Boston University
College of Engineering
Department of Electrical and Computer Engineering

High-Throughput Digital Biodetection

Master's Project End of Semester Report

Harrison Cao & Yang Zeng Lu

Advisor: Professor M. Selim Ünlü

December 15th, 2023

Table of Contents

I. Introduction

Considering the growing threat posed by infectious diseases and viruses, particularly in light of the post-Covid-19 era, there is an urgent and pressing need for cost-effective multiplexed, label-free, high-throughput virus detection methods for small single samples. One emerging method for single molecule detection involves harnessing optics through Interferometric Reflectance Imaging Sensors (IRIS chips) to precisely gauge the size and quantity of particles within individual samples.

The objective of this project is to design and prototype an automated system capable of seamlessly transporting and centering a standardized microtiter plate's wells in front of a stationary IRIS optical system. This system will facilitate high-throughput digital measurements of distinct binding events within each well of a microtiter plate during the sample incubation process. This innovative approach unlocks unprecedented capabilities in high-throughput digital bio-sensing, offering levels of sensitivity and resolution that surpass traditional labeling methods.

II. Goals Outlined

During the initial project proposal, the approach and subsequent goals were divided into three phases: research, prototyping, and testing. In the research phase, various methods for positioning 96-well microtiter plates while minimizing vibration were studied. Different techniques for inserting IRIS chips into the 96-well microtiter plates were also explored. The concepts derived from this research were then used to create a 3D model of the design using SolidWorks. During the prototyping phase, we optimized the CAD model based on feedback received. This optimized design was then constructed as a proof of concept. Finally, in the testing phase, we focused on the reliability testing of the prototype assembly. This involved conducting repeatability tests for positioning the microtiter plates along two different axes and evaluating the consistency of our dipping mechanism.

III. Proposed Design

The proposed design for this project involves two systems integrated together. These systems include a XY positioning system and the dipping mechanism.

The objective of the XY positioning system is to move a 96-well microtiter plate with high precision and accuracy from well to well. Specifically, the system should position a designated well, as selected by the user, to a focal point in a 2D space. This positioning enables the IRIS optical system to focus on the selected well.

