool transfers such a small amount of liquid that conventional liquid handling methods such as pipetting do not work accurately. Instead the pin transfer tool transfers liquid by cleverly manipulating liquid to surface adhesion on the pins. The pins are calibrated so that each transfer makes a very precise dilution ratio in the destination microplate. The two main parameters that can be controlled which affect the amount of liquid that is transferred is the depth which the pin tool is dipped into the liquid in the microplate wells and by the speed of the pin transfer tool as it withdraws from the liquid in the microplates and moves to deliver into the destination microplate. Because of this, both of these parameters can be customized using the operator LCD display plus keypad. This will be elaborated on in the *LCD Display and Keypad* subsection.

The pin transfer tool hangs from a gantry that can operate in two degrees of freedom. The first degree of freedom being up and down and the second being depth of the robot so that it can access all three workspace rails. Because it cannot traverse the third degree of freedom the workspace rails must relocate all objects the pin transfer tool desires to operate on to the center of the workspace directly under the pin transfer tool. An example scenario would be a simple pin transfer process from a source chemical plate to a destination cell culture microplate. The input plate stacks would unload both the chemical factor microplate and the cell culture microplate onto their respective workspace linear rails. The workspace linear rails would then independently move the microplates to the center of the workspace. The pin transfer tool will first dip into the source chemical microplate at a user specified depth and speed before raising out of the microplate at a specified speed. The pin transfer tool then translates to the destination cell culture

microplate workspace rail directly above the destination microplate where it then descends into the plate at a user specified depth and speed. After the liquid is transferred into the destination microplate the pin tool is run through the cleaning and drying procedure which is outlined in the *Washing and Drying the Pin Tool* subsection. The carts containing the microplates then slide along the linear rails to deposit the microplates at the output microplate stacks. For further detail on the output microplate stacks see the *Output Microplate Stack* subsection.

## Washing and Drying the Pin Tool

After each pin tool transfer operation the pin tool must be washed and dried so that no contaminants or chemicals cross contaminate the source or destination microplates. In most recommended standard operating procedures (V&P Scientific) up to three washing solutions are used to clean the pin transfer tool. The pin transfer tool is moved from aqueous to organic solvents last. This is because organic solvents have a lower boiling point and will evaporate from the pin tool faster than aqueous solutions. This expedites the pin tool washing and drying process therefore speeding up one cycle of the robot. A cycle is defined as the time required to unload both the chemical factor source plate and the cell culture source plate, transfer the liquid between the microplates, deposit both microplates in their respective output microplate stacks, and for the pin tool to then be washed in all three cleaning solutions and dried. A cycle ends when the pin tool and all of the pin transfer robot's subsystems return to their origin positions.

For reference the standard operating procedures are quoted from V&P scientific on how to properly clean and maintain the pin tool. There are many times that the pin tool should be cleaned including weekly cleanings, before and after starting an experiment, and in between transfers from the source chemical factor microplate to the destination live cell culture microplate. The pin transfer robot only automates the cleaning process in between liquid transfers from the source plate to the destination plate. Because the robot does not automate the pin cleaning or drying before or after experiments the robot operator must ensure the pins are properly maintained outside operation.

The washing process involves three wash reservoirs that contain a cleaning solution as well as larger reserve reservoirs that contain extra wash solution. The larger reserve reservoirs are located behind and above the wash reservoirs. There are transparent gravity fed hoses that connect the reserve reservoir to the wash reservoirs. The hose is stopped by an electronic solenoid that is closed by default. The wash reservoirs are also connected by a second hose to an external vacuum port. The vacuum is provided by the central lab vacuum system and a nozzle should be available in the biosafety cabinet or wherever the robot is being operated. There is another solenoid used on this line that is defaulted to closed. When electronically opened this drains the wash reservoir empty so that new solution can be deposited into the wash reservoir from the reserve reservoir. After emptied, the vacuum solenoid closes and the reserve reservoir opens to replenish the wash reservoir. The amount of time that the solenoid remains open is determined by parameters entered into the robot by the robot operator. The robot operator must also define how many cycles, if any, are performed before the wash reservoir is emptied and

refilled. This is done using the LCD display and keypad which is described in more detail in the *LCD Display and Keypad* subsection.

An example wash setup will be described in this paragraph. **\*\*It is important that the wash step you use is tailored to your experiment. Different chemical properties may require different cleaning solutions, number of cleaning steps, or cleaning solution replacement frequency.** This in between pin transfer cleaning procedure is copied from the V&P cleaning protocol cited in this document. A general cleaning solution setup would have the first reservoir contain DMSO. The second reservoir would contain deionized water (dH<sub>2</sub>O). The third reservoir would contain an alcohol based cleaning solution such as seventy percent ethanol (70% etOH). The pin tool should dip in and out of each of the cleaning solutions 3-4 times and should end with the alcohol based cleaning solution. This is because the alcohol based cleaning solution dries from the pin tool faster than the other cleaning solutions due to its lower evaporation point. There will be a drying fan which consists of a fan in front of a heating element that will also be included on the third workspace linear rail with the cleaning solutions. This drying fan will be used after the three wash steps to dry the pin tool by blowing on it from below. The amount of time required for the drying fan to be used will be **given a default value by the manufacturer (enter time in seconds here after testing is done)** but this time can be overwritten by the robot operator by using the LCD display and keypad. This is described in more detail in the *LCD Display and Keypad* subsection.

## Output Microplate Stack

The pin transfer robot is designed to work with standard form factor microplates. The robot can be modified to work with many different dimensions by contacting the manufacturers of the Pin Transfer Robot, but this particular prototype is designed around the dimension 11.6 mm x 127.75 mm x 85.34 mm (height x length x width). A microplate typically has 6, 12, 24, 48, 96, 384 or 1536 sample wells arranged in a 2:3 rectangular matrix.

The dimensions of the microplate is important to match the dimensions considered when building your pin transfer robot for multiple reasons. It is important that the pin transfer robot knows the exact coordinates of the microplates at any given time in the pin transfer process. This is because if the robot is incorrect in its assumption of the microplate's position the pin transfer head could be damaged by being crushed into the microplate and the microplate and its contents could also be damaged. The dimensions are also important to ensure that the pin transfer tool head properly lowers into the microplate at the correct depth. Incorrect microplate dimensions could lead to the pin transfer tool either bottoming out on the microplate and damaging the pin transfer tool or it could lead to a shallow dip into the liquid contents of the microplate which would transfer an incorrect amount of liquid contents without notifying the operator to the error.

The microplate dimensions being used are also extremely important to ensure that the microplates fit into the input and output microplate stacks. The output microplate stack will be a shelf-like unit that is built to house all of the chemicals and incubated cells residing in microplates. If the microplates are an incompatible size then they may not fit

in the shelf or may fall out of the shelf or be crushed when the shelving unit moves up or down on its linear rail to load or unload a microplate. For this reason only the correct dimension microplate should be used when operating the pin transfer robot to avoid damaging and part of the robot or any of the microplates and to avoid biological or chemical spills.

There will be two output stacks on the pin transfer robots workspace when the robot is configured for a eight microplate max handling capacity. One output microplate stack is to hold up to eight microplates which contain the chemicals to be transferred. These microplates will have any assortment of chemicals all within the wells of the microplates. All of these chemicals will be transferred onto the live cell plates at once with the pin transfer tool so it is necessary that the well layout of these chemical microplates exactly matches the desired experimental task. The chemical microplates will be housed in the microplate stack or “pantry.” There are eight shelves in the stack and any shelf can be accessed by the robot non-sequentially so the order the microplates are inserted into the stack is not relevant. The user will have to use the graphical user interface to designate which chemical plates are to be transferred into what cell microplates. The chemical microplates will be assigned their own list in code, called a *parent*, which will store in memory what cell plates will be treated by the respective plate, called the *children* of the parent chemical microplate. This process will be explained in greater detail in the *LCD and Keypad* subsection of this Robot Operations section.

The output stacks will be located on the periphery of the workspace of the right side of the workspace when looking at the front of the robot. The output stacks are geometrically rectangular prisms made out of stainless steel sheet metal. The shelves will have sheet metal bases that will divide the whole stack into eight cubbys each holding one microplate each. The stacks are supported by two linear actuators on each side of the stack. These linear actuators allow the input stacks to move up and down in the z-axis. There is a cutout beneath the stack that allows the stack to drop down into the base of the robot. The workspace of the robot is raised from the base of the robot to allow room for the stack to drop below the workspace. This enables the stack to align any of the shelves with the workspace where the microplates can then be loaded from the workspace into the designated output shelving unit where the processed plates will be harvested by the operator. Since the top of the shelf needs to be accessed the workspace is raised higher than the height of the input and output stacks. This ensures the stacks can drop far enough for the top shelf to be unloaded.

The output stack has a cutout on the side of each shelf. This shelf will be used to push the microplates out of the stack so that it can begin to slide along the workspace and be processed by the pin transfer tool in the main section of the workspace. **A flat plate is located on the back of the shelf unit.** The plate has a pin that sticks out toward the front of the robot and the pin slides along a track the length of the cubby. Once a shelf is selected to be unloaded the stack lowers so that the respective shelf is level with the workspace. The pin is then engaged by a small actuator mounted to the workspace. This actuator slides the pin forward pushing the flat plate behind the microplate. This forces the microplate into the shelf unit and onto the workspace table. The pin will be spring-

loaded so that it will be disengaged by the actuator and returns to its starting position once disengaged from the actuator used to load the microplate onto the stack. The output stack will then move to the next desired position determined by the software. From here the plate is a part of the stack and may be harvested by the operator after the whole run is complete.

## LCD and Keypad

The primary way the robot operator interacts with the pin transfer robot and that user parameters are defined is through the keypad and LCD display. The keypad contains 0-9 and A-D keys as well as . and # keys. These are used to select options prompted on the LCD display and define user parameters and variables such as pin tool dipping depth, pin tool transfer speed, number of wash steps, and number of cycles. During operation the LCD display will display current progress and will give the user updates as progress is made. An estimated time to completion will be displayed and the current step being performed on the current plate will also display.

After loading the input microplate stack the user will be prompted to identify what plates are in each of the shelves. There will be four default plate names that the user can select from. These can be changed in the code of the robot. To do this see the *software design* section. After assigning names to the plates in each shelf in the plate stack the robot will prompt the user to define pin transfer and wash parameters. The pin transfer parameters include the dipping speed and the dipping depth. Dipping speed can be set by the user in cm/second. Figures 22a and 22b from V&P Scientific show the amount of liquid transferred based on speed of pin tool withdrawal on both Aqueous and DMSO solutions.

a)

## Effect of Increasing Withdrawal Speed on Aqueous Volume Transfer

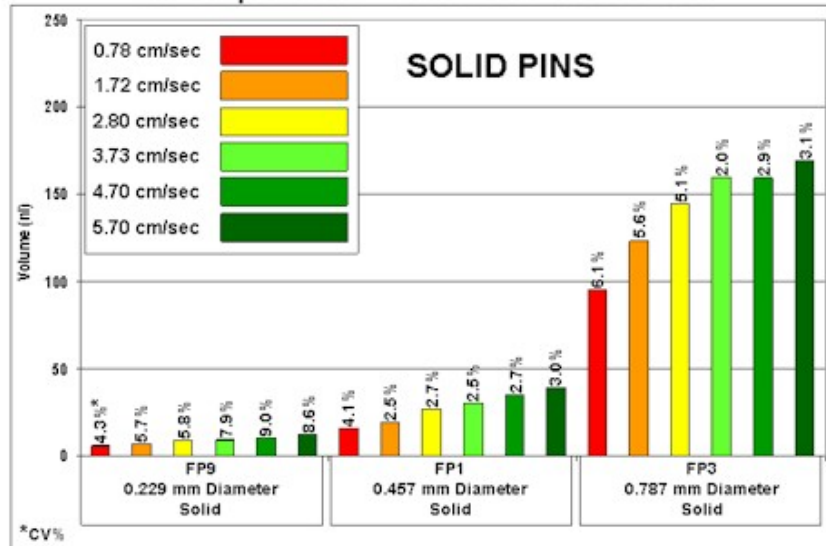


Figure 22a - Volume of liquid transferred based on pin withdrawal speed.

b)

## Effect of Increasing Withdrawal Speed on DMSO Volume Transfer

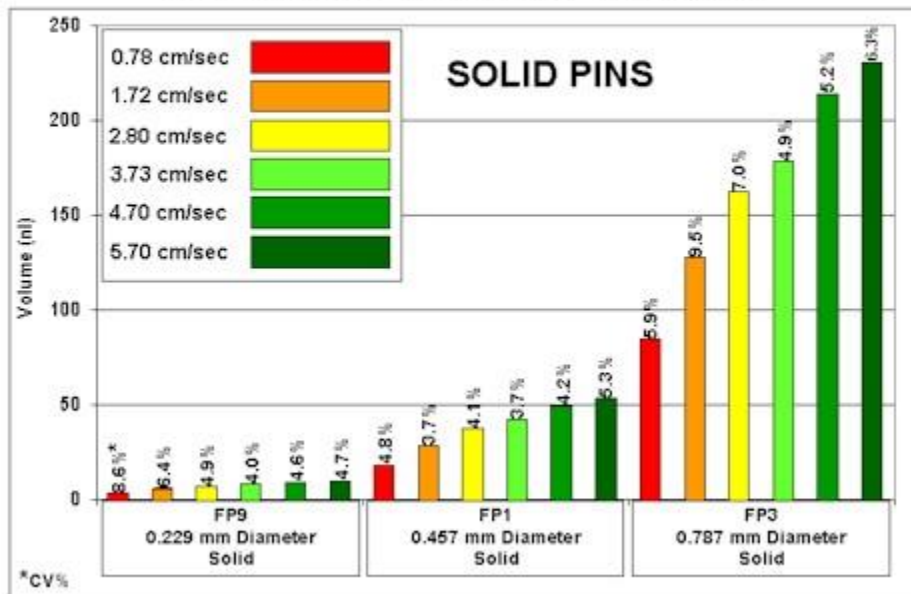


Figure 22b - Volume of liquid transferred based on pin withdrawal speed.

## Safety Features



The pin transfer robot operates on 120 volt ac power from a wall outlet. For information on power consumption see the *electrical diagrams*. In the case of any emergency where the robot is malfunctioning or one of the components or operators may be damaged or hurt there is an emergency shut off button on the main power cord. Press the red “EMERGENCY SHUT OFF” button to immediately turn off the robot in the case of any emergency. It will not damage the robot to do this out of precaution however you will lose any data that has not been saved to memory mid run. This includes barcode scans, microplate locations in the workspace, and user defined parameters

## Cleaning and Maintenance

The pin transfer robot is designed to be fully sanitizable so that it may be used in biosafety cabinets while minimizing the risk of cross contamination. The pin transfer robot should be sanitized before and after every use by spraying the exterior down with seventy percent ethanol (70% etOH) and wiping it down with scientific delicate task wipers.

The pin tool may be dismantled from the microactuator so that it may be cleaned after each experiment and each week as outlined by V&P Scientific. To find more information on standard operating procedures on cleaning your V&P pin transfer tool, see the *appendix*.

The wash reservoirs should be washed after each experiment with Dimethylsulfoxide (DMSO), deionized water (dH<sub>2</sub>O), and seventy percent ethanol (70% etOH). Generously cover the wash reservoir with each of the wash solvents one at a time starting with DMSO. Wash the wash reservoir with a scientific task wipe and then repeat the process with dH<sub>2</sub>O and 70% etOH. If for any reason a chemical or biomaterial spill happens in the workspace the robot should immediately be stopped and protocols should be followed from the laboratory’s MSDS. Clean the robot with a scientific task wipe and cleaning solution such as bleach, 70 percent ethanol, or deionized water.

If a spill occurs within the base of the workspace of the pin transfer robot it may need to be disassembled. You can remove the back panel by unscrewing it with a phillips head screwdriver and then you can wash the inside of the base or if a spill occurs within the microplate input or output stacks. If for any reason a chemical or biomaterial spill happens in the workspace the robot should immediately be stopped and protocols should be followed from the laboratory’s MSDS. Clean the robot with a scientific task wipe and cleaning solution such as bleach, 70 percent ethanol, or deionized water. After the spill has been cleaned, replace the back panel of the workstation base by screwing it in with a phillips head screwdriver.

If a spill occurs in the microplate input stacks it may need to be disassembled. Remove the microplate stacks from the workspace by unscrewing the stack from the two vertical linear actuators it is mounted to. Be very careful as the microplate stack will fall as soon as the screws are removed. You will need to hold the stack as you remove the last screws. Remove the microplate stack from the pin transfer robot. If for any reason a chemical or biomaterial spill happens in the workspace the robot should immediately be stopped and

protocols should be followed from the laboratory's MSDS. Clean the robot with a scientific task wipe and cleaning solution such as bleach, 70 percent ethanol, or deionized water. After the spill has been cleaned, reinstall the microplate stack by aligning it with the two vertical linear actuators and then screwing the phillips screws into the tapped screw holes in the linear actuator.

**DO NOT POWER ON THE ROBOT AT ANY TIME WHILE ANY OF THE PARTS ARE DISASSEMBLED OR HANDS OR BODY PARTS ARE WITHIN THE WORKSPACE.**

For any maintenance issues that are not addressed within this operators manual, please contact Christopher Clifford at [chris.e.clifford@knights.ucf.edu](mailto:chris.e.clifford@knights.ucf.edu) with the subject line "Pin Transfer Robot Maintenance Request." Describe the issue you are experiencing along with any errors produced on the LCD screen.

# Overview of Design and Technical Architecture

There are many aspects of the actual design of the robot, all of which have been conveniently modularized with the express purpose of making each part of the design as flexible as possible. For example, we considered the prospect of using more than one arduino in certain cases where we might not have enough power to power on the gantry robot. Therefore, this design overview should be taken as a guideline. Depending on the usages, there may, in fact, be more than one way of implementing it that might satisfy one or more of the design goals in a different way than to be described.

The project's design starts first and foremost with the pin tool, the scientific instrument responsible for carrying the liquid from the chemical well plates and inserts them onto the wells in the cell plates. The pin tool is the most important part of the project since without it, much of the design around it simply wouldn't make sense. The pin tool we plan on using should be capable of handling 96 Perkin Elmer well plates, but our intention is to make the attachment of a pin tool generic in case there was a researcher who wanted to use a higher quality pin tool that might be able to perform pin transfer operations with 384 wells, 1536 well plates, or even more than that. Higher quality pin tools can also perform more pin transfer operations with a reduced risk of cross-contamination. The pin tool can also be selected to be polarized, since that would increase the surface tension used to carry the liquid around in the pin tool.

In a CNC gantry machine, a piece of technology is attached as a centerpiece at the intersection of the axes. Each axis acts as a degree of freedom for the centerpiece. In other words, a 2D CNC gantry machine would be able to move through two axes freely.

In our design, we also considered the number of axes that the gantry machine would need in order to perform a successful pin transfer process. Generally speaking, if your only goal is to perform a single pin transfer operation, then you only need a z-axis CNC for that. In which case, it would only need to go up and down to perform the pin transfer operation. In practice, however, it is not so simple. The pin tool needs to undergo some washing steps as well, which means that it needs one more degree of freedom to move horizontally to a place where it gets dipped into a cleaning solution. A third axis was also considered. If the CNC gantry machine were to have three degrees of freedom, then it would eliminate the need for a conveyor rail to move them into position: you would simply need to put the well plates in specific positions on the biosafety cabinet and have the gantry service them one by one. However, we determined that it would make more sense to have two degrees of freedom as we will need to move the well plates into an output stack later anyway. Having a third degree of freedom would likely not help in that case and so it would be wasteful to implement it.

To that end, it made more sense to use two conveyor rails that take in a set of plates to move for the pin transfer operation to take place and then put them into a storage space where they can be properly picked up later.

Lastly, there is the storage method. The design of the storage method is very important as it needs to meet the constraints of height, width, and depth of the biosafety cabinet in order to be properly implementable. The design of the storage method is also important as its implementation can be very expensive, so it is unlikely that there will be too many iterations run on it. We've determined that using a stack would be the most appropriate storage mechanism for the well plates as all you would need to do is determine a way of removing well plates from the stack and moving them onto the conveyor rail as well as inserting them into the stack once the processing is done.

## Design Constraints

Design constraints are the guiding hand of all engineering projects as they determine what is and is not possible. The design constraint topics cover every possible aspect of a project that a designer must consider, from user health and safety to social and political implications to the cost of manufacturing such a project. The following sections will examine how each of the different design constraints affects the Pin Transfer Tool as this is the most important part of the entire project. For our purposes, the Pin Transfer Tool is treated as a scientific instrument and therefore needs to be carefully planned out for the rest of the project to work.

## Economic and Time Constraints

Economic constraints limit the quality and quantity of the parts available for use as well as any tools needed to create the project. Outside funding has yet to be secured up until this point, so the Pin Transfer Tool is completely student funded. Upon closely inspecting the costs for all of the parts listed earlier, you will find that the majority of the cost is based on the Pin Transfer Tool, the Power Supply Unit(PSU), the DC-DC Converter, the workspace rail, and the input and output stacking mechanisms. Outside of the workspace rail and the I/O stacking mechanisms whose cost has yet to be quantitatively measured, the Pin Transfer tool is arguably the most expensive tool in the project. Each team member is willing to put out \$250 for the production of the project, putting the total project budget at \$1000. While this is a self-funded project and a goal of ours is to keep production costs down as much as possible, we have made attempts and continue to plan on acquiring a sponsor. The reason for this is while it is possible to manage the costs of attaining some subset of the parts we have now, it is likely that the I/O stacking mechanisms and the workspace rail will go over our budget a fair bit. Either way, lower quality parts will likely need to be used whenever possible to keep production costs down. We still intend on maintaining a quality pin transfer operation in the process of that, but we take liberty in cutting costs wherever they do not meet with the design specifications of our project.

Time constraints will cause this project to be a fully fleshed out design, but necessarily industry grade. Four months (beginning of January 2021 – end of April 2021) are allotted for the design phase of this project. In this time, a complete design for the project must be created and all of the required parts for the project must be known. The short duration

of the design phase limits the available time to look into other, possibly better, project designs. Another four months (beginning of August 2021 – end of November 2021) are allotted for the assembly of the project. This limits the time for prototyping and the ability to make any major design changes as that could lead to the project not being fully assembled before the final deadline.

