NanoPack for NanoView: First Semester Report

Devin C. Bidstrup dbids at bu dot edu,
Joseph Walsh jwalsh15 at bu dot edu,
Paul Stephen Hutchinson Maltaghati psmalta at bu dot edu,
Justin Melville jem2000 bu dot edu,
George Kent-Scheller georgeks at bu dot edu

CONTENTS

***Abstract*—As NanoView Biosciences [1] expands and offers its services to more customers, it will need to ramp up the production of its signature product, which is a specially treated silicon chip that allows its customers to analyze biological samples. The current production process for these chips includes a step where a lab technician must use tweezers to manually move chips from the internal carrier called a *traveler* into their packaging for shipping called *clamshells*. This is both labor-intensive and error-prone. Our proposed solution is an automatic packaging device that is 10 times faster than human-packing and resistant to error when placing chips. This will comprise both a packing machine and an accompanying software package to control the machine. The machine will be a CNC-style device able to perform x-y-z motion with a tweezer actuator at the end of the arm to pick up the chip. From the software program, one will be able to start the packing, monitor its status in real-time, detect errors, and terminate it if need be.**

Section Author: Devin Bidstrup

I. INTRODUCTION

IN the field of bioscience, there is a demand for precise portable cell imaging technologies. Our client, NanoView BioSciences^{TM}, has developed a proprietary imaging technology that allows labs to either take in-house samples and then process them with the help of a specialized machine, or send them to NanoView for analysis. This imaging process involves placing the desired biological medium onto a specialized silicon chip and then inserting the chip into the machine for analysis. The machine is capable of detecting exosomes as small as 50 nanometers and can identify up to 5 separate biomarkers on the sample using fluorescent light and dye. This technology allows NanoView to command a competitive advantage in their field. Hence, the company must expand production in the future.

Currently, NanoView is experiencing a manufacturing bottleneck. To get the product to the market more efficiently, NanoView started off doing most of their production by hand. As they have grown, they have attempted to increase throughput at each step in their production line. To decrease bottlenecks, they have decided to have a central tray that carries chips between manufacturing processes, called the *traveler*. The manufacturing processes are conducive to this model as many of them involve scanning the chips or coating the chips (i.e. the chips do not need to be removed from the tray). One of the final processes in the manufacturing of the silicon chips is packaging these chips to send out to the customer. This step is causing a restriction on the output which our project aims to alleviate. Currently, chips are pulled from the tray by hand, checked visually against a spreadsheet, and then placed in their corresponding packaging. This process is vulnerable to human error, ties up employees on menial tasks, may damage the chips, and most importantly is slow. We aim to automate the final packing step, thereby increasing the speed at which it is performed.

To automate this process we will provide NanoView with two things: a packaging machine and a software program. The packaging machine will be a CNC-style device capable of moving along three axes. Fitted to the end of the moving arm will be a tweezer-like actuator which will be used to transport the chips, making sure to grab the side of the chip to avoid damage. Attached to the machine will be two cameras, one to capture a broad view of the entire device and one attached to the arm to capture the chip number at a given point in time. This will allow us to leverage machine vision to capture in real-time the chip numbers that the arm is moving, and the positions of the chips, traveler, clamshell, and arm. The software program will provide the user with the ability to start the packing, monitor its status in real-time, detect errors, and terminate it if need be. The software package will be portable so that it can be installed on any Windows device.

With our system, the lab technicians will be able to insert a full traveler and empty open clamshells, open the software package, load a CSV with the chip information, and begin packing. This will allow NanoView to pack with far greater efficiency than before. In addition, should the machine detect discrepancies between the chip numbers and the CSV, it will alert the user. In this way, we provide NanoView automated packing and error detection for a part of their manufacturing process.