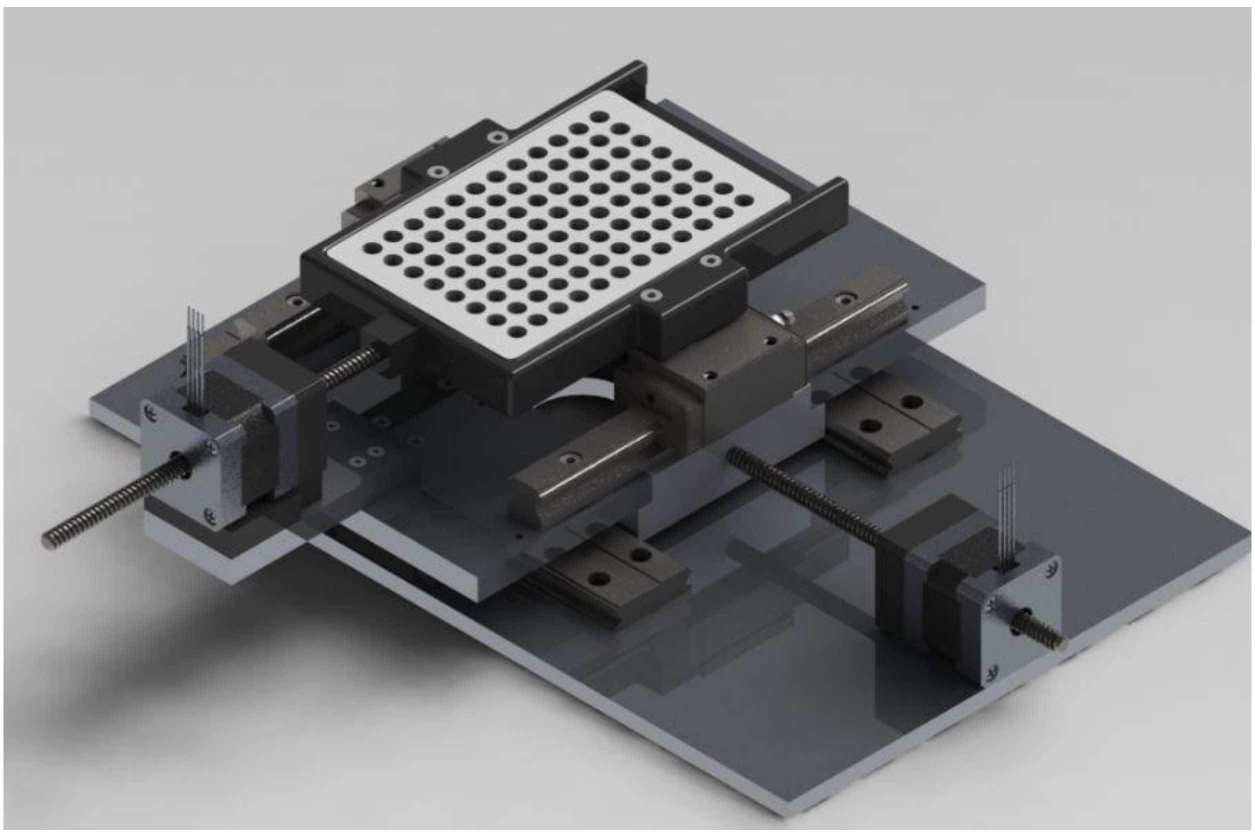

Figure 1: 3D Render of XY Positioning System in SolidWorks

The proposed design for positioning the microtiter plate uses two pairs of rails and two non-capacitive stepper motors. These motors will position the microtiter plates by pushing them along two perpendicular axes, with each axis having its own set of rails. By pushing the well plate only from its sides, it is ensured that the bottom of the wells remains unobstructed. This design allows the IRIS optics system to capture the individual wells of the microtiter plate as the positioning system moves the plate.

Additionally, an Arduino, alongside a stepper motor driver, will be used to control the non-capacitive stepper motors. The wiring diagram for this setup is shown in figure 2. The pseudo-code for driving the stepper motor can be found in the appendix.