## Manufacturing and Sustainability Constraints

Manufacturing constraints limit the ability to use any kind of novel system or piece that is not easily reproducible. Therefore, most of the physical parts of the project will be existing products already on the market. There will be two custom made parts of the project. One part that is being custom made is the PCB. While it will be custom designed, the physical piece will be purchased and assembled through an established company to increase its manufacturability. The other part that is being custom made is the Pin Transfer Tool base. Similar to the PCB, this part will also be purchased and created through an established company.

Manufacturing constraints also limit the size of the Pin Transfer Tool. The Pin Transfer Tool will be in an enclosed area that is five feet tall and three feet wide, so it needs to be smaller than those dimensions. The goal is to have the Pin Transfer Tool be no greater than four feet tall and two feet wide. This is to give the user enough room to interact with the Pin Transfer Tool without having to remove it from its casing.

The manufacturing constraints also limit the quality of the stacking mechanism and workspace rail. Our plan is to use a pair of stacks, one for the animal cells and another for the chemicals. A well plate from each stack is to be carefully dropped onto their respective workspace rails for the pin transfer operation before they are stored in an output stack after the pin transfer operation is complete. Our first goal is to have an input stack to be able to store eight well plates at a time and very carefully and safely drop them onto the workspace rail for processing. Our second goal is to have a pair of output stacks that are each capable of taking in up to 8 well plates after their processing is done.

Sustainability constraints come from the chemicals that are being worked with. The chemical that the user inserts into the Pin Transfer Tool could be very acidic or basic. Chemicals on far ends of the pH scale tend to cause damage to substances that they come into contact with. It is possible that the user will accidentally spill a chemical on to the Pin Transfer Tool. Therefore, the materials used in the Pin Transfer Tool should be robust against damage from most common chemicals as well as many chemicals that fall on the far ends of the pH scale. In addition, the wash steps will be made generic so as to facilitate the process by which the pin tool is cleaned to tailor to any individual needs and preferences.

## Societal and Political Constraints

The societal and political constraints for this project relate to the user's cost. Large scale Pin Transfer Tools already exist, but can cost thousands to tens of thousands of dollars. Usually, labs that have the large Pin Transfer Tools are dedicated to just using those machines and receive packages from other labs to run through the large Pin Transfer Tools. The goal of this project is to create a Pin Transfer Tool that is affordable to smaller labs that do not have the room space or money for the standard cost and sized ones. This Pin Transfer Tool being designed should cost the user between \$250 - \$750 to buy. Note that the sale price of the Pin Transfer Tool is directly proportional to the manufacturing costs (and this is typically the case with any manufactured product), which is why we do our best to ensure that any money spent on this project is money well spent.

There should not be any political constraints to this project. People of every race, color, ethnicity, and nationality, religion, and political affiliation should be able to benefit equally from this project and/or its results.

## Health and Safety Constraints

Health and safety constraints fall on how to protect the user from the chemicals that are being used and from the electrical parts that are part of the Pin Transfer Tool. Symbols will be etched into the front of the Pin Transfer Tool depicting proper lab safety procedures. Also, when the Pin Transfer Tool is turned on, a message will briefly appear on the user interface that reminds the user to follow lab safety procedures. All electrical components will be properly insulated and covered up. The user should not be able to interact with any electrical components.

## Ethical and Environmental Constraints

Ethical and environmental constraints relate to how the user disposes of chemicals used in the process of using the Pin Transfer Tool. Any chemical waste produced by the mixing of chemicals in the Pin Transfer Tool cannot just be thrown away or washed down the sink as such chemicals could pose a threat to humans and animals as well as harm building utilities, such as water pipes. Chemical waste disposal is regulated by the Environmental Protection Agency (EPA) via the Resource Conservation and Recovery Act, so a manual on how to store and dispose of chemical waste following the EPA's guidelines will be created and included with each Pin Transfer Tool.

If the need arises for it, we would be prepared to write a manual on how to use the tool if people find it complex or difficult to use in any way. It is imperative that anyone operating this device be knowledgeable enough with the pin transfer process to know the potential dangers from being in contact with any chemicals in the lab. Equally important is maintaining the purity of the result of the pin transferring process. The reason why this project is important is because typical pin transferring operations that are done manually run into many potential risks for cross-contamination. The goal is to offer a way of streamlining the pin transfer process that is affordable for smaller labs while at the same

time maintaining the quality of the pin transfer operation in a way that most smaller labs deem acceptable.

## Facilities and Equipment

For this project, we will use a 3D printer to print structural components as well as mounting devices. We will also be soldering discrete circuit components onto a PCB in the prototyping stage of this project, thus a soldering station with well-suited ventilation will be necessary. In addition to this, test bench equipment such as oscilloscopes and power supplies will be useful when testing parts that are purchased as well as during the testing and building cycle of the prototype stage. Accounting for these necessities, we will require a location that will allow the team to have these tools/services available to us at a location that is near us.

One location of interest is the Texas Instruments Innovation Lab. This lab is in the Engineering Atrium on the UCF main campus. According to the website for the TI Innovation Lab, the lab has a multitude of tools for designing and testing circuitry such as oscilloscopes, function generators and digital multimeters. Additionally, the lab features soldering stations.

A portion of the device will be constructed of 3D printed materials such as some of the structural components of the device as well as a component to mount the pin tool onto a linear actuator. Therefore, we will need access to a 3D printer as well as filament to print with. The TI Innovation Lab has a 3D printer that has a print envelope of 12x12x10. This should be more than enough for the components that we are using. It is worth noting that the TI Innovation Lab allows students to print for free, however, this attracts other students to want access to the printer as well. In many cases, a queue may be established for students to have access to the printer and thus we may be delayed.

Another location that will be used is the Senior Design Lab located in room 456 of the Engineering building. This lab features most of the equipment that is available in the TI Innovation Lab but without a 3D printer. This is an attractive location since we will have access to more equipment in case of the Innovation lab reaching maximum occupancy.

Access to a 3D printer is important for this project since it is likely that multiple design iterations will be done to reach a suitable and functional design for the components that are needed. In the case of the TI Innovation lab, there is a high chance that queues will form a limit the throughput of our prints. So, it is important that we have access to other locations with 3D printers. There are a few other locations on UCF campus that provide students with access to printers. The Business Administration building (BA1) has free 3D printing. In addition, the Curriculum Materials Center in the Education building also has a 3D printer with a fee of \$0.15 per gram of filament used. Lastly, some colleagues of team members have personal 3D printers. Access to an appropriate 3D printer is necessary for this project and because of the many options we have, we will be adequate in terms of having access to 3D printers.

For the design of the components that will be 3D printed, we will need to use a CAD software. There are many free online CAD software tools such as Tinkercad that should be suitable for the design of the components, however, we can also use CAD software on the computers in the Senior Design lab.

Out of all the locations listed above, the TI Innovation lab and the Senior Design lab will house the main location of the construction of our project. Not only do these facilities contain all the equipment that is required to build our prototype but are both easy for all team members to travel to.

## Personnel

Chris is the team's electrical engineer and is the expert on the medical side of the project. Chris developed the idea to create a scaled down version of high-throughput chemical screening robots during his internship at a small lab. He realized that the remedial manual work of transferring chemicals by hand takes time away from performing more important or useful work in his lab. In the industry, large expensive chemical screening robots are used to remove this work by automating the entire process, but small labs cannot afford this luxury. Chris will aid in designing the circuit schematic for the project.

Brenden is one of the team's computer engineers. He will primarily be focused on designing the circuit schematic as well as programming the microcontroller. Brenden has experience with working on Arduino devices such as the Atmega2560, Espressif microcontrollers and others. In addition to that, he has a strong background in the C/C++ and Python programming languages. This will be useful for writing the code for the microcontroller as well as conceptual knowledge of the system which will be useful for debugging.

Dominic is another one of the team's computer engineers. Dominic has a solid skill set of programming that will be primarily used for developing the user interface for the robot. He has a great deal of experience with Python and a notable grasp of the C language. This will be useful for developing some of the UI components such as the graphical OLED display logic as well as microcontroller programming. In addition to this, he has recently begun research in adversarial attacks. This skill will be beneficial in the testing and debugging portion of the software. Understanding the weak points of the code will allow us to eliminate software bugs to ensure reliability. Lastly, Dominic has a great deal of experience with analysis of software and hardware documentation as well as having a thorough experience with evaluating and applying standards and other government regulations to commercially used products. His knowledge both the documentation and legalities of standards will be useful when developing the project in general.

Yousef is a computer engineer with a firm grasp in embedded programming. He is proficient with low-level programming languages such as C and C++ which is imperative for the success of the microcontroller code and the overall logic of the robot. Yousef also has a great deal of experience with software source control. His talents with source control will ensure that incremental commits are maintained and executed properly. Yousef is



experienced with PCB design software Auto Desk Eagle which will be important for designing the circuit creating the PCB.

The structure of a team is an important consideration since it can be used to outline some of the strengths and weaknesses of the team. In the case of our team, we have 3 computer engineers and 1 electrical engineer. By solely observing this ratio, it can be inferred that we will have a weaker electrical background and a stronger embedded systems and programming background. However, this is not necessarily the case. By looking at each team member closely, the computer engineers bring a great deal of electrical theory background to the table. Similarly, Chris, the electrical engineer on the team, brings valuable first-hand knowledge of the field that the project is used in. Despite a seemingly unbalanced team structure, we have a favorable amount of knowledge of circuit theory and embedded programming that will allow us to excel in the microcontroller programming and circuit design aspects in addition to knowledge on the field that this robot will be used in.

On the other hand, there are areas that we are lacking. This project will require some innovative component designs for some features of the project. Having a mechanical engineer on the team to provide knowledge on spring design and latching mechanisms or someone who has experience with designing 3D parts to have printed would be useful. This is one drawback of our team structure.

Another drawback of our general team composition is the lack of CAD experience. Though we understand the importance of using a 3D printer to create a sizable portion of the components that will be used to build the Liquid Handling Robot, we lack any real experience with any sort of 3D printing software like AutoCAD or SolidWorks. Needless to say, Yousef did dabble somewhat with SolidWorks designs as a freshman. Despite it obviously not being enough for the needs of this project, this is something we plan on working on as we move forward along with the project. If anything, we are coming more and more into contact with similar project ideas used for inspiration that have been properly outlined using some form of CAD software like AutoCAD or Solidworks, and so we recognize the need for this.

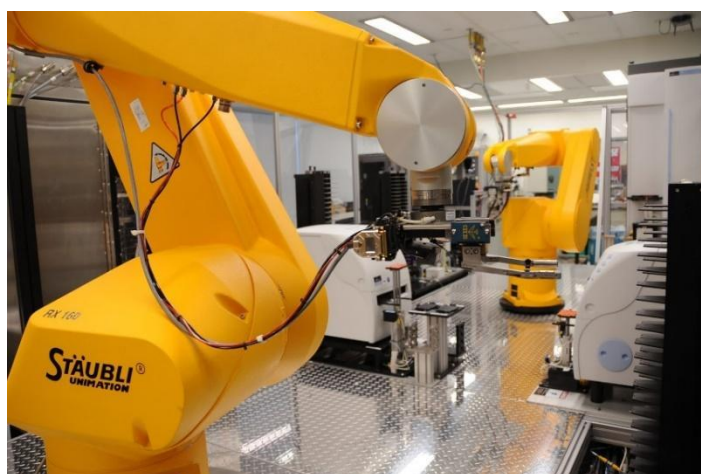
In general, it is important to have a team that specializes in different areas of the problem that is to be tackled. In our case, we have a solid understanding of the embedded system programming, circuit theory and circuit design, as well as expertise in the field that will ensure our success. Despite some limitations in 3D design and other mechanical components, we will be able to develop a suitable prototype to showcase our design.

## Research and Investigations

### Existing Similar Projects and Products

#### High-throughput Chemical Screening Robot

Figure 23 below depicts one example of an automated pin transfer robot in a high-grade laboratory. This machine uses a 3-axis gantry to transfer a small volume of chemicals to an array of live cells. The transfer of chemicals is facilitated through the use of a detachable pin transfer tool. Machines such as this have the ability to mount different size pin transfer tools to allow for a specific number of wells to be treated per cycle.



*Figure 23 – High-throughput Chemical Screening Robot (Public domain)*

#### Epson Compound Transfer Robot

Figure 24 below shows another example of a chemical screening robot found at the ICCB-Longwood Screening Facility at Harvard. This machine mounts a pin transfer tool to its arm which operates with 2 axes. Then, a 3-axis robotic arm can store the treated well plates in a carousel storage system to process at a later time.



*Figure 24 - Epson Compound Transfer Robot (reprinted with permission from ICCB-Longwood Screening Facility at Harvard Medical School, Boston MA)*

## Seiko Compound Transfer Robot

Another example can be seen in Figure 25 below. This is another example of a chemical transfer robot also used at the ICCB-Longwood Screening Facility at Harvard Medical School. Similar to the Epson robot, this machine has a 2-axis robot that transfers chemicals to well plates with live cells and another robot that can move the treated plates to another location to be stored in.



*Figure 25 - Seiko Compound Transfer Robot (reprinted with permission from ICCB-Longwood Screening Facility at Harvard Medical School, Boston MA)*

As seen in these examples, there usually exists one robot for treating the cell plates and another robot for moving cell plates to another location for storage. This is an efficient and effective model of high-throughput chemical screening. In facilities such as ICCB-Longwood at Harvard Medical, throughput is of utmost importance. Each robot used specializes in one task. The 3-axis robot excels in moving treated plates to a secondary longer-term storage location whereas the 2-axis robot specializes in transferring chemicals.

## Project Part Selection

### Pin Transfer Tool

Pin transfer tools are used to transfer very small amounts of liquid from one reservoir to another. It is one of many ways to perform a dilution. What makes a pin transfer tool unique is that it transfers a very small amount of liquid. Conventional pipettes used in laboratories struggle to transfer a few microliters. The pin transfer tool produced by V&P Scientific can transfer quantities as low as 50 nanoliters. The pin tool accomplishes this by using surface adhesion principles. The pin tools are specifically designed so that the amount of surface area the pin has correlates precisely to a transfer volume. A transfer volume of 200 nanoliters into a 200 microliter well plate would give a 1:1000 dilution. These large dilutions are hard to accomplish without a pin transfer tool.

The pin transfer tool can be customized in many ways. The pins can be made from different materials, or coated with different materials. These coatings give the pins certain chemical adhesion properties that can make them absorb more or less of a particular chemical. The coatings generally are either polar or organic attracting one or the other polarities. Polar coatings will transfer much more of a polar molecule such as water. Organic coatings will transfer much more of organic molecules such as DMSO. Surface area can be added to pin tools by using larger diameter pins or creating slits in the pins. The pin tool has one pin for each well of the microplate that it will be dipped into. It is possible to use a 96 pin tool to transfer a 384 well microplate however if it is dipped four times into the same microplate. Figure 26 shown below depicts different pin types as well as slot sizes.

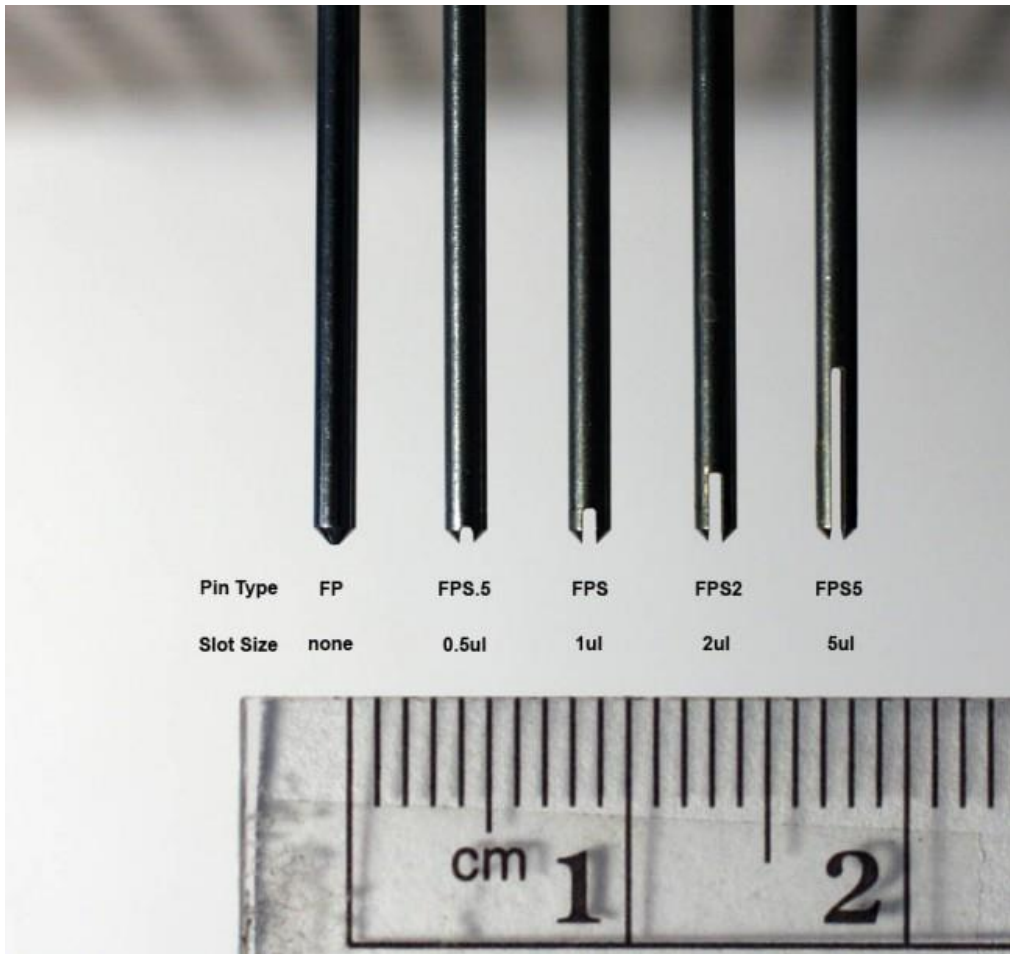


Figure 26. An example of pins that can be selected. *E-Clip Style Floating, Solid Pin - 1.58 mm chamfered to 0.38 mm diameter flat tip, 34 mm long, 23 mm exposed pin length, delivers ~100nl.* Courtesy V&P Scientific.

Typically when purchasing a pin transfer tool the buyer must determine the type of pin (size, diameter, number, coating, slits), as well as the housing for it and, if robotic, and the mounting plate needed to mount the robot pin tool to the liquid handling station. Based on the number of pins desired a fixture is chosen. In Figure 27 below, an example fixture is shown. This fixture is intended for 96 pin layout.



Figure 27. An example 96 pin fixture. *Basic Robot Pin Tool Fixture for 96 FP12 Pins, single float plate, 2.36mm diameter pins, fixture plus spacers.* Courtesy V&P Scientific.

## Manual Transfer Tool

One type of pin transfer tool is the manual form. The manual form is hand operated by a lab technician. The pin tool comes with an aligning plate that has a keyhole. The manual pin transfer tool also has a key pin that slides into the key plate when doing a replication. This key pin ensures that all of the pins of the transfer tool correctly align with the wells of the microplate that is being dipped into. The operator must be very cautious when working with a manual pin tool. This is for multiple reasons.

The first reason is that if multiple transfers are being done they should all be done as similarly as possible to ensure that the experiment does not contain large systematic

error. Small differences such as how fast the pin tool is removed from the microplate or how deep the lab technician dips the pin tool into the microplate will affect the volume of liquid transferred by the pin transfer tool. This is essentially an impossible task for a human. There will be variations in the technique used on different microplates within an experiment. When using a manual pin transfer tool it is typically done for small experiments where a certain level of error is anticipated and permitted.

A second reason is that pin transfers are often done to live cell cultures. Cells are usually adhered to an extracellular matrix in a two dimensional culture at the bottom of the microplate while submerged in cell media. If the pins come in contact with the bottom surface of the microplates the cells can easily be scraped off and will not only die but tarnish the integrity of the experiment because it will affect the cell colony and certain observed phenotypes may not be attributed to the chemical factor used to treat the cell but instead the pin tool which dislodged a portion of the culture. Shown below in Figure 28 is an example of a manual application with a pin transfer tool.

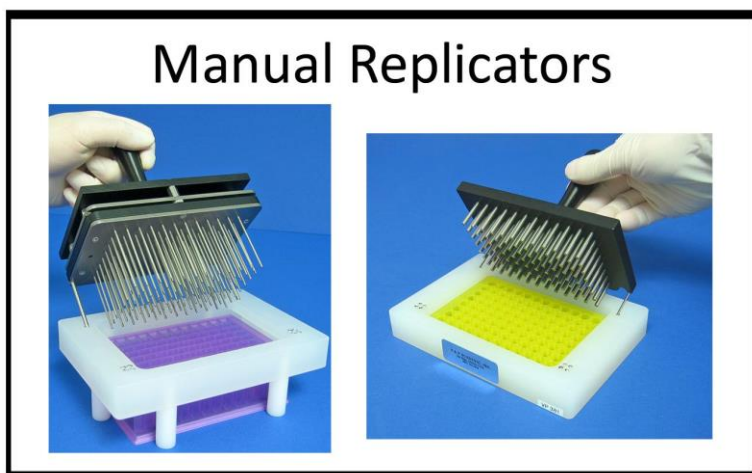


Figure 28. Manual pin transfer tool with key plate to ensure proper insertion. Courtesy of V&P Scientific

## Robotic Transfer Tool

The robotic pin transfer tool is very similar to the manual pin transfer tool in its components. The main difference is that there is no guide pin on the robotic pin tools and the pin tools must be connected to the robot using a custom mounting plate. This is because the pin tool is moved by the robotic system with much more accuracy and a key is not necessary. The robotic system should always be able to ensure the pins match their respective wells when dipped. This is done using accurate motors in concert with motor drivers and encoders.