Section Author: Devin Bidstrup

II. Concept Development

NanoView needs us to move 64 1cm2 silicon chips from an 8x8 grid to separate packages with 4 chips each for one run of the packing machine. These 4 chip packages will contain certain chips from the 8x8 grid as determined by an input CSV file so that the customer receives the correct chips. The customers' top priority is to improve the speed at which this packing happens by a factor of 10. Not only that, but the chips must remain undamaged during the packing process, with some areas of the chips being especially sensitive to touch. Our machine must be easy for the worker to load the traveler and clamshells, and it must work with their current manufacturing process and space. It must be possible for the lab technician to begin the packing process, be notified of errors, and see the results in real-time. In addition, due to the necessity of the CSV file being used to determine where to place the chips, we must provide a software program capable of operating the machine. Checking that the chips in the traveler match up with the CSV file and that the correct chips get placed in the correct clamshell is also essential for the customer to get what they desire.

Functions	1	2	3	4
Input for packer (traveler)	Human into slot for machine	Magazine of loaded travelers		
Input for the packer (clam shell)	Human	Conveyor belt	Stack	Layout on table
Output from the packer	Conveyor belt to bucket	Manual extraction	Vacuum sealing(fully packed)	Separate clamshells into kits (4 shells, 16 chips to 1 kit)
Chip grabber from traveler	claw / tweezers	Scoop from underneath with small spatula like arm	Suction cup	
Chip placement into clamshell	Dropping from low height	Pushing from Actuator		
Arm Movement	CNC-style x, y, z motion	4-axis robotic arm		
Arm guidance	Machine Vision	Accelerometer	Rotary Encoder	
Throughput	2 machines	Fast machine	Multiple chips concurrently	
Software driven control for machine	Microcontroller	ASIC		
Image Processing	Desktop with usb mux	Image processing server with usb out		
Power for machine	Wall Outlet (120V)	Battery	Wall Outlet (208V)	
Gathering of meta-data?(chip number/ QC check?)	Existing NanoView camera setup	Existing data from previous step(software input)	Existing data from previous step (scannable input)	Camera setup different than existing setup
Motor Control	i2C / serial communication with controller	PWM		

TABLE I
Morphological Chart

With all of these requirements in mind, we set about determining the best means to accomplish the functional requirements. We listed these means in the morphological chart shown in Table I. In the end, we chose the means highlighted in green, but that choice occurred after some debate on each item.

In particular, earlier in the semester we had been debating between two different types of arms to make our machine out of. First, we had the 4-axis robotic arm. This uses inverse kinematic equations to move the arm precisely across many joints, with a swivel at the base so that it can rotate. These kinds of arms have been demonstrated to be capable of picking up and putting down items that are placed on a surface [2]. We even mocked up a CAD model for this design and presented it at the alumni shark tank. However, we instead decided to design our packing machine with a CNC-like design, where x, y, and z stepper motors are converted to the linear motion used to move in the respective axis. This design is more stable, easy to build out of 80-20, easier to mount cameras to determine locations, and easier to program. Therefore, after some debate, we decided on this structure and form of arm movement.

Another feature we greatly debated was how to insert both the clamshells and the traveler into the machine. We considered heavily using conveyor belts for both input and output from the packing machine. If implemented correctly, this could greatly increase the throughput of the machine, which was one of the most important metrics for this project. In the end, we consulted with our client to determine how they expected it to operate. The client preferred that a human load and unload the machine, as they would already be carrying it from the previous step in the manufacturing process.

In total, our final solution will be a CNC-like packing machine that utilizes tweezers at its head to pick up the chips, whose arm is guided by the locations given by machine vision from the traveler to the clamshell where the chip will be deposited from a low height into the clamshell. The machine will be powered from the wall, and the motors will be controlled by digital stepper drivers which receive a PWM signal to stimulate rotation. An initial scan will determine the chip numbers of all chips in the traveler and this will be checked against the existing data from the previous step, given as a software input (CSV). These images will arrive at the computer via two separate USB cameras, one for the entire setup, and one attached to the arm to read chip numbers. A human will insert the traveler and the clamshells into the machine, and then remove the filled clamshells once they are packed.

