

Product Design Review

Devin Bidstrup

Justin Melville

Joseph Walsh







Our client, NanoView Biosciences, is a biotechnology company which makes machines capable of detecting markers on exosomes and other biological systems

They are a startup company based in Brighton, and their President is a former BU professor, David Freedman

### **Problem Statement**

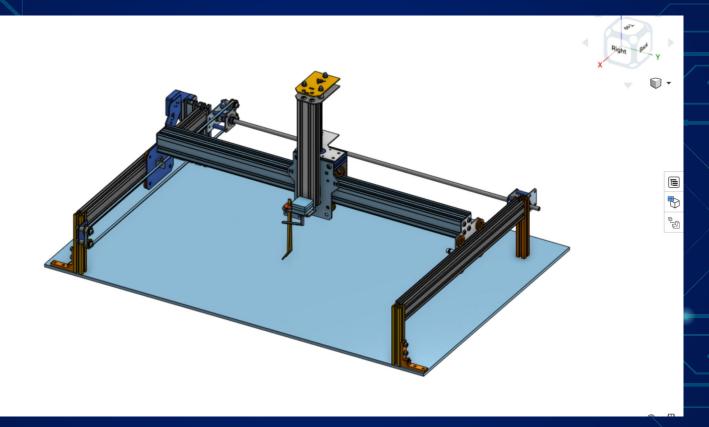
- NanoView currently uses a person with tweezers
   when they transfer silicon chips from their lab
   container, "the traveler" into user cases, "clamshells"
- Our job is to automate the process of transferring the chips from the traveler to clamshell
- This will speed up the process, prevent human error, and reduce risk of contamination

### **Deliverables**

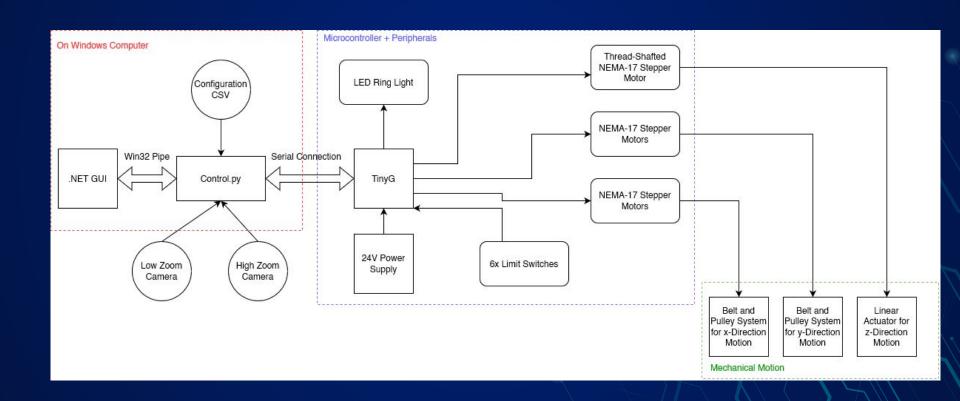
We will deliver a product meeting the following requirements:

- 1. Comprised of packing machine and accompanying software package
- 2. 10x increase in packing speed
- 3. Same level of care as a human
- 4. Check chip ID as chips are packaged
- 5. Integrate into existing manufacturing pipeline
- 6. No damage to chips during packing
- 7. Fit inside the manufacturing facility