The robotic pin tool is purchased in separate pieces because V&P sells them to fit as adaptors to commercially available liquid handling robots. These robots would conventionally have pipettes attached to the moving head but in its place is the robotic pin tool. The robotic pin tool is fit to the moving head firstly by the mounting plate. An

example of a mounting plate can be seen in Figure 29 below. These mounting plates are designed to fit a specific liquid handling robot brand and model. It takes whatever mounting mechanism is currently on the liquid handling robot used to attach the pipette tool and makes the new mounting plate a standard format. This standard format allows for a pin fixture to be attached to the adapted liquid handling station.



Figure 29. This is a mounting plate intended to adapt the Beckman brand liquid handling station. Courtesy of V&P Scientific

## Chemical Library

The chemical library is an organization of microplates that contain a variety of small molecules, chemicals, and growth factors. These chemicals are dissolved into solution either, in many cases DMSO, or an alcohol, and then stored in the microplate. The microplates are frozen to maintain the integrity of the chemical as many of them need to be frozen. The layout of the chemical library microplates must match the desired experimental design because it will be copied onto the cell cultures identically if using a pin transfer tool. The chemical library plates contain a relatively high molarity (think 20-.2mM). Once the pin transfer is completed the concentration is greatly reduced in the cell culture microplates.

It is very common that the chemical library will contain many copies of the same chemicals (duplicates) and at different concentrations. Having many concentrations of the same chemical microplates allows scientists to look for dose dependent responses when conducting a chemical screen. Duplicates are valuable because it saves time to make many plates when creating a chemical library. This way when a chemical runs out a duplicate library plate can be grabbed. In large laboratories whose focus is high throughput screening it is also common for prebuilt chemical libraries to be purchased



with common factors that are known to have certain biological pathway influences in the investigator's area of interest such as cancer drugs or liver drugs.

All chemical libraries have an associated database where plates can be identified either by a label or barcode and the spreadsheet can identify which well contains which chemical. This is crucial so that the researchers know what chemicals they have in stock, as well as for result validation so that the researchers can identify what compounds had a positive or negative effect of the cells which were treated in an experiment.

If the reader is interested in learning more about the development of a chemical library for a laboratory see the section *Personal Bibliography* where I detail a summer internship where I construct my own chemical library full of small molecules and growth factors that are relevant to pancreatic development and type 1 diabetes.

## Stacking Concepts

We had to go through several iterations and designs on the stacking concepts. It was a very important part of the development process as it is the most expensive part of the project. We would like to reduce the number of design iterations we perform on the stack when we actually go about implementing the project so that the budget can be better spent on other parts.

The first idea we had considered was a stack with flaps as platforms for the microplates that can act as a hatch to drop a plate onto the conveyor rail. This is expressed in the following diagram below in Figure 30:

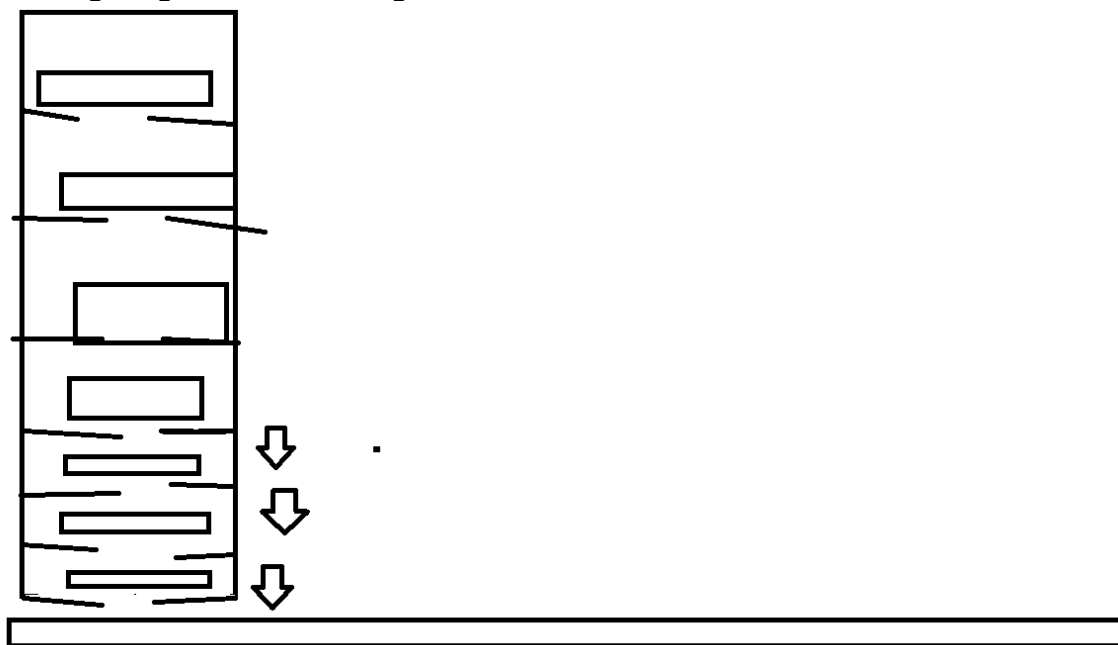


Figure 30 - Multi Hatch stacking approach

With the design shown in this figure, to drop one plate onto the conveyor rail, you need to open the hatch that holds the well plate at the bottom of the stack. After some fixed amount of time, the algorithm assumes that the well plate has been successfully dropped onto the conveyor rail and shuts. Now the hatch at the bottom of the stack is ready to hold the well plate one level above it once its respective hatch opens. This cascades until every well plate makes it onto the conveyor rail.

The main advantage of this approach is that it's quite an intuitive concept and is likely going to work without issues if properly implemented. Another advantage that this approach offers is that it is very symmetrical. If you can properly implement the stacking mechanism for the input stacks, then you can implement the exact same approach for the output stack, with a minor difference shown in the following diagram in Figure 31:

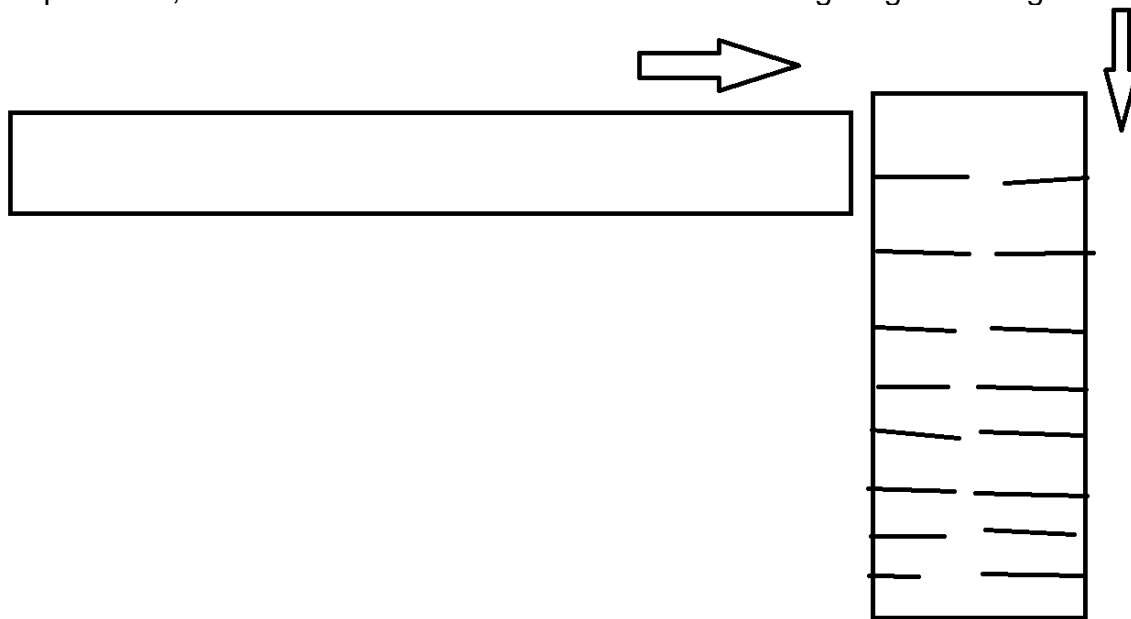


Figure 31: Multi Hatch output stacking approach

As you can see, the approach is virtually identical. The only difference is that in order to properly implement the output stacking approach, you need to align the top hatch with the end of the conveyor rail so that the transition may occur smoothly.

The main disadvantage of this approach is that it can be fairly costly. We plan on having 8 well plates in each of the stacks and we plan on having two input stacks and two output stacks. Whatever it might be, the cost of implementing the hatch mechanism would have to be multiplied by 32. If a pair of linear actuators is used here, then implementing the stacking mechanism would be much too costly for what would be acceptable for this project's budgeting constraints. Another issue with this approach is that there is a chance that the height of the stack might exceed that the clearance allowed for it to enter the biosafety cabinet. To add insult to injury, if you wanted to insert or retrieve the well plates from the stack, you would have to have a way of manually dropping each of the plates, one by one, for inserting or retrieving the well plates, which is very awkward and mechanically inefficient. Though this is not necessarily the most robust approach to

stacking, it was certainly a good starting point for other stacking mechanisms that would follow.

The next approach comes from the observation that you can take advantage of the fact that you can move each platform holding the plates to the conveyor belts, one by one. The figure 32 below expresses this in better detail:

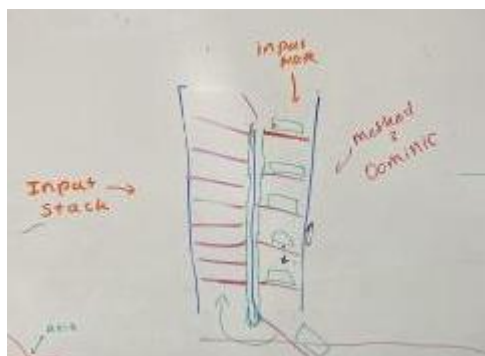


Figure 32 - Cyclical queue stacking approach

In this approach, you can see the well plates are arranged in a cyclical queue. Each well plate is moved to the bottom via circular conveying action, each dropping the well plate they contain before being moved to the back of the queue.

The main advantage of this approach is that it is very mechanically efficient: the only piece of equipment you need to spend your budget on is the conveying action, which is going to consist of a handful of motors at most. Another advantage that this approach offers is ease of insertion and retrieval. You can make a “cabinet” of sorts that can be used to open and close the insides of the stacking mechanism. That way, if you want to insert the well plates into the stack, all you need to do is open the cabinet and insert them, one by one.

The main disadvantage of this approach is that it takes up a lot of height to implement. There is a fair chance that an implementation of this approach might have a height that exceeds that of the biosafety cabinet. This is a fairly reasonable approach to use, but there is a way to reduce the risk of exceeding the height constraints of the biosafety cabinet.

The third approach simply involves extending the conveyor rail’s length to being able to hold the eight well plates at a time. This carries with it some nice advantages. If all that is needed is to extend the conveyor rail, then there is no concern for the height. There may, however, be a chance that the surface area might be exceeded, which is likely its biggest disadvantage. Furthermore, it is likely the simplest approach to implement, since it avoids much of the nuance that has to do with implementing a stack mechanism to move well plates onto a conveyor rail.

If there is surface area to fit it, it would be a fairly legitimate design concept to simply extend the conveyor rail to fit perhaps either the input or the output if it cannot fit both.

This can massively simplify the design and may well make it cheaper as you would not have to buy any extra equipment for any separate conveyor rails.

The last approach that was considered was similar to the Circular queue stacking approach, except the circular queue takes the shape of a ferris wheel. The ferris wheel stacking approach is shown in the figure 33 below:

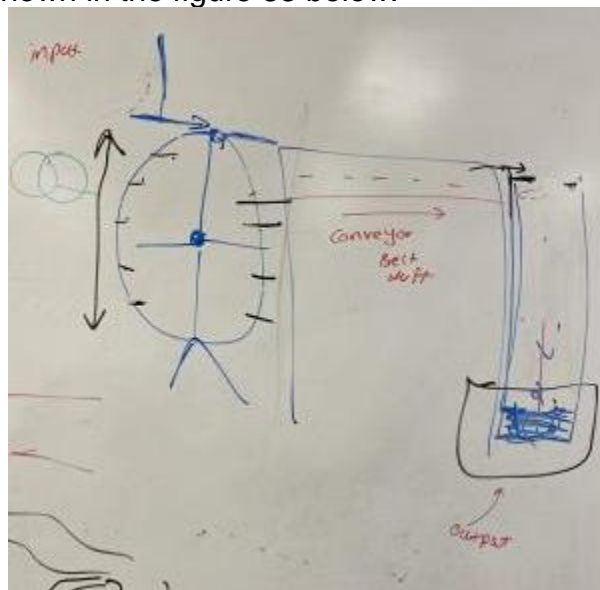


Figure 33 - Ferris wheel cyclical queue stacking mechanism

The main advantage of this approach is that it takes less height than a simple circular queue. This will make it easier to fit into a biosafety cabinet. That's just about the only advantage to this approach.

The main disadvantage of this approach is the difficulty of implementation. You would have to have a way of lining up the top of the ferris wheel with the conveyor rail. It is also worth noting that this approach isn't symmetrical either, meaning that it cannot be applied to both the input and output sections of the Liquid Handling Robot.

To conclude, there are many possible iterations of the stacking mechanism that are viable and may well be used in the final build. It is worth noting that these approaches are not final: it may well be the case that there can be more of these iterations to come. As it stands, it is probably most efficient to use the Ferris wheel on the input and perhaps either an extended conveyor rail for the output or a simple cyclical queueing mechanism for the output.

## Microcontroller

Microcontrollers are small computers that can execute instructions. Most commercial microcontrollers such as Arduino and Texas Instruments' MSPxx boards, contain I/O pins that can be used to connect sensors and other types of components to be controlled by the microcontroller by user-programmable code. Some microcontrollers even come with

Wi-Fi and Bluetooth capability which allows these devices to connect to the cloud. These devices belong to a category of microcontrollers that are in the domain of Internet of Things (IoT).

Microcontrollers consist of the IC chip itself which houses the processor and other silicon components. Some common features of microcontrollers include:

- Analog-to-digital converter (ADC)
- Digital-to-analog converter (DAC)
- Timers
- PWM
- Internal pull-up resistors
- I2C
- SPI
- UART
- SRAM
- EEPROM
- Flash Memory

Due to the advent of companies like Arduino, there are millions of people around the world who use microcontrollers to automate tasks, monitor environmental conditions, and for a plethora of other applications. Because of this, there are many microcontrollers on the market with varying properties and technical specifications. This is not only great since the market competition keeps the price relatively low, but also allows for an individual to choose a certain microcontroller that fits their needs. In addition to this, some companies, such as Arduino, open source all their schematics and software for the boards, allowing for individuals to construct their own boards to meet their project's needs.

For our project, the microcontroller will operate all functionality of our robot including controlling motors for linear actuators, workspace rail movement, controlling servos, and managing the user interface with a keypad and graphical display.

It is imperative that we select the right microcontroller for our design to remain simple. The key point in using an embedded system is to have a centralized piece of hardware used to interface all the moving mechanical parts and LED/OLEDs to the project. We will now compare some of the boards of interest to determine which board best suits the needs of the project. However, first we will list some of the necessary features that we will need for this robot:

- A lot of GPIO pins to interface with all components.
- Interrupt capability
- ADC
- DAC
- PWM for controlling motors.
- Decent memory to store the code.

Given these needs for our desired microcontroller, we will compare some of the popular microcontrollers available for commercial use.

## ATmega328P (Arduino UNO)

One of the most common microcontroller boards is the Arduino UNO based on the ATmega328P microcontroller. The ATmega328P is a high performance 8-bit microcontroller that is capable of up to 16MHz clock speed. It also has a variety of useful peripheral features such as:

- Two 8-bit counters
- 1 16-bit counter
- 6 PWM channels
- 8-channel 10-bit ADC
- SPI, UART, I2C interfaces
- Interrupts
- 6 Low Power modes

This microcontroller supports roughly 14 digital I/O pins and 6 analog I/O pins. Of those 14 digital pins, 6 of them can be used for pulse width modulation (PWM). This is an attractive board since it can be programmed with Arduino's efficient and easy to use code and software. This board is relatively inexpensive with a cost of roughly \$20. This board meets all but one of the requirements for our project in that there are not enough GPIO pins. To ameliorate this, it is possible to buy more than one Arduino Uno, though it can complicate the design of the project, especially if you need to interface both Arduino's at once in order to effectively make use of the digital and analog pins of both boards **[ATMEGA-328P]**.

## ATmega2560 (Arduino MEGA 2560)

Another option is the ATmega2560 microcontroller that is found on the Arduino MEGA 2560 board. This board supports all the capabilities of the ATmega328P but with a total of 54 digital I/O pins, 15 of which can be used for PWM. In addition to this, there are 16 analog I/O pins and 4 UARTs. This board makes up for the shortcomings of the ATmega328P and adds even more useful functionality. This board has a slightly higher cost of roughly \$35. Like the ATmega328P, the ATmega2560 can be programmed using the Arduino software. This simplifies the processes of uploading the code to the microcontroller as well as writing the software for the microcontroller. As you can tell, this can potentially fix the issue with the Arduino Uno in that it can have all of its pins in one board for roughly double the price if it's enough **[ATMEGA 2560]**.

## MSP430G2452

The MSP430G2452 is a microcontroller that is developed by Texas Instruments. This microcontroller can run up to 16MHz and has a variety of features such as:

- Multiple 16-bit timers with different timing configuration modes
- Pull-up Resistors internal to the pins
- 5 Low power modes and 1 Ultra-Low Power mode

This board supports some of the same peripheral features as the Arduino boards including:

- ADC
- GPIO pins
- I2C, SPI interfaces

These boards can be programmed in either low-level C or by using an IDE like Arduino's software development platform. This allows for a simplified way to upload the program to the board. One difference though is that there are not as many supported libraries for this board as there are for the Arduino boards. On a similar note, there are not enough GPIO pins to control all the components that are used for this project **[MSP430G2452]**.

## MSP430FR6989

Another microcontroller developed by Texas Instruments is the MSP430FR6989. This microcontroller is quite like the MSP430G2452 board however it has a significant increase in the number of GPIO pins. This board has 83 GPIO pins as well as 2 UART interfaces, 2 I2C interfaces, 4 SPI interfaces and all the other features of the MSP430G2 board. This microcontroller is more attractive than the MSP430G2452 board due to its increase in the number of pins, however, there remains the drawbacks of the limited software support for this board in comparison to the Arduino line of products. This lack of software support can aggravate the development process as it might be the case that we would have to write our own drivers and external hardware APIs in order to interface specific hardware components. Also note that writing your own libraries and drivers has no guarantee of actually solving the problem as it is very possible that those who've written those same APIs for Arduino have been maintaining them for many years, so any issues with our own codebase would have to be resolved on the spot **[MSP430FR6989]**.

## Summary of Boards

Below, Table 9 summarizes the functionalities and properties of each microcontroller:

	ATmega328P	ATmega2560	MSP430G2452	MSP430FR6989
--	------------	------------	-------------	--------------

Cost	\$2.31	\$13.68	\$ 2.18	\$ 10.32
Digital I/O Pins	14 (6 of which provide PWM output)	54 (15 of which provide PWM output)	16	83
Analog I/O Pins	6	16	-	-
SPI	Yes	Yes	yes	Yes
I2C	Yes	Yes	Yes	Yes
UART	Yes	Yes	No	yes
Operating Voltage	2.7V – 5.5V	1.8V - 5.5V	1.8V – 3.6V	1.8V – 3.6V
Clock Rate	0-8MHz @ 2.7V- 5.5V  0-16MHz @ 4.5V – 5.5V	0 – 2MHz @ 1.8V – 5.5V  0-8MHz @ 2.7V – 5.5V  0-16MHZ @ 4.5V – 5.5V	0 – 8MHz MCLK timer 0 – 16 MHz MCLK timer 0 – 50 kHz ACLK timer 0 – 16 MHz SMCLK timer	0 – 8MHz MCLK timer 0 – 16 MHz MCLK timer 0 – 50 kHz ACLK timer 0 – 16 MHz SMCLK timer
Power Consumption	Active mode: 1.5mA @ 3V – 4MHz  Low power mode: 1µA @ 3V	Active mode: 500µA @ 1.8V; 1MHz  Low Power mode: 0.1µA at 1.8V	Active Mode: 220µA @ 1MHz (2.2V)  Standby: 0.5 µA  Off: 0.1 µA	Active mode: 100µA/MHz
Memory	32 kB flash 1kB EEPROM 2kB SRAM	256kB flash 4kB EEPROM 8kB SRAM	8 kB flash 256B RAM	128kB FRAM 8kB SRAM

Table 9 – Comparison of different microcontrollers

## Design Choice



After comparing the microcontrollers above, we decided that the board that best suits our design goals is the ATmega2560 microcontroller due to its compatibility with the user-friendly Arduino software as well as its abundance of GPIO pins. The ATmega328P simply does not have enough pins, the MSP430G242 has a lack of software support on top of not having enough pins. The MSP430FR6969 does have enough pins and is, in fact, a very familiar board with us, but it is simply not practical to rewrite the software libraries from scratch for it.