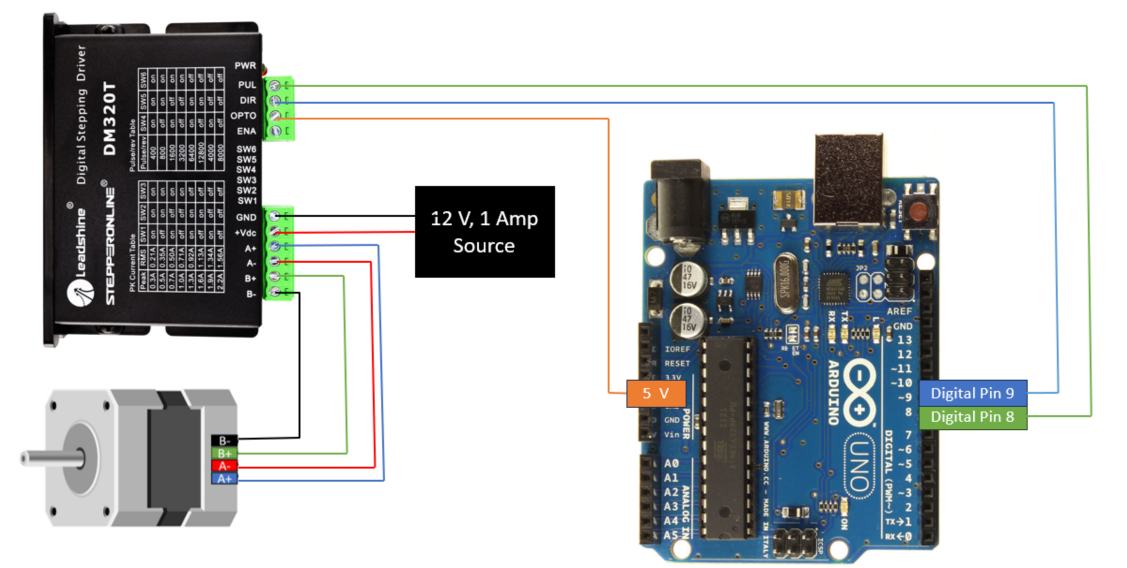

Figure 2: Wiring diagram for Arduino, stepper motors, and stepper motor drivers

To facilitate the insertion and cleaning of IRIS chips in a microtiter plate for the IRIS optical system, a specialized dipping mechanism needed to be designed. This mechanism ensures the IRIS chips are held securely and make direct contact with the base of the microtiter plate, without causing damage to either component. The IRIS chip must be positioned no more than 0.5 mm from the inside glass surface of the well plate for the optical system to function effectively. Additionally, to prevent cross-contamination between samples, a cleaning method for the IRIS chips after each sample is necessary.

Addressing these requirements, the proposed solution involves a resin-printed part called the "buckle," to which the IRIS chip will be friction-fitted to. This buckle will then be attached to a 3D-printed holder alongside eight other buckles, each equipped with springs. These springs provide a 2 mm cushion when the IRIS chip contacts the microtiter plate, protecting the glass at the bottom of the plate from damage. Ultimately, the entire assembly will then be connected to a linear actuator with the same wiring diagram as what is shown in figure 2, facilitating the insertion and retraction of the IRIS chips.