Section Author: Devin Bidstrup

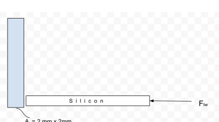
III. System Description

A. Mechanical

Our client will receive a system consisting of two principal components. Firstly we will deliver a physical machine that will be capable of packaging a silicon wafer chip into a set of containers. This machine will be constructed primarily out of aluminum extrusion bars which are fastened together. It will be able to fit in our clients facility by having dimensions that do not exceed 2' x 2' x 2'. There will be an interior area called the packing bed that will be a flat surface with dimensions of 2' x 1.5'. The machine will be able to move an actuator with 4 degrees of freedom within the packing bed.

This movement will be powered by a series of NEMA stepper motors all running on 12 v DC. They will be controlled by a series of motor drivers taking input from a microcontroller with a large number of output pins capable of creating a 5 volt PWM signal. These motor drivers will be powered by a 50 watt 12 volt dc power supply running off of 120 volts 60 hz ac. The microcontroller will also have an lcd panel to display the current progress of the packing procedure. There will also be two emergency stop switches. A first which sends a software interrupt to the arduino to allow it to fail into a safe state, and another which cuts all the power to the motors.

The movement of the actuator will be controlled in the Z axis by a servo connected to the microcontroller to control Z axis rotation as well as a microstepper linear actuator to control Z axis elevation. The actuator will grab the chips, with dimensions of 9.5mm x 9.5mm x 2.5mm, by closing a moving arm onto a fixed arm to create gripping force through the use of either a stepper motor or a servo. This force shall not exceed 520 N to avoid deforming the silicon wafer. Our calculations for this figure can be seen below.



$$\sigma_{Si,max} = 130MPa = \frac{F_{tw}}{A_x}$$
$$\frac{F_{tw}}{(2*10^{-3})^2} = 130MPa$$
$$F_{tw} = 520N$$

The entire assembly will be able to place a chip and return to start position with a time of no more than 8 seconds. It will also be able to precisely place the chip within the packaging wells that each have a dimension of 10.5mm.

B. Software

The other component of the project our customer will receive is the associated software package to run the machine. This will consist of a GUI and control system that allows the user to start packing jobs from a windows computer. The software will consist of three sub elements.

1) Graphical User Interface (GUI): There will be a user interface that will be written in the .net framework. This user interface will give the user the ability to load the necessary configuration files to start the job. These files will be CSV's and should already be created as part of the previous step in our clients production line. After loading this the user will be able to start and stop the packing process from the GUI. They will also be able to see the progress of the job and they will be able to change the classification of chips in the design. The GUI will also create logs of the packaging process and allow for the user to look at a live output of the current process being done by the machine to aid in debugging. The GUI will also communicate with the python control module asynchronously to allow the user to change settings on the program without having to wait for the program to complete.

2) The Control: There will be a control program that is responsible for sending instructions to the microcontroller. The control module will communicate with the GUI through a Win 32-pipe. It will also communicate with the microcontroller over usb serial. There will be two custom supporting python libraries that support the control program.

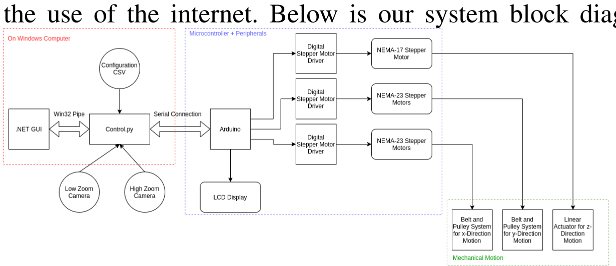
One will use a statically placed camera that can capture the entire field of the packing bed. This module will use computer vision implementing simple object detection to create a list of object locations in 2d space. It will then provide that list to the control module. The second module will use a Convolutional Neural Network to classify image data into 10 different numerical digits. It will output these labels to the control program. The second module will be fed image data from a high zoom camera attached to the actuator.

The Control program will also create an efficient pathing plan given a chip's location in the actuator. This position is given by the csv file and the text scanning step. The locations of the traveler and the clamshell will be given by the object detection module. After creating the pathing the control module will output these instructions to the microcontroller along with metadata about which chip is being placed.