### ATMEGA16U2-MU

As mentioned above, the sole purpose of this chip is to act as a USB to Serial interface between the computer and the main microcontroller IC. Another attractive feature of this microcontroller IC over using an FTDI is that this chip is cheaper than FTDI. FTDI can be found online for roughly \$3-4 dollars whereas the ATMEGA16U2-MU is only \$2.53 **[16U2 Cost]** As a result, we do not take into consideration some of the properties that were considered for the microcontroller driving the robot. Some of the only necessary features are listed below in the Table 10:

	ATMEGA16U2-MU
Cost	\$2.53
Digital I/O Pins	22
USART	yes
SPI	yes
Operating Voltage	2.7 V - 5.5 V
Clock Rate	8MHz at 2.7V 16MHz at 4.5V
Memory	16KB In-System Self-Programmable Flash 512 Bytes EEPROM 512 Bytes SRAM

Table 10 : Properties of ATMEGA16U2 IC **[DATASHEET 16U2]**

One of the only considerations to make for this IC is operating voltage. The input voltage of the PCB will have a constant regulated 5V, thus the ATMEGA16U2-MU chip will be sufficiently powered. To properly set up this chip to program the ATMEGA2560 microcontroller chip, we will use the following open-source schematic from Arduino's Atmega 2560 microcontroller board. The schematic is shown below in Figure 34 :

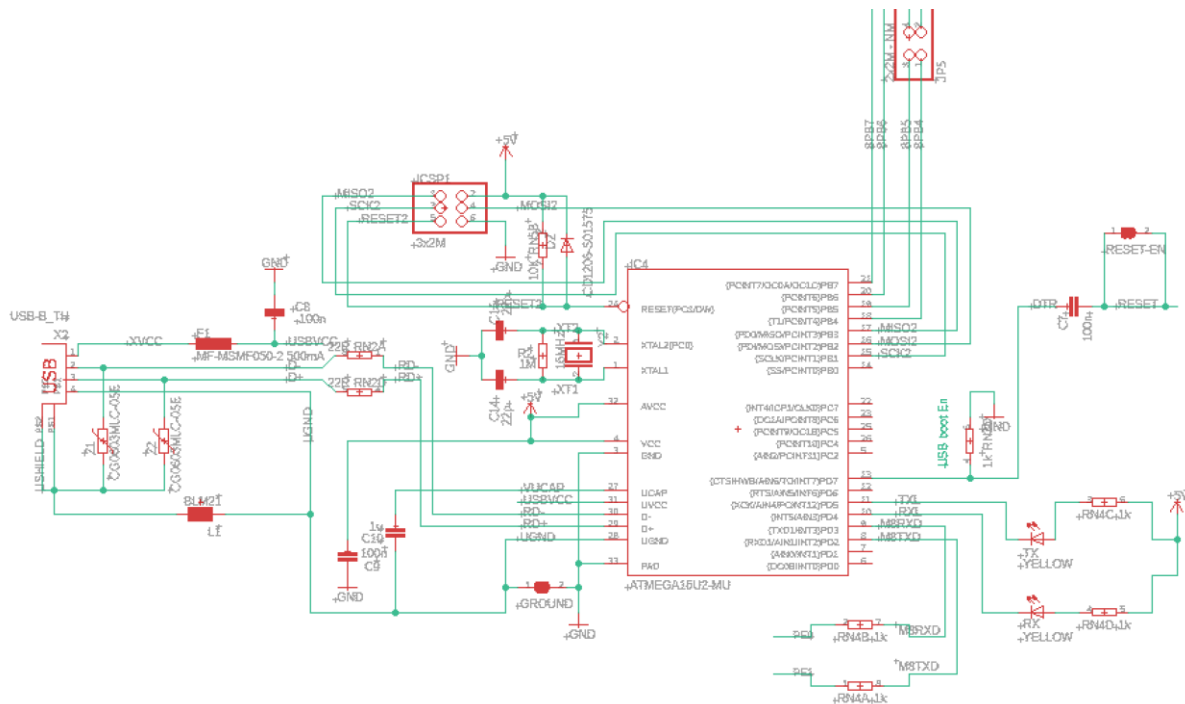


Figure 34 - ATMEGA16U2-MU IC wiring schematic for ATMEGA2560 IC as USB-to-Serial (Open-source from Arduino)

As mentioned in the paragraph above, the reason for using this electrical schematic for the ATMEGA2560 IC is that this wiring is a well-tested design for this specific purpose. Additionally, this is an open-source schematic from Arduino that works with the specific microcontroller IC that we are using in this project so it is clear that there should be no problems with this method of programming the board.

## Wireless Connectivity

Wireless connectivity is a form of communication over some medium other than wires. Some common examples of wireless communication include Wi-Fi and Bluetooth.

Wi-Fi is a commonly used wireless communication protocol that accommodates to the standards set by IEEE 802.11. In general, wireless devices such as phones or computers connect to access points by radio waves to access the internet. Wi-Fi is a useful tool since it can allow a device the ability to connect to the internet and therefore reach any other device also connected to the internet. As a result, we drafted ideas on utilizing Wi-Fi to simplify the user interface to a web application. An easy to implement Wi-Fi in a project is by using Wi-Fi transceiver modules. These inexpensive devices can be connected to a microcontroller and then send and receive data packets over the internet providing a simple and effective way to add wireless connectivity to a project. Despite the simplicity of adding Wi-Fi capabilities to our project, it did not appear to be a necessity for the prototype. In addition to this fact, there are other drawbacks. The most noticeable problem is that if there is loss of connection to the internet or poor internet connection in general,

unexpected issues may arise. Also, adding extra hardware for wireless communication further complicates the project and leads to more entry points to hardware failure. For the prototype of our project, it does not seem wise to implement Wi-Fi as it does not bring any major benefits to the end goal. **[WIFI]**

Bluetooth is another widely used wire form of communication that involves the short-range transfer of data among other Bluetooth connected devices. Unlike with Wi-Fi, devices connected over Bluetooth do not access the internet. The IEEE implemented a standard for Bluetooth as IEEE 802.15.1, but this standard is no longer maintained. However, Bluetooth is now managed by the Bluetooth Special Interest Group. For this project, Bluetooth could be used for some of the same reasons as implementing Wi-Fi but removes some of the problems associated with poor network connection. However, after more consideration with the team, it was decided that there does not need to be Bluetooth connectivity. The issues with complicating the hardware outweighed the benefits of having a modernized user interface. **[Bluetooth]**

Wireless communication would provide a modernized interface with the robot. However, the cons of including more hardware further decentralizes the logic of the robot which could result in possible failures. Wired communication is not only reliable but also cheaper. Most of the wired communication protocols are already available on many if not all microcontroller boards. If anything, the only real place in which any wireless communication might be used is if there is any intention to interface the device with a remote, but that is likely out of question. To summarize, any inclusion of any form of wireless communication will likely have to be fairly simple to implement and be worth considering. Otherwise, we maintain the essence of the original design by leaving it out.

## Motors

Motors are an essential electro-mechanical component that can be used in many robotic applications. For this project, we will focus on DC motors. DC or direct current motors work by converting electrical energy from an external power source into mechanical energy. The electrical energy can be supplied by a battery or even an AC-DC converter. In short, a DC motor works when an electrical current flows through the motor inducing a magnetic field. This magnetic field in conjunction with magnets in the motor propel the motor to spin **[DC MOTOR]**.

In this project, motors will be used to drive belts that will move chemical well plates to different stages in the chemical transfer process. Also, motors will be used to position the pin transfer tool between different workspace rails as well as raise and lower the pin transfer tool for dispensing and collecting chemicals. Since the applications of the motors will be the same for both the positioning of the pin transfer tool and the moving of the chemical well plates to different stages of the machine, we can use the same motors for both. This will simplify the selection of the motor drivers as well. However, one of the motors, more specifically, the linear actuator needed for controlling the vertical positioning of the pin transfer tool is of a different variety than the other motors. Thus, we will discuss this motor separately in another section.

## Motor Types

There are two types of motors that are suitable for this application of this project. Namely, they are stepper motors and servo motors. Both stepper motors and servo motors are used in similar applications. CNC machines, which this project is quite similar in nature to, can be developed with either servo motors or stepper motors. Both motors work fundamentally the same way: an electrical current induces a magnetic field which propels the motor to spin which in turn produces torque. The differences in the motors boil down to the construction and implementation of each. We will now go into detail of the two types of motors. **[DC MOTOR]**

## Stepper Motors

Stepper motors are composed of a single armature surrounded by permanent magnets in conjunction with a fixed stator which houses the windings of copper coil. As an electrical current flows through the windings, a magnetic flux interacts with the magnetic field distribution of the armature which induces a turning force. Stepper motors consist of many poles or magnets which allow the rotations to be controlled in increments of steps. The more poles there are the more continuous the rotation of the stepper motor appears. Stepper motors can produce accurate incremental motion by relying on the steps. This leads to one of the main advantages of stepper motors over servo motors in that stepper motors do not need external hardware such as an encoder or resolver to indicate positioning. Since the stepper motor does not know exactly where the step is or how many steps were precisely taken, this implies that stepper motors exist in an open-loop system. This means that the number of steps may not be entirely accurate (+/- some steps). However, the drop in accuracy can be negligible in most cases or can even be supplemented with encoders or resolvers to remedy the loss in accuracy. But overall, the price of a stepper motor is often cheaper and more realistic than a servo motor. Stepper motors primarily operate at a slower pace in the order of 1500 RPM or fewer. **[Servo Vs. Stepper]**

## NEMA 17 Stepper Motor

First, we will discuss the NEMA 17 stepper motor. This stepper motor has the lowest amount of torque among the other stepper motors at 76 oz\*in. Similarly, this is the cheapest of the three types of NEMA stepper motors. Of the three stepper motors, this has the smallest weight which would decrease the total weight of the machine. Overall, this motor meets the main requirements that we need for this project since it not only has a lower torque which would be better for slow movement speeds but also is the cheapest of the bunch of motors. This motor will be able to properly move well plates across a specific stage as seen in Figure x above. However, it is unlikely that this motor will be able to handle the weight of the load on the gantry. More specifically, the NEMA 17 stepper motor does not have enough torque to move the combination of the pin transfer tool and the micro-linear actuator pair. **[NEMA 17]**

## NEMA 23 Stepper Motor

Next, we will discuss the NEMA 23 stepper motor. This motor has a slightly longer frame size at 2.3 square inches and supplies more torque at 175 oz\*in. Along with the increase in torque and frame size, the cost of the NEMA 23 stepper motor is about \$10 more per unit. Other than the increase, its longer frame size increases the weight of the device by almost double. This stepper motor will provide enough torque to move the combination of the micro-linear actuator and the pin transfer tool effectively and accurately. Unlike the NEMA 17 stepper motor, the NEMA 23 has a slightly longer shaft size which would allow it to be perched off of the gantry allowing for more clearance in the housing of the machine. **[NEMA 23]**

## NEMA 23 High-Torque Stepper Motor

Next, we will discuss the NEMA 23 high-torque stepper motor. This stepper motor has the same frame size as the regular NEMA 23, however, it outputs twice the amount of torque as the NEMA 23 and weighs more than 2 times the weight as the NEMA 23. This motor seems to be overkill for this project since it provides an excessive amount of torque that is not necessary for this application. **[NEMA 23 HT]**

Pictured in Figure 35 below is an example of what a NEMA 23 motor:



Figure 35 - NEMA 23 stepper motor (permission from OpenBuilds.com)

All of the NEMA motors look similar to each other with the main difference being the diameter of the face of the motor.

## NEMA Motor Comparison Table

Now that we have described the different types of NEMA stepper motors, we will now summarize the properties of each of the stepper motors in Table 11 below to easily see how they compare:

	NEMA 17	NEMA 23	NEMA 23 high-torque
Cost	\$17.99	\$27.99	\$43.99
Torque	76 oz*in	175 oz*in	345 oz*in
Input Voltage	12-24 VDC	12-48 VDC	24-48 VDC
Step angle	1.8 degrees	1.8 degrees	1.8 degrees
Shaft size	5 mm	6.35 mm	6.35 mm
Weight	0.35 Kg	0.75 Kg	1.2 Kg
Rotor Inertia	68 g*cm <sup>2</sup>	300 g*cm <sup>2</sup>	670 g*cm <sup>2</sup>

Table 11 – Summarizing the properties of different sized NEMA stepper motors

## Servo Motors

The construction of the servo motor is like the construction of the stepper motor in that the servo motor also has a fixed rotor or armature with permanent magnets as well as a fixed stator composed of a number of copper windings. This motor works with the same principle of an input of electrical current and an output of the rotation or spinning. One of the main differences in the construction or implementation of the servo motor compared to the stepper motor is that the servo motor has fewer poles or fewer magnets. This reduces the incremental step process used in stepper motors. Because of this, servo motors must be used in a closed-loop system to ensure accuracy of the movement of the motor. Servo motors are more technologically advanced than stepper motors. They can be roughly 2-5 times faster than stepper motors and provide much more torque. The closed-loop system ensures that the motor is positioned accurately since output of the motor is fed back to the input of the motor which tweaks the positioning constantly.

### **[Servo vs Stepper]**

Table 12 below shows the main differences between the two types of motors:

	Stepper	Servo
Cost	\$	\$\$\$
Accuracy	Error of a few steps	Continuous adjustments
Speed	0 - ~1500 RPM	0 – ~5000 RPM
Torque	Reliable at low speeds	Excellent at high speeds
System type	Open-loop	Closed-loop

Table 12 – Summary of properties between stepper motors and servo motors

## Design Choice

As seen in the information above as well as in the summarized outline in Table x above, the servo motor has better performance at a higher price whereas the stepper motor has a relatively suitable performance for most applications at a much lower price. For the application of this project in general, the positioning of the pin transfer tool does not need to be perfect. There is enough room for error such that a stepper motor may over-step or under-step its rotations without ruining the process of chemical transfer. Additionally, it is more cost effective for us to use stepper motors rather than servo motors in this project since we are quite limited on expenses. As a result, we will use stepper motors in this project.

Now that we have decided to use stepper motors for this project, we need to select the specific motor. Stepper motors come in a variety of different sizes, speeds, torque, however there is a generalized set of standards implemented in stepper motors that dictates the tolerances of the stepper motors. These standards are developed and upheld by the NEMA or the National Electrical Manufacturers Association. NEMA step motors are stepper motors that are ensured to have a standardized quality and performance as specified in the product's technological specifications and properties. As a result, NEMA stepper motors are well-trusted and reliable. NEMA stepper motors are labeled as NEMA XY where the XY defines the square size of the frame of the stepper motor as X.Y inches squared. For example, a NEMA 17 stepper motor is a stepper motor with a frame size of 1.7 inches squared.

As the frame size is increased, the higher the torque in the motor. Typical units of torque for stepper motors are in ounces x inches or oz\*in. For this project, we are not interested in a hyper-reactive motor with a high torque output because the chemical well plates that are transferred along the machine must be kept stable enough such that the liquid does not spill. Therefore, slower speeds and lower torque are not only acceptable but necessary.

Now, we will compare some NEMA stepper motors to determine which one best suits the needs of the machine. It is worth noting that this project might necessitate different motors for different parts of the machine. For instance, there are 3 stages that will be used to transport well plates across the length of the machine as seen in Figure 36 below. The three stages are Cells, Chemical and Washing.

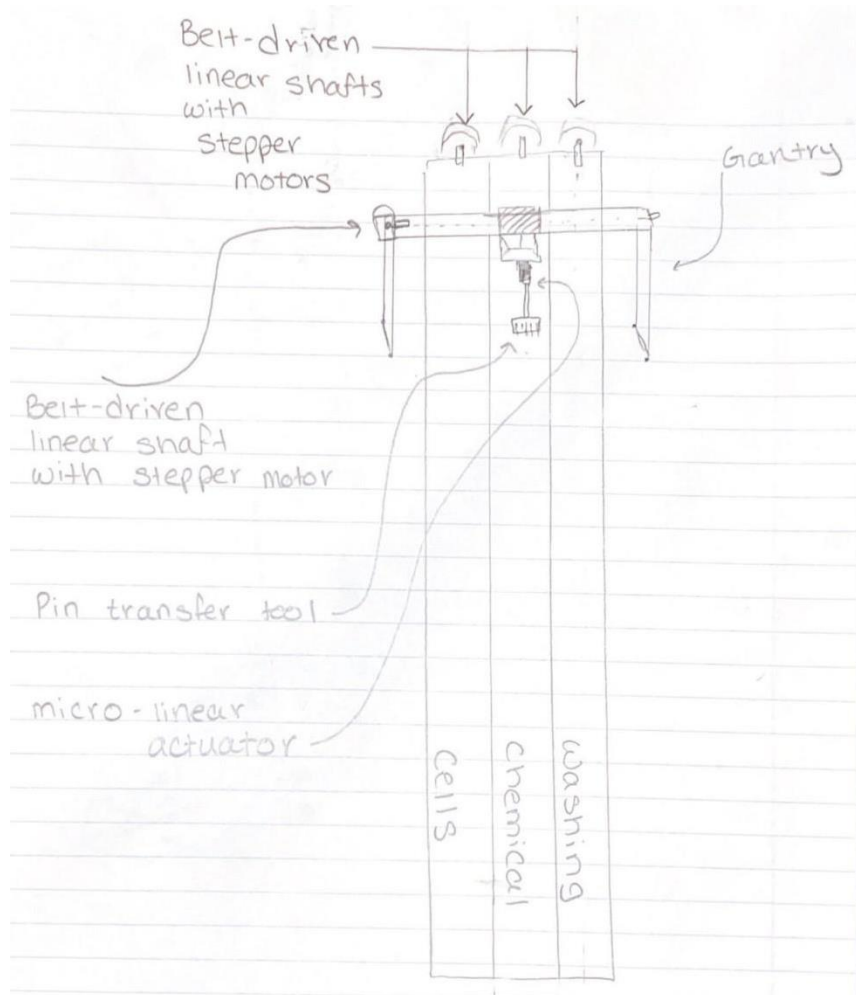


Figure 36 – Example of machine stages and positioning of motors as well as pin tool

Each stage will need its own individual belt-driven motor system to move a chemical well plate to different stages. Each of these three stages will use the same motor since they each perform the same task. However, there is another stepper motor that will position the pin transfer tool to one of the three stages. This belt-driven linear actuator will be across the gantry and will be used to position the pin transfer tool to a specific stage. The motor controlling this belt might need more torque to drive the combination of the pin transfer tool and the micro linear actuator controlling the vertical positioning of the pin tool. As a result, another motor may need to be selected.

In conclusion, we decided to go with the NEMA 17 motors for each of the three stages of the machine. This motor will supply enough torque to move a well plate across a stage



and is a cost-effective option. Also, since we need more torque for the belt-driven linear actuator on the gantry, we will use a NEMA 23 stepper motor.

## Motor Driver

Although some motors can be controlled with solely a microcontroller, larger and more powerful motors need a higher current input than a microcontroller can produce on its pins. One component that can act as a middleman between the microcontroller and the motor is a motor driver. Motor drivers have a similar working principle as relays in that a smaller input signal controls a larger output signal. A microcontroller can control a motor by first passing its signal to the motor driver which then amplifies the current through a certain configuration of FET amplifiers to a suitable level for the motor to use. Without motor drivers, microcontrollers would be unable to effectively utilize the full capabilities of a motor.

For this project, we are using two different types of motors: NEMA 17 and NEMA 23. Since the NEMA 23 stepper motor has different properties than the NEMA 17 motor such as weight and output speed, not all motor drivers are compatible with the NEMA 23 motor. In fact, there are a wide range of options for the NEMA 17 stepper motor but far fewer for the NEMA 23. In this section, we will look through different stepper motor drivers to determine suitable drivers for both the NEMA 17 and the NEMA 23 motors.

## TB6600

First, we will discuss the TB6600 motor driver. This motor driver can be used to drive both NEMA 17 and NEMA 23 stepper motors. This is one of the only stepper motor drivers that is compatible with the NEMA 23 stepper motor. This driver is capable of controlling a two-phase stepper motor. Additionally, this motor driver can be controlled via a microcontroller such that more complex timing and pulse regulation can be programmed to handle the motor. This driver supports up to 32 micro step control. With larger current in these motors comes more heat dissipation. As a remedy for this, the TB6600 provides a large-area heat sink to reduce the thermal energy on the driver. **[TB6600]**

Other safety features include:

- Anti-reverse input protection
- Overheat protection
- Over-current protection
- Short circuit protection

The TB6600 motor driver can be controlled with straight C++ code or the AccelStepper library that will be discussed more in detail later in this section. This allows for easy control and utilization of the motors. The motor driver itself has multiple input and output pins; however to control the motor driver, only 3 pins on a microcontroller are required. In the case of our project, if we use this motor driver for each of the motors, then we would need 12 pins on the microcontroller reserved. If additional motors are needed for this machine, then including more TB6600 motor drivers will increase the pin count by 3 pins per motor.

The TB6600 motor driver is one of the more expensive motor drivers at roughly \$10 per driver.

## TB6560

Next we will discuss the TB6560 motor drive. This is the predecessor of the TB6600 motor driver mentioned above. Similar to the TB6600 driver, this driver is capable of controlling both NEMA 17 and NEMA 23 stepper motors. Despite some differences in the input voltage requirement and the output torque, the TB6560 is quite similar to the TB6600, although the TB6600 has a finer micro-step resolution and is cheaper. Another similarity between the TB6560 and the TB6600 is that they both use the same code libraries. In other words, it is possible to program the operation of this motor driver with either regular C++ code or with the AccelStepper library. The main difference between this motor and the TB6600 is that this motor only requires 2 pins of a microcontroller rather than the 3 pins the TB6600 needs for operation. This is a notable feature of this driver. With fewer pins need be required, more pins can be allocated elsewhere. Similar to the TB6600, the TB6560 comes with a variety of safety measures to ensure that the driver performs effectively overtime. **[TB6560]**

Some of these safety features include:

- Over-voltage protection
- Short-circuit protection
- Automatic idle-current reduction

## A4988

The A4988 motor driver is capable of only controlling the NEMA 17 motor driver. Compared to the previous two stepper motor drivers, the A4988 driver is much smaller and can easily be integrated onto a PCB. This driver offers up to 16 micro-step resolution with only using 2 pins on a microcontroller. This is a huge advantage over the other two stepper motor drivers. As mentioned before, freeing up pin space is significant since it would allow us pins for other components on the microcontroller. This motor driver can be controlled using the AccelStepper library or just C++ code. **[A4988]**

This driver also comes with the following safety measures:

- Current limiting
- Cross-over current protection
- Short-to-ground protection
- Short load protection

## L298N

The L298N motor driver is capable of only controlling the NEMA 17 motor driver. This motor driver interfaces with a microcontroller with 4 pins. Unlike the other motor drivers mentioned thus far, the L298N stepper motor driver, can be controlled using regular C++ code, the AccelStepper library and even the Stepper library provided by Arduino in the

Arduino IDE. One drawback of this motor drive is that the L298N does not have a simple method of current limiting. This means that it can release a great deal of thermal energy in a relatively short period of time. This thermal energy or heat can be dangerous since it may be able to damage some of the components of the motor driver and the motor itself. This is one of the cheaper motor drivers at roughly \$2.20 per driver. **[L298N]**

## DRV8825

The DRV8825 motor driver is capable of only controlling the NEMA 17 motor driver. This stepper motor driver has a resolution up to 32 micro-steps and even has a sleep mode. This motor driver is quite similar to the other NEMA 17 stepper motor drivers and is the cheapest of the bunch.