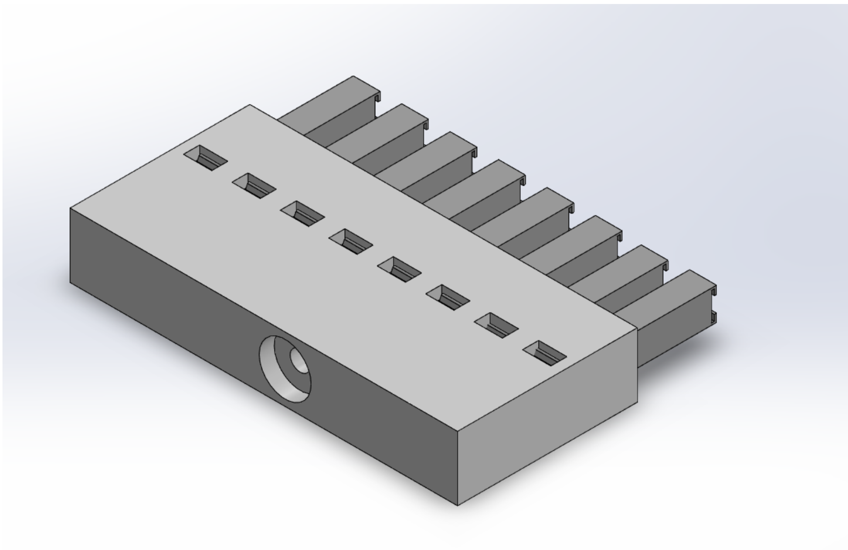

Figure 3: CAD Design of Buckle Holder with 8 Buckles Attached

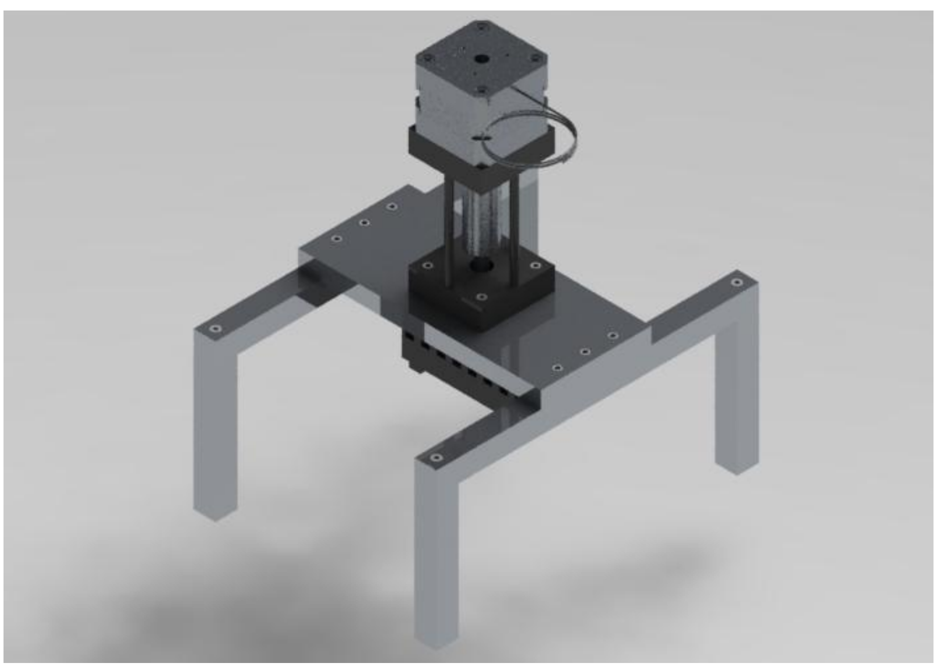

Figure 4: 3D Render of Dipping Mechanism in Solidworks

For cleaning the IRIS chips, it's proposed that half of the wells in the microtiter plate be designated for cleaning, while the other half are used for sampling. This arrangement allows the device to sequentially sample eight samples in a column, clean itself, and then proceed to resample.

Ultimately, our entire system is shown below in figure 5. A bill of materials can be found in the Appendix

