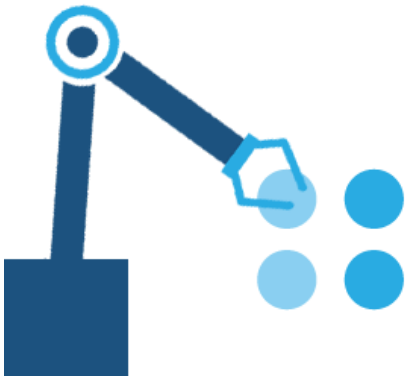
Boston University
EC464 Senior Design Project
Final Prototype Testing Plan



NanoPack
A PACKING MACHINE FOR NANOVIEW BIOSCIENCES

BY:

Devin 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*

Required Materials:

Hardware:

NANOPACK MACHINE

- LitePlacer Robotics kit
  - All mechanical parts
    - Framing, sliders, connectors, etc.
  - TinyG motor controller
  - USB cameras, LED ring lights
  - Stepper motors (3 NEMA 17, 1 NEMA 14)
  - Belts and pulleys
  - Limit switches
  - Cable chain
  - Vacuum Pump
- 24V power supply
- Wiring & connectors
- E-stop
- Test Computer
- Faux leather green screen

Software:

1. Python Control software
2. Machine learning program to process image from wide angle camera
3. Machine learning program to process image from actuator camera
4. .NET GUI for User Interface

Set Up:

The setup consists of both hardware and software components. From the GUI we will run a control program which will move the machine's actuator from rest to hover over a traveler, whose location will be determined using the large angle camera and a machine learning script. The machine will then use the actuator camera to take a photo and locate chips with the image processing. Then, the actuator will jog down the z-axis, create a seal with the chip using the vacuum pump, and return back to the clearance height. Then, the machine will move to the clamshell location using the object detection machine learning script. From there, the chip will be lowered down into the specified well where the vacuum seal will be broken as the chip makes contact with the sloped surface inside the clamshell slots. Our mechanical goals will be to have the machine moving within the workspace as well as a demonstration of the actuator movement. Our software goals are to locate the traveler and clamshells from the wide angle camera, move to these locations, use number recognition to verify the correct chip number, and lastly pick up and place the chips in their correct location.

Pre-testing Set Up Procedure:

- Connect the control computer to the TinyG and plug in the power supply.
- Ensure that the PWM light is solid red and no other red lights are on/blinking, otherwise reset the board.
- Open the lab computer, and open the file location of the GUI.
- Place clamshells and traveler and chips on the workspace.
- Place the chips on the traveler using the description in the csv
- Place the number of clamshell wells that correspond to your chip groups and number of chips.

Testing Procedure:

- Execute packing procedure in the software.
- An image is taken of the workspace by the wide angle camera.
- A machine learning program will process the image; locate the traveler and clamshells within the workspace.
- The control script will home the machine, then proceed to hover the actuator over the traveler.
- The program will then find the edge of the traveler in Y by itteritivately moving to the edge
- A second image will be taken with the actuator camera of the traveler
- A second machine learning program will process this image; checking that the number on the chip matches the number on the csv file.
- The machine will jog down to the work space, the actuator connects to the chip and then returns to the clearance height.
- The actuator will move over the proper clamshell well.
- The actuator will lower until the vacuum seal on the chip is broken and

the chip is in the well.

- This process will be done repetitively to demonstrate the packing process has been automated.
- A chip will be placed incorrectly in the clamshell due to an error in the CSV so that we demonstrate how the machine handles the situation when it encounters a mismatch between the csv and the machine learning

Measurable Criteria and Scoring:

The criteria for successful running and output is as follows:

Criteria	Completed (Y/N)
Object recognition and edge detection shows location of chips in traveler	
All digits of all chips are successfully extracted	
All numbers are correctly classified	
Chips correspond to the information given in the CSV	
Software notifies user of chip position error	
Machine moves to specific traveler and clamshell locations	
Vacuum pump picks up chips	
Chips are released in the clamshell slots	

Demo:

- From the GUI we run a program that moves from the center over to a point determined from the large angle camera, takes a photo and locates chips with the image processing, down on the z-axis, actuates the grabber (not necessary to pick up chip), moves up and back to where it started.

Things that need to be done to demonstrate:

- Figure out communication between GUI and control loop for the demonstration
- Actuator built
- Actuator motor wired
- Configuration automated with checks
- ML traveler location program working
- ML chip location within traveler program working
- Unit conversion from traveler location program to mm that the TinyG understands
- Home the TinyG units
- Fix the control loop for the demo
- WRITE AND PRINT 4 COPIES OF TESTING REPORT