In addition to this, it has the following protection and safety features:

- Overcurrent protection
- Thermal shutdown
- Undervoltage lockout

Like the L298N motor driver, this drive needs only 2 pins to be controlled by a microcontroller. This provides a lot of flexibility with the selection of microcontroller since such few pins are used. Other than the above, this motor controller is the cheapest of all of them at a price of \$1.80 per driver. **[DRV8825]**

## DM542T

The DM542T is another stepper motor driver with considerable advantages over the other drivers listed above. This driver has advanced digital signal processing technology to provide a fully digital motor driving experience. Due to its advancements in DSP, this driver has much smoother steps which produce almost no audible noise. In the case of a lab setting, loud stepper motors can be an annoyance. Having little to no audible noise will eliminate any such issue. Another unique feature of this driver over the other is auto-identification and configuration of motors. This means that the driver is able to detect which motor is attached and can configure internal parameters to optimize performance **[DM542T]**. As a result of these advanced features, this driver has a higher price compared to other drivers that were already mentioned.

Below are a list of some of the safety features this driver supports:

- Over-voltage protection
- Over-current protection

According to the datasheet of this motor driver, it is recommended to supply this driver with a power supply that can sustain an output voltage of +20 V to 45 V DC. Like most stepper motor drivers, this driver has programmable switches to configure the microstep of the motor. This driver can be programmed with either plain Arduino C++ code or by using the AccelStepper.h C++ library. This motor driver can also be used for both the NEMA 17 and NEMA 23 stepper motors. **[DM542T]**

## Code Libraries

As mentioned earlier, a motor driver is the interface between the microcontroller and the motor. In the case of these motor drivers mentioned above, the stepper motor drivers perform a bit more complex logic than regular motor drivers. Because of this, libraries were developed to simplify the programming process.

### C++

All motor drivers can be programmed using the regular C++ programming language. To control the motor drivers, the input pins of the motor driver are set to oscillate between HIGH and LOW signals for a variable amount of seconds to get the desired amount of steps. Using the regular C++ code without any additional libraries means that to accurately program the motors requires trial and error.

Below are the following functions used to interface with the motor drivers:

- `digitalWrite()`
- `delayMicroseconds()`

### [ARDUINO REF]

#### Stepper.h

This is a library provided by the Arduino company to interface with stepper motors. Here a stepper object can be instantiated to represent a single stepper motor. Then simple methods such as `setSpeed()` and `step()` can be applied to the object to control the motor. This provides a simple and effective way to control the stepper motor.

Here are the following methods used in this library:

- `setSpeed()`
- `step()`

#### AccelStepper.h

This is another library used to control a stepper motor. Similar to the Stepper.h library this library employs an object oriented approach to interface with the stepper motor driver. In this library, the acceleration of the stepper motor is exploited.

Here are some of the object methods that can be used on the AccelStepper object:

- `setMaxSpeed()`
- `setSpeed()`
- `runSpeed()`
- `setCurrentPosition()`
- `moveTo()`
- `setAcceleration()`

- runToPosition()

As seen above, this library supports much more functionality than the Stepper.h library.

### [STEPPER.H REF]

Table 13 below summarizes the different properties of the motor drivers.

	TB6600	TB6560	A4988	L298N	DRV8825	DM542T
Cost	\$10.99	\$19.00	\$2.20	\$2.50	\$1.80	\$19.90
Input Voltage for Motor (VDC)	9 - 42	7-32	8-35	5-35	8.2-45	20-45
Compatible motors	NEMA 17 NEMA 23	NEMA 17 NEMA 23	NEMA 17	NEMA 17	NEMA 17	NEMA 17 NEMA 23
Pins used by Microcontroller per driver	3	2	2	4	2	3
Compatible Library	AccelStep per or C++	AccelStep per or C++	AccelStep per or C++	Stepper or AccelStep per or C++	AccelStep per or C++	AccelStep per or C++
Microstep Resolution Minimum	1,2,4,8,16, 32	1,2,8,16	1,2,4,8,16	1,2,4,8	1,2,4,8,16, 32	1,2,4,8,16, 32,64,128, 5,10,20,40 ,50,100,125

Table 13 – Summarizing the characteristics of different motor drives.

### Design Choice

To conclude, we decided to use the DM542T stepper motor driver since it has the most advanced stepper motor driver technology as well as having the highest resolution in terms of microsteps. This allows for precise operation for the placement of the pin transfer tool with minimal noise output.

### Linear Rails

Linear actuator rails will be used to shuttle well plates from one stage to another. There are many types of linear actuator rails that can be used for this application namely, belt-driven, lead screw-driven and even chain-driven. However for the design of this machine, only belt-driven and lead screw-driven are considered.

## Belt-Driven

Belt driven rails convert the rotary motion of a rotary actuator into linear translational motion through the use of a timing belt. In Figure 37 below, the thin black timing belt can be seen. The timing belt is flat on the outside of the belt and has teeth on the inside of the belt. These teeth are included to prevent slipping along the rail. When the stepper motor turns, it rotates the timing belt precisely based off the number of steps taken by the stepper motor. As a result anything placed on the belt path will move accordingly in the same number of precise steps **[RAILS EXPLAINED]**.

Below pictured in Figure 37 is an example of a belt driven linear actuator rail:



Figure 37 - Belt driven linear actuator (permission from OpenBuilds.com)

Listed below are some of the primary advantages and disadvantages of belt-driven rails **[RAILS]**.

### Advantages:

- Long Strokes
- High Linear Travel Speed
- Higher Efficiency
- Lower input RPM
- Higher Duty cycles

### Disadvantages:

- Higher cost
- Lower accuracy and positional repeatability

- Velocity ripple
- More input torque needed compared to screw drives
- Short stroke decline
- Belt materials
- Belt retensioning

## Lead Screw-Driven

In contrast to the belt-driven rail, screw driven rails use the rotational from the attached stepper motor to drive the rotational motion of the screw. Anything placed on the screw will also be transferred precisely based on the stepper motor's step rate. In Figure 38 below, the long screw can be seen. The mounted carriage is fixed on the grooves of the screw and is able to move based on how the screw rotates **[RAILS EXPLAINED]**.

Similarly, pictured below in Figure 38 is an example of a lead screw linear actuator rail:



Figure 38 - Belt driven linear actuator (permission from OpenBuilds.com)

Listed below are some of the primary advantages and disadvantages of belt-driven rails **[RAILS]**.

### Advantages:

- Lower cost
- Higher Accuracy and positional repeatability
- Quick response in short stroke applications
- Smoother and quieter
- Light load, high duty

### Disadvantages:

- Limited load capacities
- Limited speed
- Not recommended for high low, high speed and continuous duty

The use of the rails can be seen in the drawing of the machine in Figure X above shown on page 71. There will be one linear actuator rail to move a pin transfer tool across the three stages horizontally, and there will be 3 linear rails spanning the entire length of the machine to transport well plates from one stage to the next.

The linear rail needed to move the pin transfer tool across each stage will not travel along distance. In fact, it will only require short strokes. This is one of the drawbacks of belt-driven linear rails since the repeated motion of short movements will unevenly wear the belt down. On the contrary, lead screw-driven excels in this scenario. Short strokes do not wear down the screw rail since the material of the rail is less-prone to wear. Screw-driven rails prevail again since they are more precise than belt-driven rails. In this specific case where the pin transfer tool must be positioned precisely above a well plate, screw-driven rails take the cake **[RAILS]**. For these reasons, it makes sense to use a screw-driven linear rail actuator.

The linear rails needed to move well plates across the length of the machine will travel long distances at a faster rate. Belt-driven linear rails excel in this scenario since belt-driven rails can move at a much faster rate than screw-driven rails **[RAILS]**. For these reasons, it makes sense to use belt-driven linear rails for transporting the well plates from one stage to the next across the length of the machine.

## Part Selection

We have decided to use the 1000mm belt-driven V-SLOT NEMA 17 linear actuator rail from OpenBuilds.com for each of the three linear rails needed to transport well plates across the length of the machine **[BELT RAIL]**. Also, we decided to use one 500mm screw-driven V-SLOT NEMA 23 linear actuator rail also from OpenBuilds.com for the linear rail to position the pin transfer tool over one of the three stages **[SCREW RAIL]**.

Below in Table 14 shows some of the properties of each of the rails chosen:

	V-SLOT NEMA 17 (belt)	V-SLOT NEMA 23 (screw)
Price	\$ 103.99	\$ 151.99
Length	1000 mm	500 mm
RailType	Belt	Screw

Table 14 - Properties of rails chose

## Linear Actuator

Linear actuators are motor powered parts that extend and contract along a single axis. Linear actuators will be used in two places on the Pin Transfer Tool: one will push well plates from the plate stack onto the conveyor rail and the other will allow the pin tool to



extend into the well plates to take or deliver chemicals. Shown below in Table 15 are the differences in some of the properties of two linear actuators considered.

<b>Name</b>	<b>Micro Linear Actuator PA-07</b>	<b>L12-R</b>
<b>Manufacturer</b>	Progressive Automation	Actuonix
<b>Cost</b>	\$69.99	\$70.00
<b>Stroke Size</b>	12.7mm, 20.3mm, 25.4mm, 50.8mm, 101.6mm, 152.4mm, 203.2mm, 254mm, 304.8mm	30mm, 50mm 100mm
<b>Retracted Length</b>	93.5mm, 101.1mm, 106.2mm, 131.6mm, 198.4mm, 249.2mm, 300mm, 350.8mm, 401.6mm	82mm, 102mm, 152mm
<b>Extended Length</b>	106.2mm, 121.4mm, 131.6mm, 182.4mm, 300mm, 401.6mm, 503.2mm, 604.8mm, 706.4mm	112mm, 152mm, 252mm
<b>Maximum Force</b>	22N	80N
<b>Maximum Speed</b>	15m/s	25m/s, 13m/s, 6.5m/s
<b>Input Voltage</b>	12V	6V
<b>Max Duty Cycle</b>	20%	20%
<b>Noise Level</b>	45dB	55dB

Table 15 - Linear actuator selection

There are strict size limitations, so it is better to use models with smaller bodies. The models mentioned in Table 15 were chosen because of their compact size. Both models cost the same amount. Both models have a number of stroke sizes available that allow them to be customized for both of the uses mentioned previously. With so much of the two models being equally good, the main issue comes down to the maximum load each model can sustain. Neither of the models' datasheets mention the maximum load that each model can hold. Therefore, it is being assumed that a greater maximum force correlates to a greater maximum load since the part must be able to sustain a greater force pushing back on it. Also, the pin tools will not weigh more than 1000 grams, so a model able to support at least that much should be suitable. The maximum force of the Actuonix L12-R is almost four times that of the other model, making the Actuonix L12-R the preferred linear actuator for the Pin Transfer Tool. **[L12-R][PA-07]**

## Fans

Once the pin tool has been removed from the cleaning solution, it will have to be fully dried before it can be used again. Waiting on the pin tool to naturally dry will increase the total runtime of a single batch by a large amount. To mitigate the time spent waiting for the pin tool to dry, a fan will be mounted on the reservoir. Once the pin tool has spent the allotted amount of time in the cleaning solution, it will spend another allotted amount of time in front of the fan before it begins a new batch. Below in table 16, we can see some of the differences between the fans that we considered.

<b>Name</b>	<b>CF-014LB</b>	<b>4468</b>	<b>28-1740</b>
<b>Manufacturer</b>	Kingwin	Adafruit	Newark
<b>Cost</b>	\$6.99	\$2.95	\$10.30
<b>Input Voltage</b>	12V	5V	12V
<b>Max RPM</b>	1000 RPM	N/A	2200 RPM
<b>Airflow</b>	58 CFM	N/A	25.98 CFM
<b>Dimensions (LxWxD)</b>	140mm x 140mm x 25mm	30mm x 30mm x 8mm	80mm x 80mm x 25mm
<b>Noise</b>	23 dB	N/A	25 dB

Table 16 - Fan selection

The main considerations are the size of the fan and airflow of the fan. Pin tools are about 128mm by 76mm. The only fan in the table that is larger than the pin tool is Kingwin's CF-014LB. Both Adafruit's 4468 and Newark's 28-1740 are smaller than the pin tool, meaning that multiple of each fan would need to be used to cover the entire pin tool. The 4468's would need to be arranged in a three row by five column array to cover the entire area of the pin tool, which would cost \$39.90 (Adafruit discounts parts bought in bulk). Two 28-1740's would need to be stacked on top of each other to cover the entire pin tool, which would cost \$20.60. By these metrics, the CF-014LB is the cheapest option. The other consideration is the airflow of the fan. Airflow is the amount of air pushed out of the fan, measured in CFM (cubic feet per minute). Fans with a higher CFM should be able to dry objects faster since they are pushing more air per unit time onto the object. The CF-014LB has the highest CFM of the fans in the table. Since it is the most cost effective and has the best airflow, Kingwin's CF-014LB will be used to dry the pin tools. **[NEWARK FAN] [KING FAN][ADAFRUIT FAN]**

## Solenoid Valve

Solenoid valves allow for water to flow when powered, but restrict flow when unpowered. They are very useful in creating seals that can be temporarily broken to release a liquid or gas. Most labs have a kind of vacuum hose used to take in waste liquids or gases that would be able to attach to the output end of a solenoid valve. In the case of the Pin Transfer Tool, a solenoid valve will be part of each cleaning solution reservoir. This would allow for an easier disposal of the cleaning solutions than having to manually dump them in a liquid waste receptacle. Below in table 17 depicts the properties of the solenoid valve that we chose to use in the machine:

<b>Name</b>	<b>2W025-08</b>
<b>Manufacturer</b>	Tailonz
<b>Cost</b>	\$14.99
<b>Input Voltage</b>	12V

Table 17 - Solenoid valve selection

There are many different solenoid valve brands, but they all seem to offer the same sizes, input voltages, and materials, so the main factor comes down to the cost. Larger sizes and input voltages raise prices. Therefore, a smaller valve with a low input voltage will be used. Another reason a low input voltage needs to be used is because the input voltages jump from 12V, to 24V, to 110V. Having pieces that are not central to the main operation of the project take up so much voltage seems like a design flaw. Different materials also raise prices. Stainless steel valves are the most expensive and do not seem the most cost effective for this project. Plastic valves are very cheap, but there is concern that a

plastic valve might not last very long with chemical residuals being stored in and flowing through it. Brass is in the middle of the previous two. It is durable enough, yet not overly expensive, which is why the valves will be brass. The 2W025-08 from Tailonz will be used since it is the least expensive brass solenoid valve available. **[SOLENOID]**

## Power Supply

The means and manner of supplying power to a device is a critical decision to make. In some cases, a device may require battery power if used in a remote location or just may just need a power cord if used in a single location. Despite this, designing a proper power supply is not something to overlook.

The robot that we will be designing for this project will always operate in a fixed location. In other words, it is to be assumed that once room is made for the device it will remain there with minimal significant displacements. As a result, battery power will not be needed. This makes the situation with power supply much simpler. Our device will be powered solely from a wall outlet.

In order to determine the proper power supply to use for this project, it is necessary for us to recognize the required input voltages of some of the components in the machine. Most of the smaller components of this machine do not require the application of a power supply. For example, components such as the following below can be powered from just the 5V output from the microcontroller:

- Keypad
- TFT LCD screen
- Vacuum

However, the following components will require a voltage greater than the available 5V output of a microcontroller:

- NEMA 17 stepper motor
- NEMA 23 stepper motor
- TB6600 stepper motor driver
- TB6560 stepper motor driver
- A4988 stepper motor driver
- L298N stepper motor driver
- DRV8825 stepper motor driver
- Micro-linear actuator (L12-R)

Some of the voltage requirements of the stepper motors and stepper motor drivers can be found in the Table x from the Motor section. The higher the voltage input for these motors the more power output and torque is produced in the motor. Thus, it is beneficial to be able to support close to the higher end of the input voltage range to accommodate for these benefits. However, it is suitable to generate an input voltage in the middle of the range as well to achieve appropriate results.

In the motor selection and motor driver selection part of this report, the TB6600 stepper motor driver and the DRV8825 stepper motor driver were selected. The TB6600 has an input voltage range between 9V and 42V while the DRV8825 has an input voltage range of 8V to 45V. These are the components with the highest input voltage requirements. If these voltages can be accommodated, then any of the remaining input voltage requirements can be handled as well through the use of a voltage regulator circuit.

We will first look at the 24V Meanwell power supply. This power supply as described in the name of the unit, outputs 24V of DC voltage. This power supply can output this voltage at 14.6A. This would make the power output of this unit equal to  $(14.6A * 24V) = \sim 350.4W$ . This unit also has 3 DC output terminals which can be easily used for connecting to individual motor drivers or even to a power bus line. In addition to that, the power supply unit has a built-in cooling fan to ensure that thermal readings are within range to protect the circuitry. **[LRS-350-24]**

Next, we will look at the LRS-150-48 power supply. This power supply unit can output 48V at 3.3A. This equates to roughly a 158W output. The maximum voltage for the motor drivers is around 45VDC. Although this PSU has a maximum output voltage of 48V, it has additional output voltages of 12V, 15V, 24V and 36V. All of these voltages are well within the ranges for the input voltages of all of the components necessary in this project. **[LRS-150-48]**

Lastly, this unit has the following safety features:

- Short-circuit protection
- Overload protection
- Overvoltage Protection
- Overtemperature protection

All these safety features make this power supply unit an attractive option for this project.

Another power supply to look at is the RS-35-48 unit from MEAN WELL. Similar to the previous power supply mentioned above, this unit outputs 48V. The difference between this unit and the LRS-150-48 is that the RS-35-48 unit outputs its voltage at only 0.8 A producing an output of  $\sim 38W$ . This is a much lower power output than the LRS-150-48. Other than this factor and the reduced price of the RS-35-48 compared to the LRS-150-48, it supports all of the same safety features. **[RS-35-48]**.

Below in Table 18, some of the power characteristics of the power supplies are summarized to facilitate the selection of the PSU for this project:

	24V Mean Well	LRS-150-48	RS-35-48
Cost	\$31.50	\$19.80	\$12.70
Output voltages (V)	24	48	48
Output current (A)	14.6	3.3	0.8

Output Power (W)	358	158	38
------------------	-----	-----	----

Table 18 – Summarizing the properties of different power supplies

To conclude, it was decided to choose the 24V Mean Well power supply unit because it has much more output power than the out power supplies. This is important since we are driving a number of motors. With more output power, we can ensure that the functionality of the motors is not diminished by a power supply that cannot keep up with the workload.

## Code Base

As mentioned before, we chose to use the Arduino ATMEGA 2560 microcontroller to manage all functionality of the robot. The Arduino ATMEGA 2560 is usually programmed in the Arduino native programming language which acts as a wrapper library for C++. In addition to this, the popular programming language Python can also be used to program this microcontroller. Both languages provide an easy and intuitive method of interacting with the microcontroller. Since Arduino provides a great deal of well-documented reference to their API of the wrapper library, it is generally an attractive way to interface with an Arduino or Arduino compatible microcontroller. Similarly, Python is an easy and intuitive programming language that has many 3<sup>rd</sup> party libraries and packages from the open-source community. Deciding on which programming language to use for this project in the end comes down to preference of those programming the microcontroller. We will now go into more depth in some of the advantages and disadvantages of each programming language.

### Arduino Wrapper of C/C++

As mentioned before, the team that produces Arduino products has developed a well-documented API wrapper for C/C++ for interfacing with their microcontrollers. This API provides some of the core libraries that are most frequently required for projects such as communication libraries to implement I2C, SPI and other communication protocols, display libraries for interfacing with common types of displays such as LCDs and OLED, and a variety of other valuable libraries. The documentation for these libraries is well-written and is supplemented with sample code to quickly get started with the methods available in the library. In addition to this, Arduino offers a software development platform that allows users to easily connect Arduino compatible microcontrollers and program them. This is one of the hallmark features that sets apart the Arduino from other languages like Python. The IDE or integrated development environment is a graphical user interface that provides features such as a text editor to compose code in, terminal to view progress of code compilation as well as view bugs and other errors, a library manager to manage 3<sup>rd</sup>-party libraries, and many other useful tools to simplify the coding process. Without such an IDE, it takes a great deal of additional effort to upload code to the microcontroller. Though IDEs are known to be a fair bit slower than traditional text editor and compiler combinations, they are made to be heavyweight software products that are prepared for any and all software development and debugging. Standardizing the development environment as well as style early on will ensure the success of our software

development process. Another advantage to using the Arduino C++ wrapper is that Arduino is a universally used language and has a lot of support from the IoT community. There are thousands of forum posts and comments about typical questions regarding the API which will be useful for all steps of the programming of the microcontroller. **[ARDUINO REF]**

## Python

Python is the most popular programming language among beginners due to its intuitive human-friendly syntax and among professionals due to its wide-support from the open-source community. Python can be used with Arduino devices through use of libraries such as CircuitPython, Pyduino, and other user-created libraries that port the functionality of Arduino libraries to a Python base. These libraries provide similar functionality that the Arduino libraries provide but ported to Python. Unlike C++, Python is an interpreted language meaning that it does not get compiled before running. This means that it is platform independent since there is no need for a general compiler for the code. One downside of this is that code can be hastily uploaded to the board without any indication of bugs. The compilation step present in compiled languages like C++ notifies the programmer of such bugs, warnings and errors that can break the code. This prevents unnecessary time wasted combing through code to discover even the simplest of errors. On the other hand, the lack of a compiler for the language reduces the code size drastically. A small code size is an important consideration to make when writing code for a microcontroller since instruction memory is limited. As described above, there are both advantages and disadvantages to utilizing an interpreted language like Python instead of C++. Another advantage of using Python rather than Arduino's language is that there are no restrictions on the code structure. This means that instead of conforming to Arduino's `setup()` and `loop()` structure, code written in Python can be structured in any manner. This adds customizability to the code which can improve performance of the board. One disadvantage of using Python over C++ is the performance of the software overall. If running the software isn't particularly performant, the overhead of using an interpreted language like Python over a compiled language like C/C++ can compound with each pin transfer operation, but this is to be expected. As it stands, the current expectation is that our Liquid Handling Robot will be able to handle at least one pin transfer operation per minute. If it is clear that the software performance is the issue, then the switch to C/C++ should not be particularly difficult, though the overhead will probably be more present in the quality of our servo motors, for example, in which case we can tweak and optimize certain parts of the project to our liking to tailor to the needs of the design specifications. It should be said, however, that due to the popularity of the Python programming language, there will likely be more support for the language and less compatibility issues overall: in essence, we can almost guarantee that any one piece of code we use to accomplish a particular procedure related to programming the Liquid Handling Robot has already been done before in some way, shape, or form and will take advantage of that wherever necessary. Though C++ is also a fairly popular language, it does not have the same favorability and is known for having a lot of non-orthogonal behavior that can make a lot of software in projects break for reasons that are unbeknownst to the amateur C++ hobbyist. Simple things like the use of the bracket operator can quickly lead to all sorts of

undefined behaviors if an array or a vector is indexed out of bounds. While the C++ compiler is a fairly strong one, we can say with experience that it does not do a good job of catching the kinds of errors that people without much knowledge in C++ can even know about and while it is true that it can catch some bad errors before they are spotted in production, the compiler is still very indifferent to a plethora of undefined behaviors that simply exist in C++ as in inherited from C. To add to that, handling memory management of any kind, be it from C or C++ in this project, can be very painstakingly difficult, especially with projects of this calibre. On top of that, the verbose template error generation is often summoned without need on certain pieces of code that have nothing to do with templates.