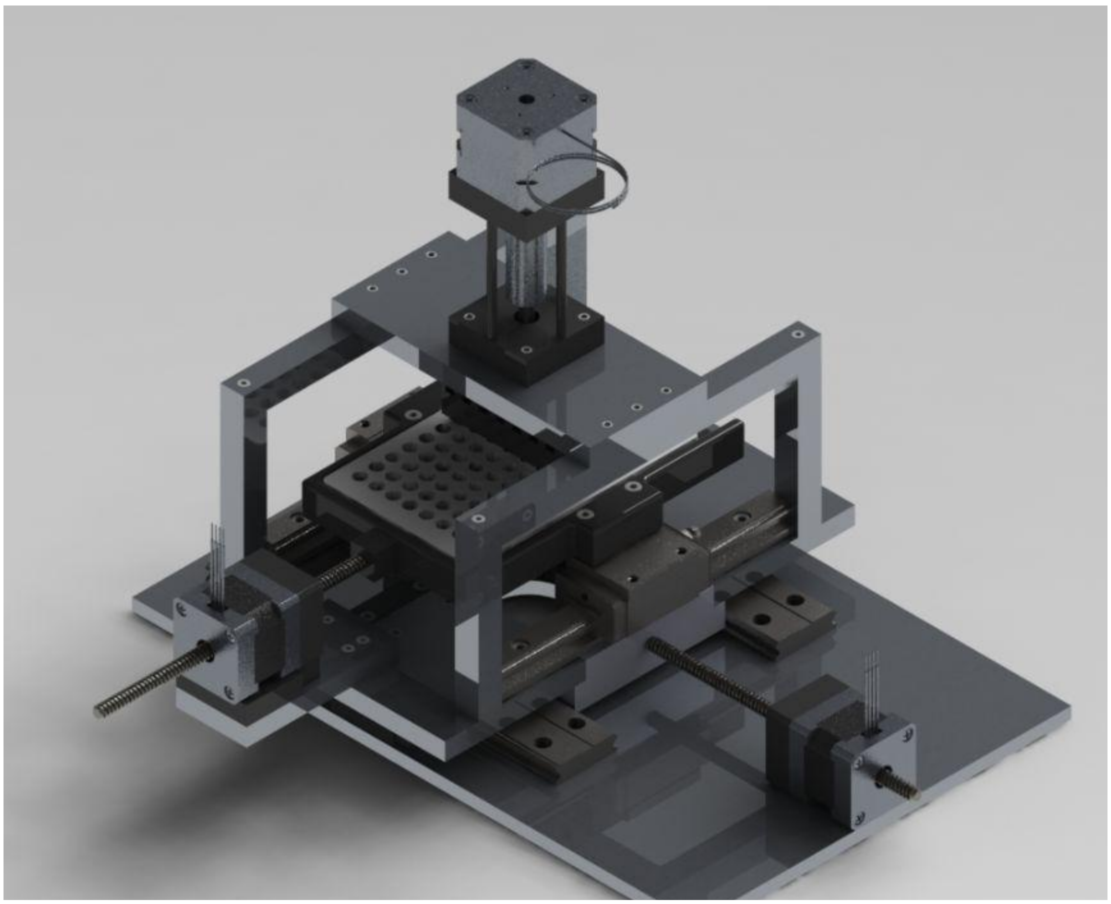

Figure 5: 3D Render of the XY Positioning and Dipping System in SolidWorks

IV. Testing & Results

To evaluate the precision and accuracy of the XY positioning system, each axis was assessed independently. The x-axis, controlled by the stepper motor pushing on the shorter edge of the microtiter plate, determines the positioning of rows, labeled A-F. Conversely, when the motor pushes along the longer edge or the y-axis, it positions the columns of the microtiter plate, each labeled with a number labeled 1-12.

For specific cell positioning tests on the microtiter plate, the X and Y axes were individually examined using a consistent method. Regarding the X-axis, an arbitrary zero point was established to align row A with the hole for the optical system. After setting this point, the distance traveled between rows was measured ten times. The process involved commanding the stepper motor to move sequentially from row A to H. Upon reaching row H, the plate returned to A for repeated measurements. These tests were conducted ten times, and the resulting data was used to calculate the standard deviation, reflecting the precision of the X-axis positioning system. For accuracy assessment, the average deviation from each row was determined. This value, along with the known 9 mm ground truth value, was used to calculate the average percentage error. Please refer to table 1 below for the values. Overall, there is less than a 0.1 mm standard deviation and less than 2% error for moving the stepper motor.

Rows	A	B	C	D	E	F	G	H
Average (mm)	19.51	28.51	37.53	46.447	55.36	64.32	73.17	82.14
Standard Deviation (mm)	0.077	0.044	0.040	0.047	0.053	0.078	0.088	0.079
Average Percent Error	-	0.02	0.17	0.87	0.94	0.47	1.6	0.40

Table 1: X-axis stepper motor data, showing the mean, standard deviation, and percent error averaged across 10 iterations.

The Y-axis positioning was evaluated in a very similar way to the X-axis. The only difference is that the Y-axis stepper motor is numbered from 1-12 rather than labeled A-H. The data for the Y-axis is shown in table 2.

Col	1	2	3	4	5	6	7	8	9	10	11	12
AVG (mm)	12.51	21.36	30.26	39.28	48.19	57.08	65.97	74.96	83.84	92.82	101.70	110.59
STD (mm)	0.005	0.004	0.028	0.031	0.010	0.011	0.008	0.009	0.025	0.011	0.012	0.084
% Error	-	1.6	1.1	0.19	1.0	1.2	1.1	0.17	1.3	0.18	1.3	1.2

Table 2: Y-axis stepper motor data, showing the mean, standard deviation, and percent error averaged across 10 iterations.

In terms of the dipping mechanism used for inserting iris chips, measuring the tolerances of each buckle is extremely difficult. Consequently, we conducted a pass-fail test to validate the consistency of the mechanism. The design successfully passed 10 iterations of repeated dipping across columns 1 to 12 without any issues. However, it's important to note that the dipping mechanism requires significant fine-tuning before it can achieve consistent performance. This need arises from the mounting mechanism's design, which connects the buckle housing to the stepper motor. Currently, there's free rotational motion, as nothing is preventing the buckle holder from rotating. This issue, which will be discussed further in the conclusion section, will need to be improved.