The control program will also be able to be interrupted by either the Microcontroler or the GUI and to pass the interrupt through to the other program.

3) Microcontroller: The microcontroller will receive data from the control module via serial communication. It will then interpret the instructions and display the metadata on the lcd screen. The microcontroller will also interpret the control instructions and output the desired motion to the corresponding motor controller. The microcontroller will also send feedback about the taken steps back to the control program and any interrupt signals that the program receives.

The entire software suite will be compiled into a windows executable and will be runnable without any separate installs for supported libraries. It will also be able to function without the use of the internet. Below is our system block diagram.



Section Author: George Kent-Scheller

IV. First Semester Progress

We have been working hard throughout the semester to produce a viable design for NanoPack. While each component of our project has gone through a lot of trial and error, we are now in a good position going forward into next semester.

A. Mechanical

At the very beginning of the semester, we brainstormed different robotic devices which we could build. The two which we ended up presenting in our PDRR were a robotic arm and a CNC-style machine. At first, it seemed like the robotic arm would be easier to design. We even demonstrated a robotic arm to the alumni at the Shark Tank presentation. However, after receiving feedback and realizing that the arm design would be difficult to implement given the dimensional requirements of the project, we decided to switch over to the CNC machine. After doing this, we then had to decide what components and motors to purchase. 80-20 always seemed like the go-to in terms of chassis material due to its high motor utility and overall strength. For our motors, we decided to mainly work with stepper motors with a very small step angle. Members of our team had previous experience with using stepper motors for high accuracy projects, and since we would not be requiring high torque due to the low weight of the chips, stepper motors seemed to be the most efficient. For the section of our machine which would actually be responsible for grabbing the chips, we designed tweezers which would be powered by a servo motor due to the precision needed. After determining all of our parts, we were able to model our design in CAD. The designing was done in OnShape due to the collaborative nature of the program. Once all of the individual sections of the connectors, linear actuator, and end effector were created in OnShape, the CAD files were exported to SolidWorks and combined. SolidWorks CAD is the framework which we demonstrated in the first prototype testing. After our first prototype testing, we wanted to add a bit more detail, so we added the entire chassis framework as well as a more detailed tweezer design of the end effector, which was presented during the PDR and is our current design. A lot of our prototype will likely be changed next semester since NanoView will be providing our team with a robotic kit which comes with a structural frame, motors, pulleys, and belts. Our team will be assembling the kit over winter break and determining whether or not it can be used. We will have to see how much of our current mechanical design will need to be implemented.

B. Electrical

Our project is very software and mechanical heavy, and the main electrical portion of our project has so far been the wiring connecting the microcontroller to our mechanical robot. Our electrical design has stayed consistent throughout the semester, and it has been very dependent on the components which we purchased. All of the motors which we will be using will be powered by a stepper motor driver. This stepper driver will receive instructions from an Arduino Mega microcontroller which will be plugged into the computer. The driver receives its power from a CNC power supply that is plugged into the wall. All components are connected via wires, and the LCD display is stationed within a breadboard. After we received news that we would be receiving the kit, we decided to wait and test the kit's electrical components before doing more wiring for our motors.

C. Software

We have spent most of our efforts this semester working on the software package, which as described earlier consists of a graphical user interface (GUI), a control system, and the microcontroller. All of these parts have been developed extensively since we first started.

1) Graphical User Interface (GUI): Our team knew that we would be using the .NET framework ever since we first started designing the GUI. Our team had previous experience working with .NET, and we knew that it could be implemented at NanoView since all of their computers were Windows based. Our GUI was first created with all of the buttons as well as the CSV data display. It also included a diagram of all chip locations within the traveler. As the semester progressed, we added more color to the GUI, as well as incorporated the logo into the home screen. Most of semester went into the development of connecting the GUI with the rest of the Python script, which will be further discussed in the next section.