# Visualization

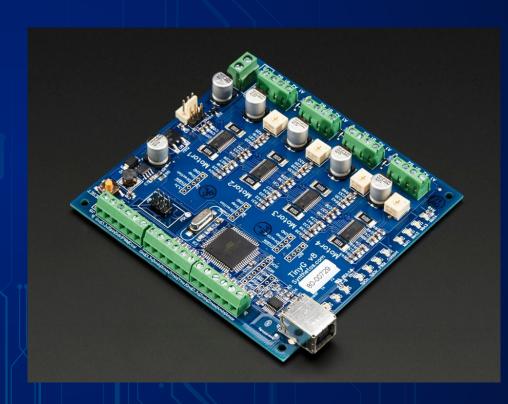


# **System Block Diagram**

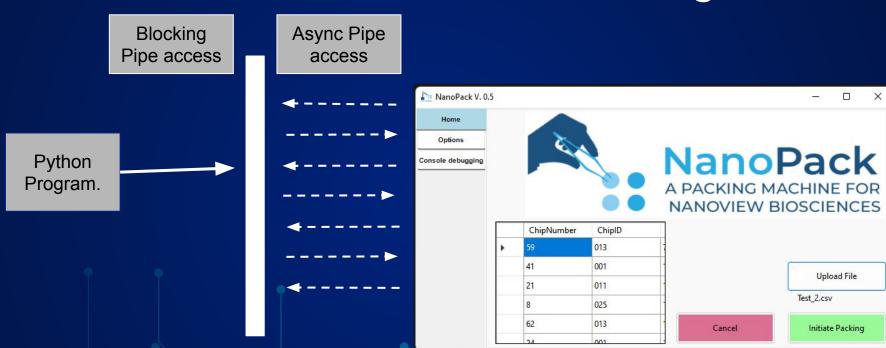


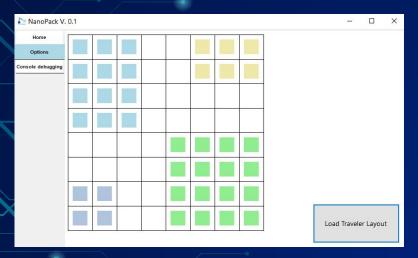
### **Technical Design - TinyG Microcontroller**

- Serial G-code → stepper motor
- Communicate via Python and pySerial
- Serial port detected automatically with the VID of the board
- TinyG motor and communication protocols are configurable via serial
- G-code commands to move the machine are sent JSON format
  - e.g. {'gc', 'G0 X 1'} to perform a G0movement on the X-axis



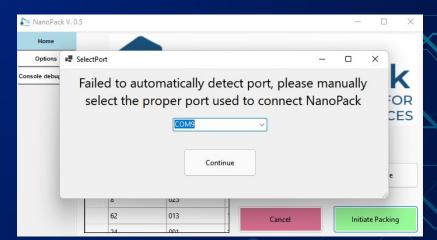
# **Technical Design - GUI**





# **Traveler Visualization**

### **Port Selection**



# Example of inter-process communication

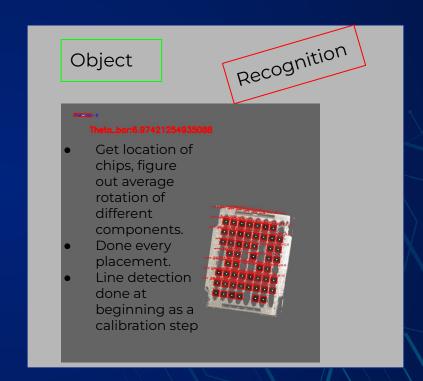
# NanoPack V. 0.1 Home Options Console debugging Give me CSV, initialize Sending packing report 0 Sending packing report 1 Sending packing report 2 Exited with code: NormalExit

	Reading CSV
	ipNumber -> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
52	26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 2, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64]
3	lipIO -> [25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25, 11, 1, 3, 25,
C1	11, 1, 3, 25, 11, 1, 3]  LamshellGroup -> [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7, 8, 8, 8, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13, 14, 14, 14, 14, 15, 15, 15, 16, 16, 16]
	Chip Locations from Sample Image
	LOOP 0ssage from .NET GUI: Data Received
	LOOP 1ssage from .NET GUI: Data Received
	LOOP 2
	kited normally cosing pipe
C:	\Users\justi\Documents\GitHub\NanoView_G33\dev\first_prototype>