V. Conclusion

All the goals outlined in our initial proposal have been largely achieved. During the research phase, we iterated multiple times and finalized a design for the XY positioning and IRIS chip dipping mechanism. This led to the development of a 3D model defining the prototype. In the prototyping phase, the CAD model was brought to life through machining and 3D printing. Subsequently, in the testing phase, we conducted several reliability tests on the prototype. The primary shortfall, however, lies in integrating the prototype with the IRIS optical system. Due to our limited experience in machining, this process was more time-consuming than anticipated, delaying most testing and adjustments until recently.

For improvement, several aspects need attention. Currently, the electronics are operated using three Arduinos to simplify wiring and isolate testing conditions. Ideally, this should be streamlined to a single controller managing all three stepper motors, reducing costs and potential failure points. Additionally, the placement of the electronics was not considered in the design phase. They are currently attached to the prototype with adhesive. Incorporating additional tapped holes on the base plate would allow the electronics to be securely bolted down and covered, minimizing hazards. Moreover, the dipping mechanism's positioning relies on manual adjustment by the user and has nothing to prevent it from rotating. Designing a guide would ensure correct placement without user adjustments and prevent free rotation. Finally, the machined aluminum parts could benefit from remanufacturing with tighter tolerances and improved surface finishes for enhanced quality.

As this project comes to an end, we'd like to reflect on some key takeaways and lessons learned. First and foremost, we learned the importance of not underestimating any aspect of a project. Initially, we believed this project would be straightforward and moderately challenging. However, as we delved deeper, we quickly realized the complexity was far greater than anticipated. This led to certain goals becoming more time-consuming and demanding than we had expected. Consequently, a significant lesson for us was the need to avoid overpromising. As we progressed, it became clear that what we had thought feasible within our timeline required much more effort and resources. Despite successfully achieving most of our objectives, we found ourselves in a constant race against time. This experience has taught us the value of realistic planning and setting achievable milestones. It has also highlighted the importance of flexibility and adaptability in managing complex projects, especially when faced with unexpected challenges. Looking back, these insights have not only enhanced our technical skills but also enriched our understanding of effective project management.

VI. Appendix

Bill of Materials

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	9057K16	Base Plate Everything is mounted to, Tight-Tolerance Multipurpose 6061 Aluminum with Certificate	1
2	14N21xx_1.STEP	The NEMA 14 Stepper Motors used in the XY positioning system	2
3	Stepper_motor_hold er	3D Printed Part Responsible for holding the X Axis Stepper Motor and Base Plate together when screwed in	1
4	X_Carriage_Adapter _Plate_one_carriage	Adapter Plate for the X Direction Carriages, nearly everything is mounted onto here	2
5	6382K111	33 mm Wide x 190 mm Long Guide Rail for Extra-Wide Ball Bearing Carriage	2
6	6382K21_Extra-Wide Ball Bearing Carriage	Extra-Wide Ball Bearing Carriage	2
7	92855A408	18-8 Stainless Steel Low-Profile Socket Head Screws	3
8	Dipping_support	The short support legs of the Dipping Mechanism	4

9	Dipping_base	The Long supports the Dipping Adapter Plate sits on	2
10	92855A316	18-8 Stainless Steel Low-Profile Socket Head Screws	27
11	92855A410	18-8 Stainless Steel Low-Profile Socket Head Screws	8
12	91290A164	Alloy Steel Socket Head Screw	4
13	microwellPlateHolderFinalRev (1)	Holder for the 96 Well Microtiter Plate	1
14	6709K301	15 mm Wide x 160 mm Long Guide Rail for Ball Bearing Carriage	2
15	6709K12_Ball Bearing Carriage	Ball Bearing Carriage	2
16	microtierplatetruedim	Model of the 96 Well Microtiter Plate	1
17	90666A117	Super-Corrosion-Resistant 316 Stainless Steel Socket Head Screw	4
18	X_Adapter_Bar_Push	Linch Pin Piece, responsible for connecting both X Adapter Plates	1
19	Y_Stepper_Block	3D Printed Block that Y Stepper Motor is placed on	1
20	Stepper_motor_holder_Y_axis	3D Printed Part Responsible for holding the Y Adapter and Stepper motor together when screwed in	1