2) The Control: Since our team was most experienced with Python, and since Python is very useful when working in machine learning, we determined to use it for our serial communication as well as the image recognition. Initially, we thought that we would be able to write a single function within the .NET program which could connect it with Python. This did not work, so we had to implement a Win32 pipe. A lot of work recently went into making the pipe asyncronous, and we will be continuing to make improvements going into next semester. With regards to the computer vision and machine learning, we were able to do a lot of work with the chip recognition, but due to a lack of data and materials, we have yet to make an algorithm which can consistently do number indentification.

3) Microcontroller: Our team also had experience working with Arduino microcontrollers before and in embedded systems engineering, so programming the Arduino was something which was done relatively quickly. The microcontroller now controls both the LCD display and the motor driver, and we had demonstrated that it reliably sends commands to the robot.

Section Author: Paul Stephen Hutchinson Maltaghati

V. Technical Plan

Task 1: Assemble client-provided CNC kit

As soon as we receive the CNC kit from our client, we will assemble it. We will evaluate if all the of the provided parts are capable of fulfilling the specifications set for us by the client. One area of concern is the camera resolution, which may not be high enough, depending on which kit model was ordered. Some aditional parts not available in the will be printed in PETG on an Ender 3 pro. This includes the tweezers module to pick up the chips.

Lead: Joseph

Assisting: Paul Stephen

Task 2: Order parts

While we believe the kit should contain all the parts we need, we need to be prepared to order new materials if it does not contain everything we expect. We will do this as soon as we receive the kit to ensure we receive them in a timely manner.

Lead: Joseph

Assisting: All

Task 3: Integrate our existing motor control with the kit

One potential challenge with the kit is we the uncertainty surrounding the compatibility of our current software suite with the hardware that will be provided to us. We currently control our peripherals (e.g. Motor) through an Arduino Mega. The kit we will be provided uses a motion control board called the TinyG. We will determine which board to use, and then integrate it with our existing software suite so that we can control the motors on the machine.

Lead: Devin

Assisting: Justin, George

Task 4: Generate real data for machine learning models

Currently, our machine learning models have been trained off of generated data as opposed to real pictures of chips contained within the traveler. This will involve using the intended USB cameras once they arrive, at least to test the error rate of our model. Using real pictures will present many different problems that do not currently exist in the software-generated data. Chiefly, we are worried about the effect that light sources may have on the chips, which are very reflective.

Lead: Paul Stephen

Assisting: George

Task 5: Implement ML model for number recognition

While we currently attempted to use some existing models to do this task, the idiosyncracies of the font used on the chips mean that pre-trained models are not effective at recognizing certain numbers, such as the 8, which is missing the top right corner, and is therefore identified as a 6. Therefore, we will build our own convolutional neural network to implement number recognition on the chips.

Lead: Paul Stephen

Assisting: George, Justin

Task 6: Update existing CSV parsing to use NanoView's format

Despite asking for sample CSV files from NanoView since the beginning of the semester, we still have not received any. Hence, we decided to make our own CSVs to temporarily proceed with. As soon as we get the final format we need, we will update our existing parsing and visualizations to use the new format.

Lead: Justin

Task 7: Complete GUI

While we have added the features we believe will be helpful to the GUI, it will be necessary to contact the client to ensure we have implemented the features they would like to see added to this app.

Lead: Justin

Task 8: Pick up chips

We will need to implement the capability of the arm to pick up chips. We will do this with very high precision as to avoid breaking or dropping the chips. Include a way to assure the chip will not be damaged by the tweezers.

Lead: Joseph

Task 9: Control arm movement

We will need to implement the capability for our arm to move from the the location in the traveler as specificied by the CSV to the location in the clamshell as detected by our machine learning models. This will need to be done with a high level of precision to ensure consistency when picking up and placing our chips. This will involve coordinated movement of the motors to move the arm to the desired location as given by our pathing algorithm (see task 10).

Lead: Devin

Assisting: All

Task 10: Control arm path

We must ensure that the control arm is moved relatively quickly to meet the speed requirements set by our client. This will involve a pathing algorithm located in python to compute the best order to move the chips in to waste the least time. While it is unlikely that we will reach the optimal solution, optimizing the path to the best of our ability is essential.

