

Boston University
EC463 Senior Design Project
First Prototype Testing Plan



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:

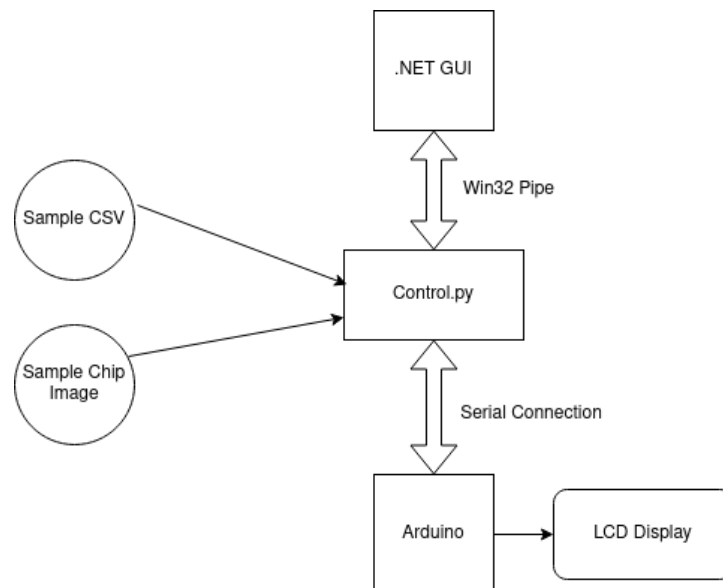
Hardware:

- Arduino Mega
- LCD display for arduino
- Breadboard and wiring for arduino
- Test Computer

Software:

1. .NET Gui
2. Python Programs
 - a. Picture location
 - b. CSV file input - [generated input]
 - c. Picture Text input
 - d. Serial Communication with Arduino
 - e. Pipe Control with .NET
 - f. Overall Control Module
3. Arduino .ino code

Set Up:



The overall setup consists of a python script which does the machine learning and main computations, a .NET Gui with which the user interacts, and an Arduino which controls the hardware peripherals (eventually the movement of the machine itself). The .NET process currently starts the python control process on startup of its GUI and then communicates via a pipe to indicate which CSV file should be used as the test case. Once it has been indicated that the machine should begin, the input CSV file is parsed and machine learning is performed on the generated chip image to find the locations. If

everything passes then the machine sends commands to the arduino to perform the movement of the motors. For this prototype, those commands will be displayed on the LCD screen, but in the future this will result in movement of the motors.

Pre-testing Set Up Procedure:

- Connect the arduino setup (wired and flashed before testing) via USB to the computer.
- Open the lab computer, and open the file location of the GUI .exe file "NanoPack UI (draft).exe" and the solidworks model.
- Open powershell to the path of the python control module (/dev/first_prototype/)
- If needed run the script:
 \dev\generateTestImagesMacro\create_random_Traveler_and_chip_and_clamshell_placement.py to generate the test images if needed

Testing Procedure:

- First Test: Test software stack with Python run manually
 - Open "NanoPack UI (draft).exe"
 - Use the first button to locate the CSV file
 - From powershell run the command "python Control.py"
 - Press the second button on the GUI to begin packing
 - Verify that output on Arduino, Python, and GUI are correct
- Second Test: Test software stack with Python run from GUI
 - Follow same steps as first test, except omit the powershell command
 - Verify output on Arduino and GUI (python cannot be viewed).
- Third Test: Demonstrate the solidworks model
 - Open up the model and demonstrate the x, y, z motion of the chassis of the mechanical design.

Measurable Criteria and Scoring:

The criteria for successful running and output is as follows:

<u>Criteria</u>	<u>Completed (Y/N)</u>
A windows application opens the CSV and tells python to begin packing.	
Using one of the generated test images, a machine learning model returns the locations of all of the chips in the traveler.	
Based on a preconfigured CSV file, we say what clamshells will contain what chip numbers and where they are currently located (Displayed to terminal).	
Prove that the arduino can perform serial communication with the computer by displaying the amount the chips needed to move to reach the different clamshell slots.	
Using one of the generated test images, a ML model returns the chip ID's for each chip.	
Generate test images randomly to robustly test image recognition.	
Solidworks assembly will show construction of the chassis of the machine as well as the machine's x, y, and z motions.	