21	Y_stepper_motor_adapter_plate	Adapter Plate for the Y-Axis Stepper Motor	1
22	4290N12	Compact Stepper Motor Actuator Used in Dipping	1
23	actuatorMount		1
24	92196A110	18-8 Stainless Steel Socket Head Screw	4
25	dipping_adapter_plate	Adapter Plate for the Dipping Mechanism	1
26	92196A023	18-8 Stainless Steel Socket Head Screw	10
27	Buckle_Holder_8_dip	Holder for the "Buckles"	1
28	buckle2mm_7mm_1_1mm_hang	"Buckles", holder for the IRIS Chips	8
29	Arudino UNO R3	Microcontrollers responsible controlling the logic of the stepper motors	3
30	StepperOnline DM 320T	An interface that connects the stepper motor with the Arduino	3

Arduino Pseudocode for Positioning along X and Y Axis:

Import the Stepper.h library

Initialize constants:
- stepsPerRevolution: [Specify the number of steps per revolution for your motor]
- stepDistance: [Specify the distance covered per step, in millimeters]

Initialize the stepper motor:
- Use Pin 8 as the pulse (PUL) pin
- Use Pin 9 as the direction (DIR) pin

Create stepper motor object with stepsPerRevolution, Pin 8, and Pin 9

Setup:
- Set the speed of the stepper motor to the desired RPM
- Initialize serial communication with the desired baud rate

Loop:
- Check if there is serial input available
- If input is available:
  - Read the input and check if it is a letter between 'A' and 'H' or positions 1-12
  - Convert letter to a corresponding position if needed:
    - Map 'A' to position 0, 'B' to position 1, ..., 'H' to position 7
  - Calculate the distance to move:
    - Each position corresponds to a movement of 9mm
    - Calculate the number of steps to move using the formula: steps = mmToSteps(9 * (targetPosition - currentPosition))
  - Move the stepper motor by the calculated steps
  - Update the current position to the new position

Function mmToSteps(mm):
- Convert the distance in millimeters to steps using the formula:
  return round(mm / stepDistance)

[Optional]
Function for stepper movement:
- Define a function to handle the movement of the stepper motor

Arduino Pseudocode for Dipping Mechansim

Include Stepper library

Define constants:
- stepsPerRevolution: 400 (Number of steps per revolution of the motor)
- stepDistance: 0.00396875 inches (Distance covered by the motor per step)

Initialize stepper motor object (myStepper) with stepsPerRevolution, Pin 8 for pulse, and Pin 9 for direction

Setup Function:
- Set the speed of myStepper to 500 RPM
- Initialize serial communication at 9600 baud rate
- Flush the serial communication to clear any existing data

Define steps as the number of steps for a 1mm movement using mmToSteps function
Define iters as 6 (number of iterations for the movement)

Loop Function:
- Check if the user presses 'y' to move the stepper motor down
  - If yes, call runner function with iters and -steps (to move down)
  - Wait for 1 second (delay)
- Check if the user presses 'y' to move the stepper motor up
  - If yes, call runner function with iters and steps (to move up)
  - Wait for 1 second (delay)

Function mmToSteps(mm):
- Calculate mmPerStep as stepDistance converted to millimeters (1 inch = 25.4 mm)
- Return the number of steps for the given mm distance, rounded to the nearest integer

Function runner(iterations, steps):
- Repeat for the number of iterations:
  - Move the stepper motor by the given number of steps

Function promptUser(expectedChar):
- Prompt the user to press a specified character (expectedChar) to move the stepper motor
- Keep checking for serial input indefinitely
- If the expected character is received, return true