Lead: George

Assisting: All

Task 11: Place chips

We will need to implement the capability for our arm to reliably and accurately emplace chips in the clamshell. We will account for exact location as well as angle to place the chip in the safest possible manner.

Lead: Joseph

Task 12: Error handling

Although we will reduce our error rates as much as possible, there will still be situations outside of our control, such as mismatching data. We must ensure that these situations are all handled in an appropriate manner. We will work systematically to ensure all indeterminate behavior is eliminated.

Lead: Justin

Assisting: Devin

Task 13: Client Installation and Review

We will install the machine and software at our clients location once it is complete, and receive feedback from our clients using the machine.

Lead: All

Section author: Justin Melville

VI. Budget Estimate

The budget we were given for this project was $1000, with the note that more funds could be made available if needed. The way we plan to source a lot of our materials will be from a Lite Placer robot kit purchased by NanoView for us to modify. The price of the kit online is $1799. Items we will be using from the kit are marked with an asterisk in the budget chart. The prices for these items were estimated from prices quoted to us from sellers we had planned to use before being told of the LitePlacer kit. Additional parts, such as the 80/20 connectors and some of the sliders included in the kit may be used as well rather than being 3D printed. Most of the rest of these parts were ordered from Stepper Online and came in a single shipment. The micro linear stepper motor came along with the NEMA-23 motors from an Amazon oreder request and the IPSG board was purchased from Microcenter

For 3D printing, the filament we will be using will be PETG because of its strength and durability as well as being easily printable. The cost of PETG is about $0.05/gram and for this project we plan to use around 400 g for a total material cost of around $20. We will use a 1.75 mm diameter filament fit for printing on most FDM printers including the Ender 3 pro.

Item	Descripton	Qty.	Cost
1	80-20 bars	16 ft	$215.60*
2	NEMA-17 stepper motor	1	$17.99
3	NEMA-23 stepper motor	2	$47.98
4	CNC power supply	1	$29.18
5	GT2-3M timing belt	16 ft	$55.84*
6	GT2-3M timing pulley	6	$47.94*
7	Micro linear stepper motor	1	$23.19
8	Arduino MEGA 2560r3 Main Board	1	$14.99
9	Digital stepper motor driver	3	$51.48
10	PETG filament	400 g	$20

Section Author: Joseph Walsh

VII. Appendices

A. Engineering Requirements

1) Increase the throughput of clamshells from 100 chips placed in eight hours to 1000. This would be a 10x increase compared to current speeds.
2) Create a machine that will package these clamshells with at least the same level of care that a human packer uses.
3) Create an accompanying software package that allows the client to operate the machine in a configurable manner with relatively little input.
4) Create a system to check chip identification numbers so that the correct chips get placed into the correct clamshell.
5) Integrate the packaging process into the existing manufacturing pipeline.
6) Ensure that there is no contamination of chips during this process.
7) Make sure that this machine will fit in the clients manufacturing facility.