To conclude, while our team consists of programmers who specialize in both Python and C++, the majority favors Python's intuitive, on-rails, and safe programming platform in which we prefer the bugs to belong to the programmer being inexperienced as opposed to simply being unknowledgeable of the quirks of the compiler as is the case with C/C++. While C/C++ can be used in the event that the performance penalty is large enough, there are a lot of hurdles with the nature of the C/C++ family of programming languages that need to be overcome by developers who program at a scale any larger than a homework assignment, namely, the reduced support relative to Python, the numerous undefined behaviors that it hosts, the indifference of the compiler to such behaviors, and the superfluously verbose and borderline misleading template error generation. In light of this, some of the most popular Python libraries compatible with interfacing with microcontrollers were researched.

Pyduino is a Python library that supports the most basic core methods to interface with hardware:

- `pinMode()`
- `digitalRead()`
- `digitalWrite()`
- `analogRead()`
- `analogWrite()`

These functions are directly mapped from Arduino's code base and make up the majority of all the other libraries created by Arduino. Since Pyduino does not support any other libraries, we would have to manually interface with all other components. One workaround to this is to use additional 3<sup>rd</sup>-party libraries that specialize in certain components. However, one problem with this is that these additional libraries are not regulated nor are they guaranteed to work together without errors. It is possible that software compatibility will be an issue that will need to be handled independently, but handling that will depend on the quality of the source control currently on deck, which is in abundance. **[PYDUINO]**

CircuitPython is a collection of libraries and drives sponsored by Adafruit. This is more attractive than Pyduino since there is some sense of unification among the different libraries. Not only does CircuitPython provide all the functionality that Pyduino does, but also a multitude of other APIs for the components of our project including:



- User Interface Libraries
  - LCD
  - OLED
- Motors and Servos
- Debouncing
- I2C

## [Circuit Python]

### Design Choice

In the end, we chose to implement the code base in the Arduino programming language due to its overwhelming support from not only the Arduino team but also the open-source community. Arduino provides all of the essential libraries that we will need to program the robot. These libraries include the Servo library, the Communication library that provides all the I2C, SPI, and other wire-based communication libraries, the Display library for interfacing LCD and OLED display with the microcontroller, the Stepper library for controller stepper motors, and the Sensor library which provides the essential functionality with interacting with other types of sensors.

### User Interface

One part of a User Interface (UI) is how the user can interact with it. A keypad and touchscreen were considered for the user interaction role. Touchscreens are very common in embedded systems nowadays, so a touch screen would likely be more natural to a user than a keypad. However, the screen that will be used will be small, so there is concern about the frequency of user mis-input. Keypads are still susceptible to user misinputs, but the size of a key will likely be larger than the size of a touchscreen button. Using a keypad will cause less user input errors than a touchscreen, so a keypad will be used for interactions with the UI.

In addition to the keypad, some sort of visual interface is useful. It provides a way to monitor progress of a device and even display any errors that may occur. For this robot, we need a display to give a lab technician or the intended user the following:

- A way to monitor the progress of a cycle.
- A way to determine cycle parameters such as wash time, chemical volume, etc.
- A way to report any errors.

### Keypad

As part of the user interface, the user will be able to specify parameters such as pin tool size, number of well plates to be treated, wash time, etc. We will include a keypad into the project to allow the user to select certain options in the UI as well as enter specific numerical values that will be decoded and then applied to the operation of the device.

Matrix keypads are a common type of consumer-grade keypads that are easy to use in projects with microcontrollers. As the title of the keypad states, each button of the keypad is part of a matrix of rows and columns. To detect which key is pressed, the microcontroller or device controlling the keypad performs an action called scanning where it checks each key or button of the keypad. The scanning procedure begins by setting all of the rows to high. Then, it loops over each row and does the following:

1. Set the current row to LOW
2. Then for the current row, loop over each column
3. If the column is LOW, then that means the key at that row and column is pushed
4. Otherwise, the key at that row and column is not pushed.

Figure 39 below, shows the electrical wiring of a keypad:

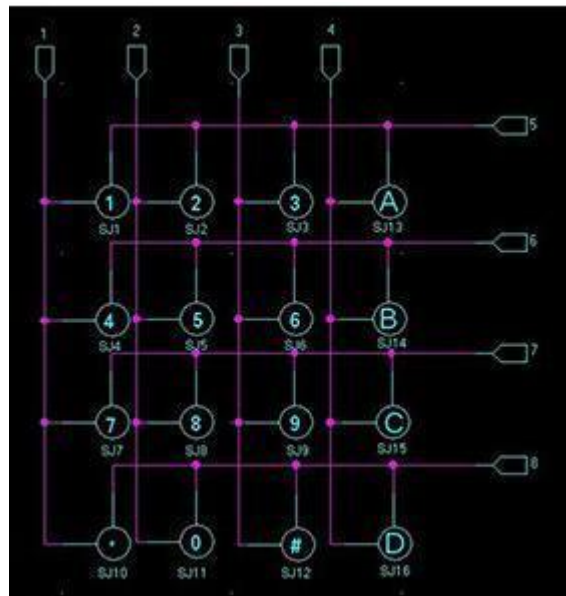


Figure 39 - Rows and Columns of a matrix keypad (permission from Adafruit)

An example for pseudo code of the scanning procedure can be found in the Appendix.

This logic can be programmed into the microcontroller. However, many programs that can be used to control the microcontrollers provide libraries that can easily perform the scanning for us in a high-level function call. **[KEYPAD FUNCTIONALITY]**

All matrix keypads work this way so when choosing a keypad for this project, we need not consider the implications of programming the keypad. Instead, we will consider the following properties:

1. Number of keys
2. Number of pins
3. Input Voltage
4. Life span (Number of presses)
5. Price

## 6. Design

The following are some of the selected matrix keypads that we found suit the needs of this project. We will now compare them to determine which one is right for our project. The first image represented in Figure 40 is a matrix membrane keypad. The image on the right depicted in Figure 41 is a matrix plastic keypad.



Figure 40 - 4x4 Matrix Membrane Keypad



Figure 41 - 4x4 Matrix Plastic Keypad

(Figure 41 has permission from Adafruit. Figure 40 is an image taken by this group)

In addition to these two keypad, we considered another keypad almost identical to figure x on the right, however it has one less column making it a 3x4 matrix keypad. The only difference between the two is the number of columns and thus the number of buttons on the keypad.

We will first discuss the membrane matrix keypad as seen in figure x. This keypad has a membrane key which allows for a softer activation time for engaging the key. This keypad supports input voltages up to 24VDC and has a life span of 1 million enclosures or button presses per key. This is more than enough for the application of this project. This keypad has 8 pins to interface with: 4 for the rows and the other 4 for the columns. The design of this keypad is unique in that it is extremely thin. This is due to the use of the membrane keys. In addition to that, this keypad comes with an adhesive on the back of it, which allows for it to be mounted quite easily and effectively. Lastly, due to the design of the keypad casing, the surface of the keypad is resistant to damage from water since it is one unified piece of plastic.

### **[KEYPAD membrane]**

The keypad on the right in figure x, has the same number of rows and columns as the membrane keypad but does not sport a membrane key. It has a more tactile activation when pressed. This gives the keypad a more professional and robust feel rather than the

soft button impression needed to activate the key of the membrane keypad. However, this means that the keypad dimensions are increased. Similar to the membrane keyboard though, each key has a life expectancy of 1,000,000 key presses as well as an input voltage of 24VDC. Both of the plastic keypads can be damaged by water introduced to the surface of the keypad. **[KEYPAD 4x4]**

Below in Table 19 summarizes the properties of the keypads.

	4x4 Membrane Keypad	4x4 Plastic Keypad	4x3 Plastic Keypad
Cost	Free (Already have)	\$5.95	\$6.50
# Keys	16	16	12
Input Voltage (Max)	24VDC	24VDC	24VDC
Life Expectancy (per key)	1,000,000	1,000,000	1,000,000
Key type	Membrane	Tactile	Tactile
Casing	Ultra-thin plastic	Plastic	Plastic
Dimensions	69.2mm x 76.9mm x 0.8mm	69.0mm x 65.5mm x 9.7mm	70.0mm x 51.0mm x 9.7mm
Force to engage key (g)	140g- 150g	160g-180g	160g-180g
Water resistant	Yes	No	No

Table 19 - Comparing keypads

After comparing the keypads, we chose to use the 4x4 membrane keypad. The reasons leading to this decision include the fact that our team already has this product, but also the dimensions of this keypad. The other keypads would require additional work to install into a prototype. Since the membrane keypad comes with an adhesive backing, we can easily install it anywhere on the surface of the prototype. Lastly, the keypad is protected from liquid damage since the keypad surface is one unified piece of thin plastic. Water can not become trapped within the keys or make its way through to the inside of the device. In a lab setting, liquid can be introduced in many ways to this device so having resistance to water or other liquids is important.

## Screen

The main component of the user interface is the display. The purpose of the display is to give the user a visual interface to the device. With this, the user can see what stage the robot is currently in, how much time is remaining, where errors may have occurred, etc. This interface in the robot should be large enough that an individual can easily comprehend what is displayed from a reasonable distance. Thus, one of the main factors in deciding on which display we will use is screen size. However, before we get into the selection of the part, we will compare and contrast the two common display types: LCD and OLED. Figure 42 below shows an example of a 16x2 LCD screen. These are typically used in consumer hobby electronics projects.

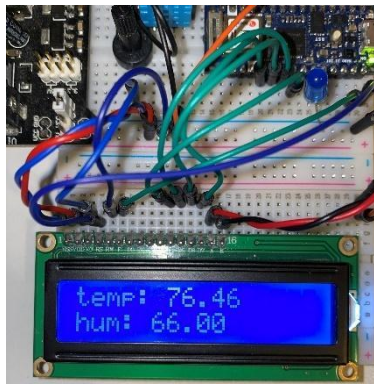


Figure 42 – Application with 16x2 LCD screen. (Taken by Brenden)

## LCD

The name LCD comes from the type component making up the display. The display is composed of liquid crystals which can be controlled depending on a voltage difference applied to them. Liquid crystals themselves do not actually produce any light **[LCD WIKI]**. Instead, an external light source such as the sun or a backlight supplies the light which then the liquid crystals can reflect light in certain ways to establish a desired image, character, sequence of pixels, etc. When a certain voltage is applied to the matrix of the LCD, the liquid crystals in that pixel can be polarized to either effectively turn on or turn off a pixel. However, this infers that the backlight must always be on which consumes energy constantly even if all of the liquid crystals of the display are effectively turned “off”. **[ LCD FUNCTIONALITY][TFT LCD]**

Another variant of LCD screens is a TFT LCD. TFT or Thin Film Technology is a type of LCD hardware that effectively adds more color options per pixel on the screen **[TFT LCD WIKI]**. In the case of Figure x below, this screen supports an 18-bit color range. This is a 2.2” display that is composed of 320x240 color pixels. One benefit to this display over regular LCD displays is that it can be controlled with the SPI communication protocol which only requires 4 pins. Compared to the previous LCD screen, the TFT display is larger, supports more color options, and uses even less pins from the microcontroller. The color pixels would provide a more modernized UI for the robot rather than having a

monochrome LCD screen. Similar to the previous dot matrix LCD screen, this display requires 5VDC for logic and backlight. Due to the more advanced technology used in these LCD screens, the price is slightly higher than the dot matrix LCD screen at roughly \$25. **[TFT]**. Below in Figure 43 is a 2.2 inch screen made of a TFT LCD Display:



Figure 43 - 2.2" 18-bit TFT LCD Display (reprinted with permission from Adafruit)

## OLED

OLEDs on the other hand are composed of individual organic light emitting diodes. OLEDs are constructed as a matrix of the diodes and thus can be programmed specifically. Rather than the need for an external light source, each OLED of the display is self-illuminating. Because of this, it is more power efficient than an LCD display since the backlight must always be on for the liquid crystals to reflect light whereas pixels that are not used can be turned off for an OLED display. **[OLED FUNCTIONALITY]**

The main difference between the two display types is that LCDs have a Light Emitting Diode (LED) backlight layer while OLEDs do not. The additional backlight layer on LCDs causes them to be bulkier and require more power than OLEDs. OLEDs are also more programmable than LCDs. Typically, each pixel is programmable in an OLED while only groups of pixels are programmable in LCDs due to their LED backlighting. OLEDs also tend to have better contrast than LCDs since LCDs are generally restricted by their backlight voltage level. Below in Figure 44 is an example of an OLED display:

### **[LCD VS OLED]**

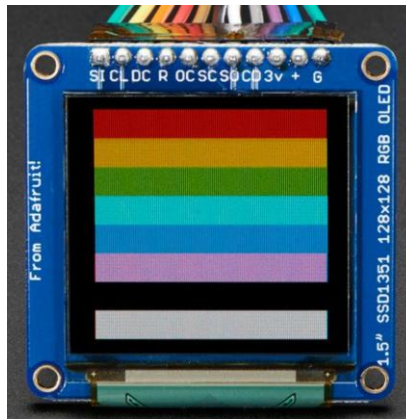


Figure 44 - Application with a 1.5" OLED screen (reprinted with permission from Adafruit)

While LCDs are drawn back by their use of the LED backlight, they do have some advantages over OLEDs. OLEDs can experience an effect called burn-in. This happens when the same pixels on an OLED are active for too long at a time. The pixels may appear to be on, or partially on, even if they have not been activated. This effect is not usually permanent on OLEDs but having this issue for even a temporary time is a downside. LCDs do not tend to experience this issue. LCDs tend to be brighter than OLEDs, so they are more easily readable than OLEDs in darker rooms. OLED displays also tend to be more expensive than LCD displays since they require more complex hardware to function.

### **[LCD VS OLED]**

### **Design Choice**

The deciding factors are display customizability and the visual fidelity. Since this project is completely self-funded, using an LCD is preferable since it has a lower cost. However, we believe that the greater customizability and visual fidelity of OLEDs is worth the tradeoff of having to pay more.

Now that we have a suitable foundation of how both of these displays work which allows us to understand the advantages and disadvantages of each display, we will now compare some displays in order to determine the best display for the user interface.

We will start with a simple LCD module as shown in Figure 45 below **[20x4 LCD]**. This is a 20x4 LCD which means that there are 4 rows each of which can display up to 20 alphanumeric characters. Characters are formed on the screen by turning on groupings of dot pixels. On closer inspection of a character on one of these displays, it is clear that each character is composed of small square dots. This LCD module is attached to a controller that takes care of the low-level assembly instructions for controlling the dot matrices of the display. Instead, we can use a library such as the LCD.h library in the Arduino IDE to program the display. This LCD requires an input voltage of ~5VDC for properly supplying power to both the controller that handles the logic as well as the backlight that produces the light. This board has a total of 16 pins. Here is the breakdown for each of the pins:

GND - Ground  
VDD - Supply voltage for the logic of the controller  
VO - Variable voltage for the LCD  
RS - Used to control the mode of the instruction sent (either data or instruction)  
RW - Read or Write  
E - Chip enable  
D0-D7 - Data lines  
A - 5V backlight supply  
K - 0v ground backlight supply

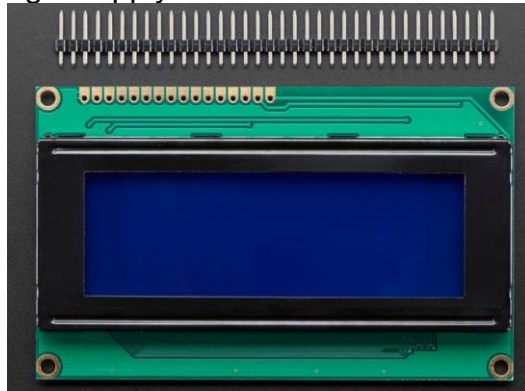


Figure 45 - 20x4 LCD (Reprinted with permission from Adafruit)

The majority of the pins needed to operate this LCD are shared among other components in this project including the common 5V reference and ground. The pins R/W is also set to ground. For this display, only 6 pins are used by the microcontroller:

- RS
- E
- D4
- D5
- D6
- D7
- 

These boards are relatively cheap only costing about \$18.

OLED screens come in many shapes and sizes. For this project, it is desired to have a square screen with a decent size. In the current market of OLED screens for hobbyists, it is clear that smaller OLED screens are desired. It is in fact difficult to find larger OLED screens currently. This may be due to shortages in this current time period or other restrictions on production and trade. OLED screens have many advantages over LCD screens as detailed in the above paragraphs , so it may be worth it to use an OLED for power saving benefits or overall aesthetic.

We will first begin with the monochrome OLED display found in Figure 46 below **[OLED grayscale]**. Although this display only supports 16 levels of grayscale pixel color, it has a reasonable screen size for the user interface. Having multiple colors on the screen is not a necessary feature for the screen. The main factor for our project is readability which



correlates to screen size. This 1.5" display is in the suitable range for this robot and it provides good power saving features due to the OLED technology. This board can be controlled with either I2C or SPI, meaning that it can be controlled with only 2 pins or 4 pins. Similar to the other screens, this screen requires 5V for the input voltage. This screen costs roughly \$25.



Figure 46 - Grayscale 1.5" OLED display (permission from Adafruit)

Below in Table 20, is a summary with the properties of each board:

	20x4 LCD	2.2" 18-bit TFT LCD	1.5" Grayscale OLED
Cost	\$18	\$25	\$25
Operating Voltage (DC)	5V	5V	5V
# of Pins	6	4	4 or 2
Communication Protocol	Controlled by onboard controller	SPI	SPI or I2C

Table 20 - Properties of Displays

After comparing these screens, we decided to choose the 2.2" 18-bit TFT LCD screen since it has the largest screen size and can support 18-bit colors. It is worth noting that some members of the team have experience with programming with this screen already and this saves time by looking into TFT screen programming.

## Part Selection Summary

List below in Table 21, is the selection of parts that were determined to be the most suitable for our application. All the reasoning for the selection of these parts can be found in the Research section of this report.

Part	Selection	quantity	Price
Pin Transfer Tool	VP 407R	1	Free
Microcontroller IC	ATMEGA2560	1	\$ 13.68
Motor 1 <sup>1</sup>	NEMA 23	1	\$ 27.99
Motor 2 <sup>2</sup>	NEMA 17	3	\$ 17.99
Motor Driver	DM542T	4	\$ 19.90
Rail 1 <sup>3</sup>	V-SLOT Screw-driven Linear Actuator Rail	1	\$ 151.99
Rail 2 <sup>4</sup>	V-SLOT Belt-driven Linear Actuator Rail	3	\$ 103.99
Linear Actuator	Actuonix L12-R	1	\$ 70
Fans	Kingwin's CF-014LB		\$ 6.99
Solenoid Valve	2W025-08		\$ 14.99
Power Supply	24V Mean Well PSU	1	\$ 31.50
Keypad	Membrane Keypad	1	Free
Screen	2.2" 18-bit TFT LCD	1	\$ 25

Table 21 - Part selection summary

## Source Control

<sup>1</sup> Motor 1 refers to the motor necessary for positioning the pin transfer tool over a specific track

<sup>2</sup> Motor 2 refers to the motor necessary for transporting well plates across the length of the track to different stages (dispensing, washing, stacking, etc.)

<sup>3</sup> Rail for Motor 1

<sup>4</sup> Rail for Motor 2

Source control, also known as version control, is a software development paradigm that involves the management of incremental changes of software. During software development, code or other documents can be stored and managed such that incremental revisions to the code can be seen. The main feature of source control is that a document that has been managed over some time can be reverted to a previous state. This allows editors of the document to keep track of other editors' work as well as fix errors. For any software developer out there in the field, it is simply crucial that they know how to use every part of the source control system, since that is what ensures that the project's versions remain in a healthy developing state.

## Github

Github is a cloud-based version control that uses Git. It is an incredibly popular and useful tool for programming as it also provides some extra functionality such as bug tracking and continuous integration. Since each member of our team has a great deal of experience with Github, it is the chosen tool for the version control of our code base.

One of the key features of using Github is the ability to use branches to pipeline your code into stages. This is very important for separating development from production. Typically what developers do is that they have code that is put onto a certain branch in production that gets merged closer and closer into the main branch as it gets maintained. The point of this hierarchical structure of development branching is used so that only code that has spent enough time being looked through and maintained in development actually makes it to production for testing, which saves money as it is much faster and cheaper to catch bugs in development than to discover it during production.