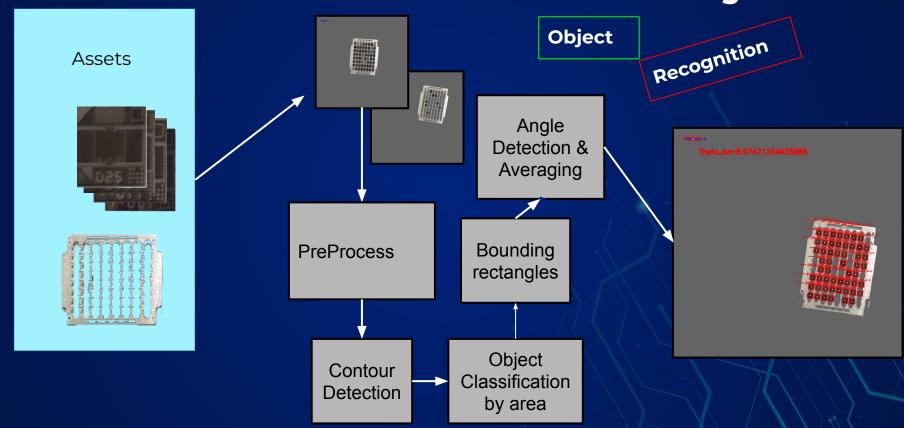
### **Technical Design - ML**

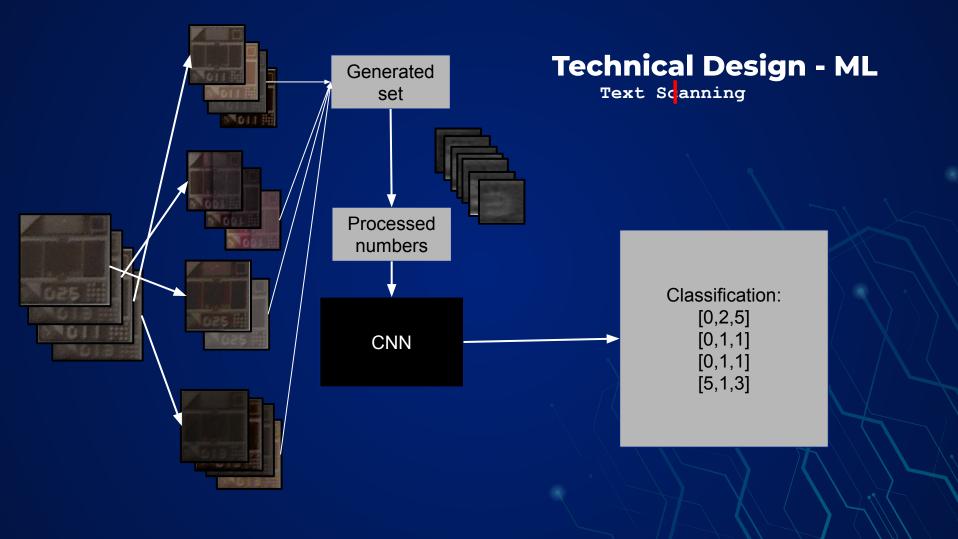
#### Text Scanning

- Use a CNN to classify images into number categories.
- Verify against csv before starting chip placement
- Only done at the beginning of the processes.
- Generated noise on top of existing image set to create a larger training.



# **Technical Design - ML**

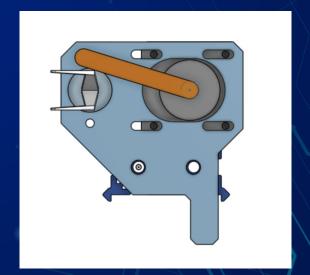




# **Technical Design - Mechanical**



- Modified design from LitePlacer robot
- Custom tweezer actuator designed to pick up chips
- Added 3D printed mount for tweezers + stepper arm to pinch tweezers



# **Schedule (Gantt Chart)**

Gantt Chart								
	September	October	November	December	January	February	March	April
Task								
Planning								
Brainstorm different robotic designs								
Software								
Develop chip and location recognizing algorithms								
Design a UI which can be utilized by NanoView employees								
Program a microcontroller to communicate with python program								
Program a microcontroller for motor control								
Utilize piping to ensure all programs connect with each other								
Create pathing program								
Debug Control Loop								
Mechanical								
Create a parts list and sketch overall design								
Create a full CAD model of our design								
Construct the physical robot								
Electrical								
Choose electrical compnents to fit design specifications								
Wiring and Motor Connection								
Test and tune the robot								
Installation and Presenation								