B. Gantt Chart

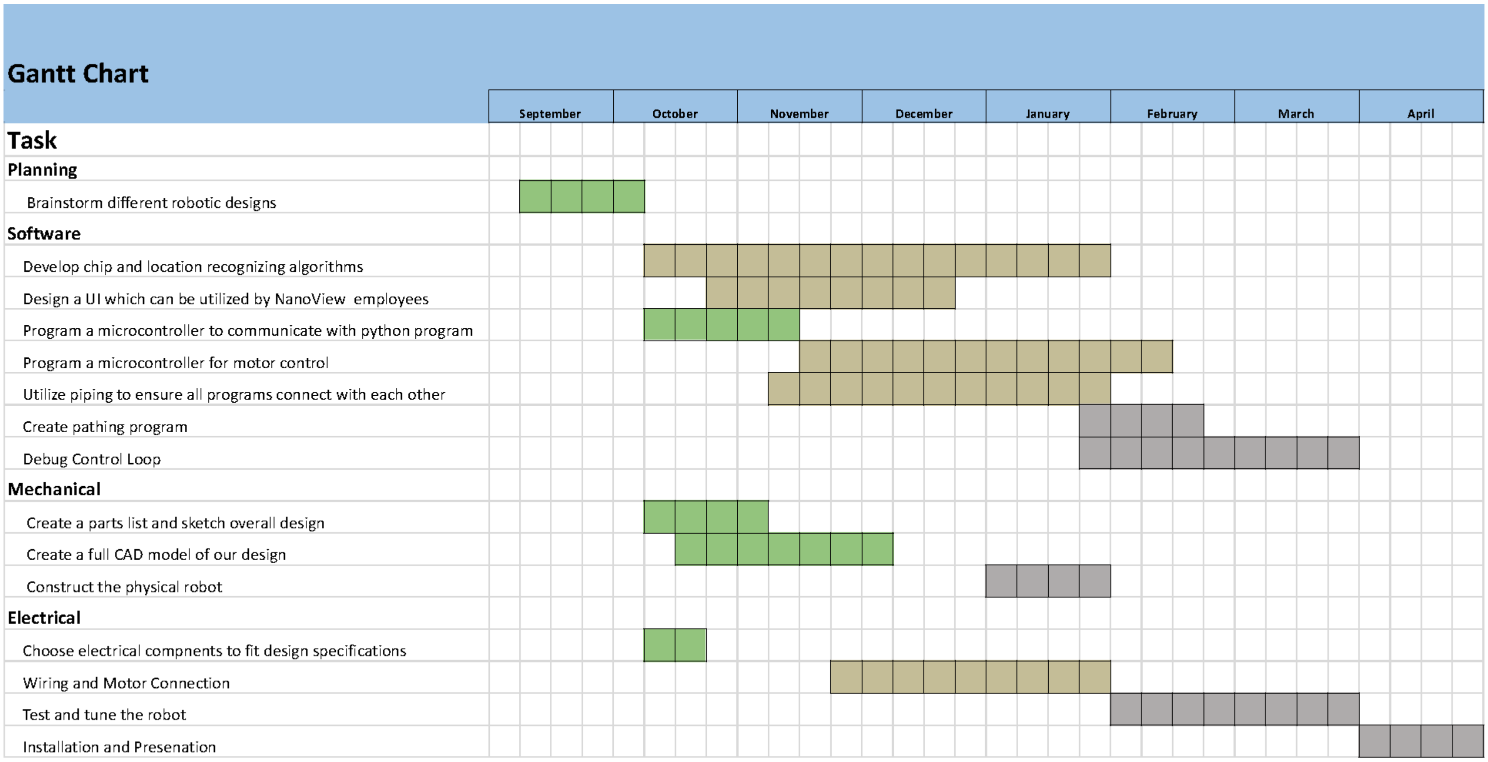


Fig. 1. Gantt Chart

ACKNOWLEDGMENT

The authors would like to thank David Freedman and Steve Scherr for sponsoring this project, providing us with the information needed to make this project possible, and for allowing us to visit the NanoView BioSciences facility to demonstrate the process of removing chips from a traveler and placing them in a clamshell.

We also would like to thank professors Alan Pisano, Michael Hirsch, and Osama Alshaykh for their continued guidance and mentorship throughout this project.

REFERENCES

[1] D. Freedman, Characterize exosomes: Extracellular Vesicles: Nanoview biosciences, *Characterize Exosomes — Extracellular Vesicles — NanoView Biosciences*. [Online]. Available: https://www.nanoview-bio.com/. [Accessed: 12-Dec-2021].

[2] Fortzero, 4 axis robot arm DIY, *Instructables*, 28-Sep-2017. [Online]. Available: . [Accessed: 06-Dec-2021].

[3] Properties: A background to silicon and its applications, *AZoM.com*, 10-Dec-2021. [Online]. Available: https://www.azom.com/properties.aspx?ArticleID=599. [Accessed: 11-Dec-2021].