On top of this, there is also the use of continuous development as well as continuous integration that can be used to automate the process of building, testing, and deploying code to the main branch. Using a series of smaller commits, you can use a feature called Github Actions. Github Actions is a tool that facilitates CI/CD. It does this by dispatching a job to a linux container that would then attempt the build, test, and deploy scripts. If any of the scripts in this process fails, then the developer is notified of this failure and the deployment is avoided. This is to ensure that any code that makes it to production should at least pass some baseline set of tests. This is obviously also done on top of all of the branching previously mentioned.

Lastly, there is also one more line of defense to ensure that code that makes it to production is properly delivered. There are many plugins on Github that perform static code analysis for correctness. Static code analysis is the process by which a program is inspected and debugged before it is even run.

Another key feature of using Github is issue tracking. Most large companies that do issue tracking on a larger scale tend to use software like Jira, but this is mostly used when they need to manage many issues from a community user base. For example, developers for a video game can have up to hundreds of thousands of concurrent users online playing their game at any point, a sizable portion of which will run into bugs that need to be

reported to the developers as issues. The developers would then, in turn, schedule them into their workload depending on their respective priorities in order to fix them whenever possible. We plan on using a smaller scale version of this with simple issue tracking from Github as we will only be seeing issues that are started by each other. We might also start issues if we see that a particular feature is missing. Alternatively, if we see a quick fix to the issue, one can also start a pull request to propose certain changes to the current code base and only have them make it to the main codebase provided that it gets properly peer reviewed.

Github also allows you to look at code on a commit by commit basis in order to see a direct timeline that accurately describes the process of developing the software at every stage. That makes code much easier to read, as it groups together related chunks of code using the timeline. There is a good reason why Git was developed to manage the linux kernel. Before then, people used to have to read large chunks of code and piece everything together, making for a high barrier to entry when it comes to developing for a codebase. Though you still have to read code and piece it out a lot of the time because you might not necessarily be interested in looking at commits where code simply gets fixed, it can still be very insightful for you to use in order to understand and analyze the kinds of errors that end up in the higher branches.

Currently, some of the main consumer version control software is available on cloud platforms. This is a major benefit since it means an individual with code or documents stored in a version control cloud-based system can access their files anywhere with internet connection.

## Workspace Base

### 3D Printing

The base of the Pin Transfer Tool needs to be custom made due to the uniqueness of the project. The base could be 3D printed. The University of Central Florida (UCF) Senior Design Lab has a 3D printer that is available for student use with instructor permission, so that is the 3D printer that will most likely be used. If permission to use that 3D printer is not secured, then the UCF Texas Instruments Innovation Lab will be the next lab to have permission requested to use. If neither of those labs give permission to use their 3D printers, then a commercial 3D printer will need to be found somewhere in the Orlando area so that the base can be created. SolidWorks will be used to design all 3D printed parts since SolidWorks is compatible with 3D printing. While 3D printing is unique in that it is very accessible and cheap, the process can take a very long time for large builds and the size of any one printed piece is limited by the size of the 3D printer. This could mean that the base would need to be made out of many different parts adding to the complexity of the project. Having the base made out of multiple parts could threaten the stability and integrity of the robot.

### Sheet Metal

The base of the robot could also be made using sheet metal. Sheet metal is more expensive than most 3D printed plastics, however it is much stronger, more durable, and it is easier to clean. Having a base made from sheet metal would be harder to fabricate and could take considerable time and money. A sheet metal base would give the robot a more professional look and would have a clean flat surface that could be sanitized with a large variety of chemicals without worry that it will deteriorate the materials which would be a concern with 3D printed plastics. Sheet metal is best used in large flat surfaces with few bends or corners. Very small intricate pieces can be hard to manufacture with sheet metal. For this reason it may be necessary to make the majority of the base out of sheet metal but for the very intricate interior parts and even the plate stackers should be made out of 3D printed pieces.

## Plexiglass (Acrylic)

Plexi-glass, also known as acrylic or Poly(methyl methacrylate), is a transparent plastic polymer that could be used to make up a majority of the pin transfer robot's base. Plexiglass is cheap, relatively easy to work with, and good for large flat surfaces. It does not bend well but it could be used for the flat faces of the base of the robot. Plexiglass however cannot be cleaned using solvents including alcohols such as 70% ethanol which is a very common cleaning solution and sterilizing agent in laboratories. This is a major drawback to Plexiglass and probably makes Plexiglass an unviable option as a material to build the base of the workspace from.

## Prototype Build Procedure

This topic has undergone much discussion, but since this project is going to require a budget of at least \$1000, we need to ensure that we only build iterations of the project that are necessary to make progress against the goals we have, given the constraints of the project.

First, the pin transfer tool will be bought. Some of the pin transfer tools by V&P-Scientific contain some polar coating to increase the force against the surface tension of water to ensure more adhesion during the pin transfer process. Though this might be of interest if we plan on obtaining a faster pin transfer operation hence yield, this can be left out of the prototype so that we can ensure a successful pin transfer operation. To test that the pin transfer tool works as intended, some testing will be done with food dye. Using food dye makes the most sense as it is both relatively easy and fast to inspect the result of the pin transfer operation and check for cross-contamination. Note that food dye is a particularly good fit for the pin transfer operation as it may take much longer to inspect the result of a pin transfer operation with certain chemicals. It depends entirely on the chemicals used in the operation.

Next, we need to build the workspace rail. This involves setting up the belt-fed linear actuators to be able to move the well plates from one side of the rail to the other. This should not take a lot of time, but it does need to be fairly well supported so that it interacts well with the input and output stacks later on. Once the workspace rail works as intended,

we will walk a well plate through it and see if the linear actuator works as intended. Once we can guarantee that the linear actuator works as intended. We can proceed with the rest of the design. Implementing the workspace rail for the wash steps can be omitted so long as we wash the pin tool ourselves. Implementing the wash steps can take a fair amount of time and may only add to the testing process as it is the longest process within the pin transfer cycle.

From there, the gantry needs to be set up. For the gantry to be properly implemented, there will need to be an accurate SOLIDWORKS/AutoCAD model of it that can be relied upon for implementing the gantry. Assuming we have a solid gantry design, the building aspect should take some time and might even require a few iterations, but shouldn't be too difficult. One important aspect of setting up the gantry robot and mounting the pin tool is the mounting plate. The mounting plate is typically specific to each pin transfer robot. There may or may not be the need for making a custom mounting plate for our purposes, but there are some basic pin tool robots.

For the purposes of building this prototype, we will relegate the building of the input and output stacking mechanisms for later. While implementing the stacking mechanisms for the input and output are important, building on any one design/implementation of the input/output stack can be quite costly and is a large subproject in and of itself. It is very important to be able to get their implementation correct. However, for generating a Minimum Viable Product, the focus will be to use as much of the budget as possible on completing a successful pin transfer cycle or a series of successful pin transfer cycles since that is what determines whether the prototype is a successful proof of concept or not.

## Testing Procedures

After the robot is properly built, we need to perform some testing procedures to ensure the validity of the pin transfer operation. We need to check that the liquid was transferred at the correct volume. In other words, We need our results for the volume transferred to be both accurate and precise.

The notion of accuracy, also known as trueness, that is being referred to is simply about just how correct the current result is with what is to be expected. In general, accuracy is the percentage error between the average value(if we intend to measure for average accuracy) or current value with the expected or accepted value, which is also typically referring to some theoretically arrived result. To properly define accuracy for our purposes, it is simply the percentage error between the average volume of solution measured with the expected/accepted value. If the result is fairly accurate, then that typically means that the error is fairly small. The equation for accuracy is described in the following equation:

$$\text{Trueness \%R} = 100 \times (V_M - V_T/V_T)$$

Equation 1 - The equation for accuracy

Aside from accuracy/trueness, a notion of precision also needs to be considered. Precision is simply about the consistency or repeatability of the results. It is also called the reproducibility of a particular result or set of results. The strongest measure of consistency or variation between the results is the standard deviation. In light of this, a general definition of precision is the percentage of the mean of which the standard deviation is comprised. For our purposes, precision is the percentage of the mean volume of solution of which the standard deviation in the volume of solution is composed. More precisely, it is defined in the following equation:

$$Precision \%CV = 100 \times SD / Mean$$

Equation 2 - The equation for precision

As is the case with just about any experimental result, both trueness/accuracy and precision/reproducibility need to be maximized for an experiment to be successful. This is because accurate results that are not reproducible implies that an experiment can be carried out at some other time with the exact same conditions and not obtain the correct results when needed. By the same token, a result that is precise/reproducible that is not accurate/true is also wrong. As a result, despite there being a result that is replicable in an experimental setting as long as all other factors are assumed to remain constant, it simply does not come out to give the value that is intended.

In order to test for the accuracy and precision of the quantity of volumetric solution, there are four approaches: Photometric measurement, Fluorometric measurement, Gravimetric measurement, and combined Photometric and Gravimetric measurement.

## Photometric Measurement

Photometric measurement involves adding dye to the liquid and using a plate reader to perform the measurements. This approach is fairly accurate and is generally very good at gathering large amounts of data, though the dye poses a risk of biasing the results. It is also the only method that applies to liquid filled systems as it so happens that there is a dilution effect that is difficult to detect just from weighing samples. The gravimetric approach unfortunately does not protect against this and this dilution effect does affect results sometimes in the gravimetric approach.

On the other hand, Photometric measurements can be fairly difficult to setup initially. You need to identify the right wavelength, acquire a reader, and prepare your calculations. Finding the right dye can also introduce some difficulties. Lastly, Photometric measurement is not ideal for volumes under 10 microliters.

## Photometric Measurement Procedure

First, you'll need to obtain a dye with the optimal wavelength and concentration, which can be found by performing a spectral analysis on the color. If it is possible to determine

the optimal wavelength and concentration, use a scanning spectrometer. Otherwise, you can make an estimated guess on the right dye color. For example, since yellow dye transmits yellow light, you can then use it to absorb blue light (450nm).

Using a flat bottom plate is optimal. Reflection and refraction on any curved parts of the plate can skew measurements. While it is possible to accommodate for plates of different shapes, the reader software will have to adjust to that accordingly.

From there, try testing multiple concentrations to determine the saturation point of the reader. You should try to get a test concentration somewhere between 50% to 75%.

From there you just need to apply Beer's law. Beer's law is an equation that models how much light was absorbed due to passing through a particular chemical. It is described as follows:

$$A = \epsilon Lc$$

Equation 3 - Beer's law

Where A represents the Absorbance in the liquid, L represents the path length i.e. how deep the light has passed through the dye. In our case, that would simply be the depth of liquid in the well. Lastly, c is the concentration of the dye.

The Absorbance of the liquid is typically also described as the log of the ratio of the intensity of light transmitted to the intensity of light received. As a result, Absorbance is sometimes shown to be as follows:

$$A = \log(I_t / I_r) = -\log(I_r / I_t)$$

Equation 4 - The equation relating absorbance to transmitted and received light intensity

Where  $I_t$  is the intensity of the light transmitted and  $I_r$  is the intensity of the light received. Absorbance can be better understood by combining the quantities in the equations by which it is defined.

$$I_r = I_t e^{-\epsilon Lc}$$

Equation 5 - combining both equations for absorbance

Once you've applied Beer's Law and acquired the Absorbance of the dye, you can create a standard curve with the same dye solution used for unknown volumes. This cancels out



the constants. You can obtain that curve by collecting data below and above the volume you want and acquire a line of best fit.

## Fluorometric Measurement

Fluorometric measurement is similar to Photometric measurement since both use light, but use it differently. Photometric measurements simply subtract the amount of light transmitted from the light emitted from the source to calculate the amount of absorbed light. In Fluorometric measurements, the light is emitted into the dye at a specific wavelength. The dye then absorbs this light and emits it in a scattered fashion. This makes it good for well plates of any color. If anything, black well plates are generally preferred for this approach since they reduce the background interference due to its ability to absorb scattered light.

This approach carries with it many nice advantages. Alongside its efficiency and ability to collect data in large amounts, it seems to be the strongest candidate for measuring volumes in smaller quantities. Finding the right dye for this approach can be fairly difficult, however. In addition, acquiring a plate reader that is specifically capable of taking Fluorometric measurements will be considerably more difficult as they happen to be even more expensive than the plate readers that Photometric measurements use.

## Gravimetric Measurement

The Gravimetric approach is a simple approach that uses a simple fact: given the density and weight of a particular chemical, you can calculate the volume. In essence, it uses an analytical balance that is placed directly on the liquid handling robot in question. The advantage of this approach is that it's very simple and logical. A quick lookup shows that acquiring an analytical balance is much cheaper than acquiring a plate reader. The other advantage to this approach is that there is no need to apply any kind of dyes or additives for the measurement to take place, minimizing the probability of bias. The obvious disadvantage of this approach is that, sometimes, it's not always clear what the density of a particular chemical is. On top of that, it restricts the researchers ability to measure the volume efficiently as you will only be able to take one measurement at a time. It is also worth noting that Gravimetric measurement is also difficult at smaller volumes.

## Consultants, subcontractors, and suppliers

As previously mentioned, this is currently a work in progress. As it stands, for our project to hold water, we have a project budget of about \$250 for each team member for a total of \$1000. Since the Pin Transfer Tool alone can cost anywhere between \$250 and \$750 without even factoring for other costs, we plan on acquiring a sponsor that can guide us through the development process of the Liquid Handling Robot. Another reason is that it's too early to be able to tell the costs of the input and output stacking mechanisms as well as the workspace rail since those are fairly complex and not exactly available in the market for the constraints that we demand, at least not without being fairly expensive for our project. Furthermore, it very well might be the case that future iterations of the project

might be a necessary undertaking for this project as it is not yet clear what kinds of challenges we might face with our current model for the project. Lastly, it is important that we be able to cover the costs of any accidentals and contingencies that occur during the developmental or experimental parts of carrying out this project.

Originally we've made attempts to contact Dr. Bradley Jay Willenberg, renowned biomedical engineer, member of the American Mosquito Control Association, authored many peer reviewed articles, and is currently an active member in the UCF College of Medicine since 2014 but has not been able to get a response.

One of our members was able to reach out to a former acquaintance at his internship at MayoClinic to help with advice and funding and while he expressed his interest in helping us, it might take a while for him to be available as he is currently very busy with work currently taking place at MayoClinic.

In spite of this, all hope is not lost. We still plan on contacting other members from the UCF College of Medicine for consultancy and funding, especially within the Burnett College of Medicine circles. We plan on contacting Carlee Thomas, director of development at the Burnett College of Medicine as he seems to be supervising the support behind biomedical researchers and graduates.

Lastly, it might be the case that we might request the consultancy of one or more mechanical engineers. As it was previously stated, there is a fair bit of mechanical moving parts that are involved with the design and implementation of this project. It would be convenient to have someone who is knowledgeable about spring latching and locking mechanisms as we plan on using this to place plates into the stack and pop them out of the stack.

Also, if the need arises for it, we might also contact someone with experience in SolidWorks or AutoCAD to aid in the modeling design of the Liquid Handling Robot. Furthermore, we might need someone with experience in the kinds of materials that are to be used in 3D printing the parts used to design and build the Liquid Handling Robot. For this purpose, I plan on contacting Dr. Ricardo Zaurin, renowned undergraduate mechanical engineering professor and associate lecturer at UCF during his office hours. There is also the possibility of contacting other undergraduate mechanical engineering professors at UCF or undergraduates for when the occasion arises.

Overall, I think it is important that we know how to get the right kind of consultancy necessary for this project to succeed as it is unlikely that this project will undergo that many iterations. Up until this point, we know that we need consultancy or experience in the areas of mechanical engineering, CAD, AutoCAD, or Solidworks experience, and potentially help with finding, buying, and using the right materials to 3D print with to ensure the project's success.

# Hardware Testing

## OLED Testing

The OLED display is the only part that will display the user interface, so any issues with this once the build is finalized would be disastrous since the user would have no way of knowing how to get the Pin Transfer Tool to start. For this test, the microcontroller will be powered by a computer via USB and the microcontroller will be connected to the display via a breadboard. To test the OLED for any issues, all of the pixels on the display will be turned on. This will achieve two things if successful. First, it will show that the OLED can connect to the microcontroller without issues. Second, it will show any dead pixels that the display has.

## Keypad Testing

The keypad is what allows the user to interact with the Pin Transfer Tool. If the keypad does not work, the pin transfer process will never begin. For this test, the microcontroller will be powered by a computer via USB and the microcontroller will be connected to the keypad via a breadboard. To test the keypad for any issues, each key will be pressed and the corresponding symbol will be printed to the computer. This will test to make sure that the keypad properly connects to the microcontroller. It will also make sure that all of the keys properly output their designated symbol.

# Administrative Content

## Milestones

Milestone Timeline			
Milestone Number	Milestone Description	Start Date	End Date
<i>Senior Design I</i>			
1	Attempt to get a sponsor	1/11/2021	4/27/2021
2	Team Formation	1/11/2021	1/15/2021
3	Discuss Ideas / Project Selection	1/11/2021	1/15/2021
4	Bootcamp	1/21/2021	1/21/2021
5	Divide and Conquer V1	1/27/2021	1/29/2021
6	Decide between Gantry(3-axis) vs Conveyor belt operation	2/1/2020	4/27/2021
7	Select Pin Transfer Tool	2/1/2021	2/12/2021
8	Divide and Conquer V2	2/15/2021	4/27/2021
9	Design power supply(AC->DC Full Bridge Rectifier)	2/15/2021	4/27/2021
10	Design PCB	2/15/2021	4/27/2021
11	Select MCU	2/15/2021	4/27/2021
12	Write Pseudocode for MCU	2/15/2021	4/27/2021
13	60 Page Draft	2/15/2021	4/2/2021
14	100 Page Draft	4/2/2021	4/16/2021
15	Final Report	4/16/2021	4/27/2021
<i>Summer Break</i>			
16	Acquire Parts	5/3/2021	8/23/2021
17	Design enclosure for Electronics	5/3/2021	8/23/2021
18	Write and Implement Code for MCU	5/3/2021	8/23/2021
<i>Senior Design II</i>			
19	Assemble Project	TBD	TBD
20	Testing	TBD	TBD
21	Finalize PowerPoint presentation	TBD	TBD
22	Final Testing	TBD	TBD
23	Final Presentation	TBD	TBD

Figure 51: Milestone Timeline

Figure 51 was the initial milestone timeline created for this project and it is still the one that is being followed now. Work will begin on the project in the summer so that there is more time in

the fall in case any issues are encountered during testing and assembly. Over the summer, all necessary parts will be ordered. This includes any parts that need to be custom made, such as the PCB and project base. Ordering everything this early will give enough time for the parts to arrive by the start of the fall semester. Also, as soon as the parts are received, code can start to be written for them. Ideally, each part will have its code completely written by the time the fall semester starts so all that is needed during fall semester is the assembly of the project. However, if that does not happen, we should at least know how to control each part so that when we start to write the code, we can focus on getting the machine to work rather than trying to understand each individual part.

The Fall will be for project assembly and testing. All of the parts will be put together, the code will be finished if it is not already, and testing the efficiency of the project will begin. Hopefully there will not be any need for large changes, but there should be enough time to make any should there be a need. Also, there has been talk of adding additional functionality. If time permits, we might try to add additional features.

## Budget Analysis

Shown below in Table 22 is the breakdown of all the components purchased for this project.

Item	Quantity	Price
Pin Transfer Tool	1	Free
Microcontroller IC	1	\$ 13.68
Motor 1	1	\$ 27.99
Motor 2	3	\$ 17.99
Motor Driver	4	\$ 19.90
Rail 1	1	\$ 151.99
Rail 2	3	\$ 103.99
Linear Actuator	1	\$ 70
Fans	1	\$ 6.99
Solenoid Valve	1	\$ 14.99
Power Supply	1	\$ 31.50
Keypad	1	Free
Screen	1	\$ 25
PCB Fabrication	-	\$ 15
Misc.	-	\$ 100
	Total	\$ 902.68

Table 22 : Cost of Project

## Summary and conclusions

To wrap up our project, we will be designing a 3D gantry robot that will be taking well plates from a pair of stacks, one for the animal cells and another for the chemicals. These well plates will then be processed by the pin transfer tool. The pin transfer tool from there should begin the pin transfer operation by taking the chemicals from the chemical well plate and put them onto the cells. From there, the pin tool will proceed to the wash steps. These wash steps will be fairly generic and left up to the user. After the pin transfer operation is complete, the well plates will simply be passed over to the output stack where they will be stored and hopefully picked up by the researcher to see the results.

As previously stated, our mission with this project is to provide an intuitive and cheap solution to smaller labs for building a library of chemicals. You will find many professional, industry-sized, liquid handling robots that do this already, but many of those solutions are relatively expensive and are only affordable by huge labs. This generally makes it so where the smaller labs have to outsource their work entirely to the larger labs, which can introduce a fair bit of overhead. A liquid handling robot of this sort can vastly enhance the productivity of those intent on building libraries of chemicals for ambitious projects such as treating Type 1 Diabetes, developing anti-cancer drugs, etc.


Our design consists of an assortment of well-designed parts. We've considered a sizable variety of stacking mechanisms and have gone through several iterations of these ideas to see which one of them fits best. We've explored C++ and Python for our programming language and determined that Python is the better choice. We've determined to use Github for its rigorous source control, its solid issue tracking, as well as its robust Continuous Development/Continuous Integration platform. We've concluded that we will have to pay about \$902. This can be split over all four teammates, but we've also seriously considered looking into sponsors but to no avail. In spite of this, the search will continue as we will most certainly need the necessary funding to perform multiple iterations on the project. The prototype build will prioritize certain building aspects over others. For example, the stacking and wash steps will be left out until we approach the final iteration of the liquid handling robot. On top of that, we've studied a series of tried-and-true testing procedures for determining the quality of the pin tool that we have at hand. Namely, Photometric, Gravimetric, and Fluorometric testing procedures have been explored in great detail.


Should this project succeed with the provided constraints, it will be taken to the next level with new stretch goals. Namely, the addition of a refrigeration component to the liquid handling robot will allow the well plates to be properly preserved so that a researcher can schedule a pin transfer process to occur as opposed to attending the liquid handling robot while it performs the pin transfer process. Another stretch goal which was thoroughly considered was adding a barcode scanner to the pin transfer robot that would identify the well plates and add them to a database for proper bookkeeping.

# Appendix Updated

## Permissions

### Adafruit


 phillip torrone <pt@adafruit.com>  
To Brenden Morton


 We removed extra line breaks from this message.

all good, you have permission to use, please credit Adafruit

### ICCB-Longwood Screening Facility at Harvard Medical School, Boston MA

Re: Permission to use Images for Senior Design Report

 Smith, Jennifer <Jennifer\_Smith@hms.harvard.edu>  
To Brenden Morton

 You replied to this message on 4/1/2021 2:16 PM.

Hi Brenden,

Yes, you can use these images, as long as you indicate they are from: ICCB-Longwood Screening Facility at Harvard Medical School, Boston MA

Good luck with your automation project and completing the report.

Thanks, take care, Jen

---

Jennifer Smith, Ph.D.  
[jennifer\\_smith@hms.harvard.edu](mailto:jennifer_smith@hms.harvard.edu), 617.432.5735

Director, ICCB-Longwood Screening Facility, <https://iccb.med.harvard.edu>  
Lecturer, Department of Immunology  
Harvard Medical School, [250 Longwood Ave.](#), SGM 604, Boston, MA 02115



## OpenBuilds.com

Re: Use of images on your site in senior design report



Support@OpenBuilds.com  
To: Brenden Morton

If there are problems with how this message is displayed, click here to view it in a web browser.

Hi Brenden,

Thank you for contacting us. We are happy to help!

Sure you can use any of our photos from our Website. If you, can footnote us :)

We wish you luck on your senior project. It sounds really awesome. We would also like to know how you make out.

Feel free to reach out with any other questions and thank you for your support!

Thank you,



OpenBuilds Team

## V&P Scientific

RE: V&P Inquiry CRM:0017002



Joel Renick <joel.renick@vp-sci.com>  
To: Brenden Morton  
Cc: Natalie Knuth; pclelandhome@gmail.com

You replied to this message on 4/2/2021 1:56 PM.

Hi Brenden

Please go ahead and use the images and give credit to V&P in your report. Cheers!

Kind Regards,

### Joel Renick

Associate Director, Chemical Sciences



P: 858-455-0643

E: [joel.renick@vp-sci.com](mailto:joel.renick@vp-sci.com)

W: [www.vp-sci.com](http://www.vp-sci.com)

9823 Pacific Heights Blvd., Ste T

San Diego, CA 92121



# Pseudo Code for Scanning Procedure

```
Set all rows to high
  For row in rows:
    row = low
    for col in columns:
      If col == low
        Key @ (row,col) is pressed
      Else,
        Key is not pressed
```

## Citations

### 1. [RS-35-48]

TRC Electronics, INC. "RS-35-48 - MEAN WELL - TRC Electronics." *TRC Electronics, Inc.*, 2021, [www.trcelectronics.com/View/Mean-Well/RS-35-48.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UhJ6PWWf4HFqnnCuvbTS0npimrGwyHqolYSMRQRvOy8AxAPQ4cUBAoaAkhLEALw\\_wcB](http://www.trcelectronics.com/View/Mean-Well/RS-35-48.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UhJ6PWWf4HFqnnCuvbTS0npimrGwyHqolYSMRQRvOy8AxAPQ4cUBAoaAkhLEALw_wcB).

### 2. [LRS-150-48]

TRC Electronics, INC. "LRS-150-48 - MEAN WELL - TRC Electronics." *TRC Electronics, Inc.*, 2021, [www.trcelectronics.com/View/Mean-Well/LRS-150-48.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UjB3MhRxRbPwlFEUFiP98zeHGUw1Omm8icGDCnOM43YhtF6D57Q\\_FkaAgphEALw\\_wcB](http://www.trcelectronics.com/View/Mean-Well/LRS-150-48.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UjB3MhRxRbPwlFEUFiP98zeHGUw1Omm8icGDCnOM43YhtF6D57Q_FkaAgphEALw_wcB).

### 3. [LRS-350-24]

TRC Electronics, INC. "LRS-350-24 - MEAN WELL - TRC Electronics." *TRC Electronics, Inc.*, 2021, [www.trcelectronics.com/View/Mean-Well/LRS-350-24.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UjcVzCFY-cCogQ018N7kaNE6b3YRzo\\_4vEa7A\\_J6EOe4H70ZzcvUiUaArlNEALw\\_wcB](http://www.trcelectronics.com/View/Mean-Well/LRS-350-24.shtml?gclid=Cj0KCQjw6-SDBhCMARIsAGbl7UjcVzCFY-cCogQ018N7kaNE6b3YRzo_4vEa7A_J6EOe4H70ZzcvUiUaArlNEALw_wcB).

### 4. [MSP430FR6989]

Texas Instruments. "MSP430FR6989." *MSP430FR6989 Data Sheet, Product Information and Support | TI.com*, Texas Instruments, 2021, [www.ti.com/product/MSP430FR6989#product-details###params](http://www.ti.com/product/MSP430FR6989#product-details###params).

### 5. [MSP430G2452]

Texas Instruments. "MSP430G2452." *MSP430G2452 Data Sheet, Product Information and Support | TI.com*, Texas Instruments, 2021, [www.ti.com/product/MSP430G2452#product-details###params](http://www.ti.com/product/MSP430G2452#product-details###params).

## 6. [ATMEGA 2560]

Microchip. "ATmega2560." *ATmega2560 - 8-Bit AVR Microcontrollers*, Microchip, 2n.d., [www.microchip.com/wwwproducts/en/ATmega2560](http://www.microchip.com/wwwproducts/en/ATmega2560).

## 7. [ATMEGA-328P]

Arduino. "Arduino Uno Rev3." *Arduino Uno Rev3 | Arduino Official Store*, 2021, [store.arduino.cc/usa/arduino-uno-rev3?gclid=Cj0KCQjwvYSEBhDjARIsAJMn0lie2Jx44o5NK63EhaezW-p1zQGn7syRmCmpssaKO0oBtlzKq\\_q4qHEaAu2UEALw\\_wcB](https://store.arduino.cc/usa/arduino-uno-rev3?gclid=Cj0KCQjwvYSEBhDjARIsAJMn0lie2Jx44o5NK63EhaezW-p1zQGn7syRmCmpssaKO0oBtlzKq_q4qHEaAu2UEALw_wcB).

## 8. [DM542T ]

"User's Manual For DM542T Full Digital Stepper Drive." *Stepper Online*, StepperOnline, 2017, [www.omc-stepperonline.com/](http://www.omc-stepperonline.com/).

## 9. [TB6600]

"TB6600\_Stepper\_Motor\_Driver\_SKU\_\_DRI0043." *DFRobot*, DFRobot, 2021, [wiki.dfrobot.com/TB6600\\_Stepper\\_Motor\\_Driver\\_SKU\\_\\_DRI0043](https://wiki.dfrobot.com/TB6600_Stepper_Motor_Driver_SKU__DRI0043).

## 10. [TB6560]

de Bakker, Benne. "TB6560 Stepper Motor Driver with Arduino Tutorial (2 Examples)." *Makerguides.com*, 18 Aug. 2020, [www.makerguides.com/tb6560-stepper-motor-driver-arduino-tutorial/](http://www.makerguides.com/tb6560-stepper-motor-driver-arduino-tutorial/).

## 11. [DRV8825]

"Pololu - DRV8825 Stepper Motor Driver Carrier, High Current." *Pololu Robotics & Electronics*, 2021, [www.pololu.com/product/2133](http://www.pololu.com/product/2133).

## 12. [A4988]

"Control Stepper Motor with A4988 Driver Module & Arduino." *Last Minute Engineers*, 2021, [lastminuteengineers.com/](http://lastminuteengineers.com/).

## 13. [L298N]

Bakker, Benne de. "Stepper Motor with L298N and Arduino Tutorial (4 Examples)." *Makerguides.com*, 18 Aug. 2020, [www.makerguides.com/l298n-stepper-motor-arduino-tutorial/](http://www.makerguides.com/l298n-stepper-motor-arduino-tutorial/).

#### **14. [NEMA 23 HT]**

"NEMA 23 Stepper Motor - High Torque Series." *OpenBuilds Part Store*, 2021, [openbuildspartstore.com/nema-23-stepper-motor-high-torque-series/](http://openbuildspartstore.com/nema-23-stepper-motor-high-torque-series/).

#### **15. [NEMA 23]**

"NEMA 23 Stepper Motor." *OpenBuilds Part Store*, 2021, [openbuildspartstore.com/nema-23-stepper-motor/](http://openbuildspartstore.com/nema-23-stepper-motor/).

#### **16. [NEMA 17]**

"NEMA 17 Stepper Motor." *OpenBuilds Part Store*, 2021, [openbuildspartstore.com/nema-17-stepper-motor/](http://openbuildspartstore.com/nema-17-stepper-motor/).

#### **17. [Servo Vs. Stepper]**

Lackey, Bill. "What's the Difference Between Servo and Stepper Motors?" *Machine Design*, 24 June 2018, [www.machinedesign.com/mechanical-motion-systems/article/21836868/whats-the-difference-between-servo-and-stepper-motors](http://www.machinedesign.com/mechanical-motion-systems/article/21836868/whats-the-difference-between-servo-and-stepper-motors).

#### **18. [DC MOTOR]**

"DC Motor." *NationalMagLab*, 17 June 2019, [nationalmaglab.org/education/magnet-academy/watch-play/interactive/dc-motor](http://nationalmaglab.org/education/magnet-academy/watch-play/interactive/dc-motor).

#### **19. [DC MOTOR WIKI]**

"DC Motor." *Wikipedia*, Wikimedia Foundation, 18 Apr. 2021, [en.wikipedia.org/wiki/DC\\_motor](http://en.wikipedia.org/wiki/DC_motor).

#### **20. [Bluetooth]**

"Bluetooth." *Wikipedia*, Wikimedia Foundation, 14 Apr. 2021, [en.wikipedia.org/wiki/Bluetooth](http://en.wikipedia.org/wiki/Bluetooth).

#### **21. [Wi-Fi]**

"Wi-Fi." *Wikipedia*, Wikimedia Foundation, 22 Apr. 2021, [en.wikipedia.org/wiki/Wi-Fi](http://en.wikipedia.org/wiki/Wi-Fi).

## 22. [ARDUINO REF]

“Language Reference.” *Arduino Reference - Arduino Reference*, 2021, [www.arduino.cc/reference/en/](http://www.arduino.cc/reference/en/).

## 23. [STEPPER.H REF]

“Stepper.” *Arduino*, 2021, [www.arduino.cc/en/reference/stepper](http://www.arduino.cc/en/reference/stepper).

## 24. [ACCEL STEPPER]

“AccelStepper Class Reference.” *AccelStepper: AccelStepper Class Reference*, 2021, [www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html](http://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html).

## 25. [Circuit Python]

“CircuitPython.” *Libraries*, 2021, [circuitpython.org/libraries](http://circuitpython.org/libraries).

## 26. [KEYPAD FUNCTIONALITY]

Pattabiraman, Krishna. “How to Set Up a Keypad on an Arduino.” *Circuit Basics*, 9 Aug. 2020, [www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/](http://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/).

## 27. [KEYPAD 4x4]

Industries, Adafruit. “4x4 Matrix Keypad.” *Adafruit Industries Blog RSS*, 2021, [www.adafruit.com/product/3844#technical-details](http://www.adafruit.com/product/3844#technical-details).

## 28. [KEYPAD 3x4]

Industries, Adafruit. “3x4 Matrix Keypad.” *Adafruit*, 2021, [www.adafruit.com/product/3845](http://www.adafruit.com/product/3845).

## 29. [KEYPAD membrane]

Parallax. “4x4 Matrix Membrane Keypad.” Parallax, 16 Dec. 2011.

## 30. [KEYPAD Library]

“Keypad - Arduino Reference.” *Arduino*, 2021, [www.arduino.cc/reference/en/libraries/keypad/](http://www.arduino.cc/reference/en/libraries/keypad/).

## 31. [LCD vs. TFT]

“Difference between TFT and LCD.” *Microtips Technology Blog*, 5 May 2014, [www.microtipsusa.com/blog/difference-tft-](http://www.microtipsusa.com/blog/difference-tft-)



“Thin-Film-Transistor Liquid-Crystal Display.” *Wikipedia*, Wikimedia Foundation, 20 Mar. 2021, [en.wikipedia.org/wiki/Thin-film-transistor\\_liquid-crystal\\_display](https://en.wikipedia.org/wiki/Thin-film-transistor_liquid-crystal_display).

#### **41. [16U2 Cost]**

ATMEL. “8-Bit AVR Microcontroller with 16K Bytes of ISP Flash and USB Controller.” 2021.

#### **42. [FTDI Functionality]**

Ada, Lady. “FTDI vs. AVR Programmer.” *Adafruit Learning System*, [learn.adafruit.com/ftdi-friend/ftdi-friend-vs-avr-programmer](https://learn.adafruit.com/ftdi-friend/ftdi-friend-vs-avr-programmer).

#### **43. [ARDUINO ISP]**

Team, The Arduino. “Arduino as ISP and Arduino Bootloaders.” *Arduino*, 5 Feb. 2018, [www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP#the-bootloader](https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP#the-bootloader).

#### **44. [RAILS]**

PBC Linear. “Belt Drive or Lead Screw?” *Belt Drive or Lead Screw? The Answer Is in the Application.*, 2021, [www.pbclinear.com/Blog/2020/February/Lead-Screw-or-Belt-Drives](https://www.pbclinear.com/Blog/2020/February/Lead-Screw-or-Belt-Drives).

#### **45. [RAILS EXPLAINED]**

Casillo, David. “Belt-Driven Versus Ball Screw Actuator: Which Is the Best Choice for Your Application?” *Isotech, Inc, Product News*, 21 Nov. 2019, [www.isotechinc.com/belt-driven-versus-ball-screw-actuators/](https://www.isotechinc.com/belt-driven-versus-ball-screw-actuators/).

#### **46. [BELT RAIL]**

OpenBuilds. “V-Slot® NEMA 23 Linear Actuator (Belt Driven) Bundle.” *OpenBuilds Part Store*, 2021, [openbuildspartstore.com/v-slot-nema-23-linear-actuator-belt-driven/](https://openbuildspartstore.com/v-slot-nema-23-linear-actuator-belt-driven/).

#### **47. [SCREW RAIL]**

OpenBuilds. “V-Slot® NEMA 17 Linear Actuator Bundle (Lead Screw).” *OpenBuilds Part Store*, 2021, [openbuildspartstore.com/v-slot-nema-17-linear-actuator-bundle-lead-screw/](https://openbuildspartstore.com/v-slot-nema-17-linear-actuator-bundle-lead-screw/).

#### **48. [PYDUINO]**

Rodriguez, Alejandro Guirao. “Pyduino.” *GitHub*, 10 Nov. 2014, [github.com/lekum/pyduino](https://github.com/lekum/pyduino).

#### **49. [DATASHEET 16U2]**

ATMEL. "8-Bit Microcontroller with 8/16/32K Bytes of ISP Flash and USB Controller." 2010.

#### **50. [C++ CORE]**

Stroustrup, Bjarne, and Herb Sutter. *C++ Core Guidelines*, [isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#main](https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#main).

#### **51. [NEWARK FAN]**

"28-1740 - DC Axial Fan, 12 V, Square, 80 Mm, 25 Mm, Ball Bearing, 25.98 CFM." *Newark*, Premier Farnell Limited, [www.newark.com/mcm/28-1740/supply-voltage-vdc-12v/dp/39C0485?COM=ref\\_hackster](http://www.newark.com/mcm/28-1740/supply-voltage-vdc-12v/dp/39C0485?COM=ref_hackster)

#### **52. [KING FAN]**

"Case Fan CF-014LB." *KWI Kingwin*, KWI Technology Inc, 28 Nov. 2016, [www.kingwin.com/case-fan-cf-014lb/#tab-1443047560105-3-7](http://www.kingwin.com/case-fan-cf-014lb/#tab-1443047560105-3-7).

#### **53. [WASTE]**

*Defining Hazardous Waste: Listed, Characteristic and Mixed Radiological Wastes*. [www.epa.gov/hw/defining-hazardous-waste-listed-characteristic-and-mixed-radiological-wastes](http://www.epa.gov/hw/defining-hazardous-waste-listed-characteristic-and-mixed-radiological-wastes).

#### **54. [OLED LIBRARY]**

"Guide for I2C OLED Display with Arduino." *Random Nerd Tutorials*, [RandomNerdTutorials.com](http://RandomNerdTutorials.com), [randomnerdtutorials.com/guide-for-oled-display-with-arduino/](http://randomnerdtutorials.com/guide-for-oled-display-with-arduino/).

#### **55. [L12-R]**

"L12-R Small Linear Servo for RC and Arduino: Actuonix." *Www.actuonix.com*, Actuonix Motion Devices Inc., [www.actuonix.com/L12-R-Linear-Servo-For-Radio-Control-p/l12-r.htm](http://www.actuonix.com/L12-R-Linear-Servo-For-Radio-Control-p/l12-r.htm).

#### **56. [PA-07]**

"Micro Linear Actuator." *Progressive Automations*, Progressive Automations Inc., [www.progressiveautomations.com/products/micro-linear-actuator?variant=182773447721](http://www.progressiveautomations.com/products/micro-linear-actuator?variant=182773447721).



## 57. [ADAFRUIT FAN]

“Miniature 5V Cooling Fan with Molex PicoBlade Connector.” *Adafruit Industries Blog RSS*, [www.adafruit.com/product/4468#technical-details](http://www.adafruit.com/product/4468#technical-details).

## 58. [PART 172]

“PART 172—HAZARDOUS MATERIALS TABLE, SPECIAL PROVISIONS, HAZARDOUS MATERIALS COMMUNICATIONS, EMERGENCY RESPONSE INFORMATION, TRAINING REQUIREMENTS, AND SECURITY PLANS.” *Electronic Code of Federal Regulations (ECFR)*, [www.ecfr.gov/cgi-bin/text-idx?SID=2bdda520863adeaf465a38c82bc9c7bb&mc=true&node=pt49.2.172&rgn=div5](http://www.ecfr.gov/cgi-bin/text-idx?SID=2bdda520863adeaf465a38c82bc9c7bb&mc=true&node=pt49.2.172&rgn=div5).

## 59. [PART 173]

“PART 173—SHIPPERS—GENERAL REQUIREMENTS FOR SHIPMENTS AND PACKAGINGS.” *Electronic Code of Federal Regulations (ECFR)*, [www.ecfr.gov/cgi-bin/retrieveECFR?gp=&SID=e323ba3c477273a4596ea206717b1ca0&mc=true&node=PART&n=pt49.2.173](http://www.ecfr.gov/cgi-bin/retrieveECFR?gp=&SID=e323ba3c477273a4596ea206717b1ca0&mc=true&node=PART&n=pt49.2.173).

## 60. [PART 261]

“PART 261—IDENTIFICATION AND LISTING OF HAZARDOUS WASTE.” *Electronic Code of Federal Regulations (ECFR)*, [www.ecfr.gov/cgi-bin/text-idx?SID=4b5157fcf8e1a99ff9fea351b21613d5&mc=true&node=pt40.28.261&rgn=div5%5C](http://www.ecfr.gov/cgi-bin/text-idx?SID=4b5157fcf8e1a99ff9fea351b21613d5&mc=true&node=pt40.28.261&rgn=div5%5C).

## 61. [PART 262]

“PART 262—STANDARDS APPLICABLE TO GENERATORS OF HAZARDOUS WASTE.” *Electronic Code of Federal Regulations (ECFR)*, [www.ecfr.gov/cgi-bin/text-idx?SID=e395f2f3b1732220c7b68bb075c93236&mc=true&node=pt40.28.262&rgn=div5](http://www.ecfr.gov/cgi-bin/text-idx?SID=e395f2f3b1732220c7b68bb075c93236&mc=true&node=pt40.28.262&rgn=div5).

## 62. [PEP 8]

“PEP 8 -- Style Guide for Python Code.” *Python.org*, [www.python.org/dev/peps/pep-0008/#a-foolish-consistency-is-the-hobgoblin-of-little-minds](http://www.python.org/dev/peps/pep-0008/#a-foolish-consistency-is-the-hobgoblin-of-little-minds).

## 63. [PYSERIAL]

“Pyserial 3.5.” *PyPI*, Python Software Foundation, [pypi.org/project/pyserial/](https://pypi.org/project/pyserial/).

#### 64. [SOLENOID]

“Tailonz Pneumatic 1 Inch NPT 12V/24V/110V/220V Brass Electric Solenoid Valve 2W-250-25 Normally Closed Water, Air, Diesel .” *Amazon*, Amazon.com, Inc, [www.amazon.com/Pneumatic-Electric-Solenoid-2W025-08-Normally/dp/B08RHLXWXQ?th=1](https://www.amazon.com/Pneumatic-Electric-Solenoid-2W025-08-Normally/dp/B08RHLXWXQ?th=1).

#### 65. [ATMEGA2560 DATASHEET]

“8-Bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash.” *Microchip*, 2014, [ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf).

#### 66. [ATMEGA16U2 DATASHEET]

“8-Bit Microcontroller with 8/16/32K Bytes of ISP Flash and USB Controller.” *Microchip*, 2010, [ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf).

#### 67. [HAMILTON]

“Methods to Measure Transferred Liquid Volumes.” *How To Measure Transferred Liquid Volume in Robotic Handlers*, 2021, [www.hamiltoncompany.com/automated-liquid-handling/everything-you-need-to-know-about-liquid-handling/measure-liquid-transfers#combined-photometric-and-gravimetric](http://www.hamiltoncompany.com/automated-liquid-handling/everything-you-need-to-know-about-liquid-handling/measure-liquid-transfers#combined-photometric-and-gravimetric).

#### 68. [LCD VS OLED]

Thornton, Scott. “Display Options for MCUs: LCD, LED, and OLED.” *Microcontroller Tips*, WTW Media LLC, 6 June 2017, [www.microcontrollertips.com/display-options-mcus-lcd-led-oled/](http://www.microcontrollertips.com/display-options-mcus-lcd-led-oled/).