Project Deliverable H:
Prototype III
GNG 1103
Prof Knox

Alex Milgram
Kalan Kok
Eric Wang
Charlie Huang
Peter Zhou

Table of Contents

Prototype Description

Hardware Component

Velocity Detector: Measures speed of passing object using laser sensor and relays signal to LCD screen

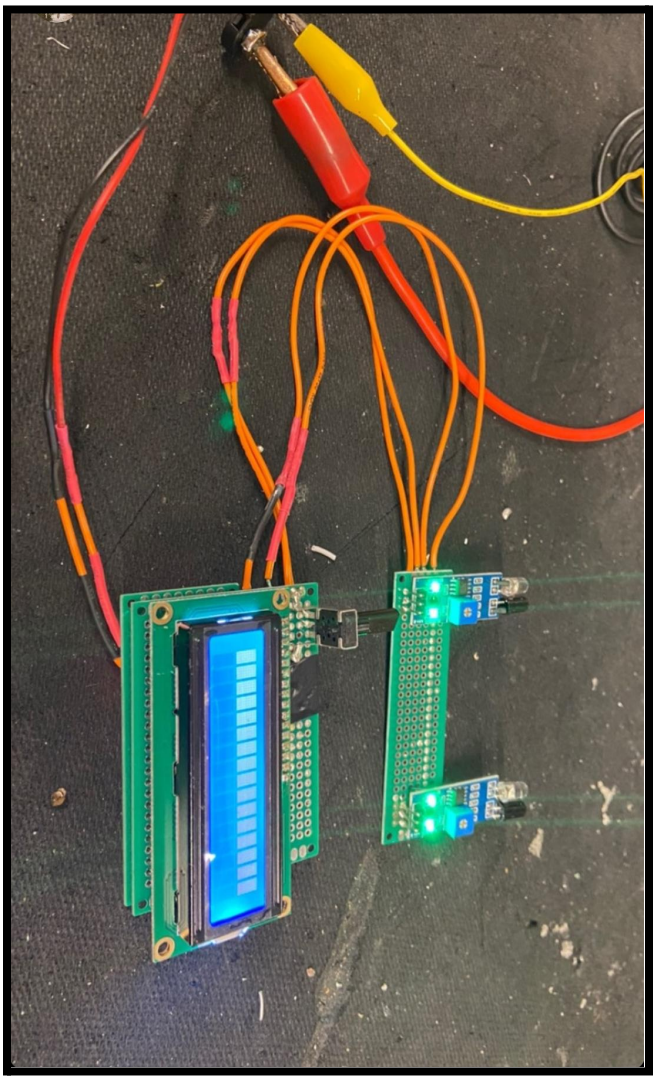

Fig 2.0

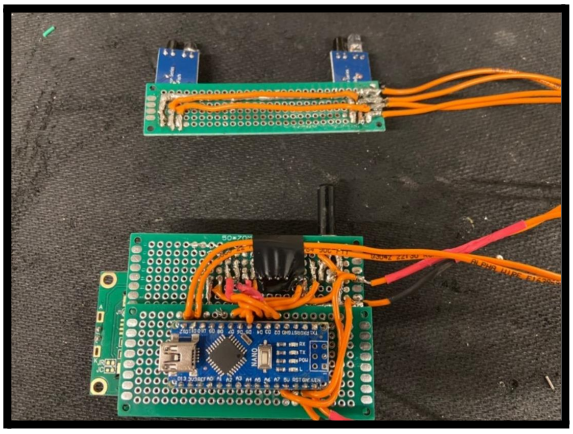

Fig 2.1

Software Component

- Accepts input values of speed at each node on the conveyor line, and calculates most efficient speed, based on theoretical v-curve theory.

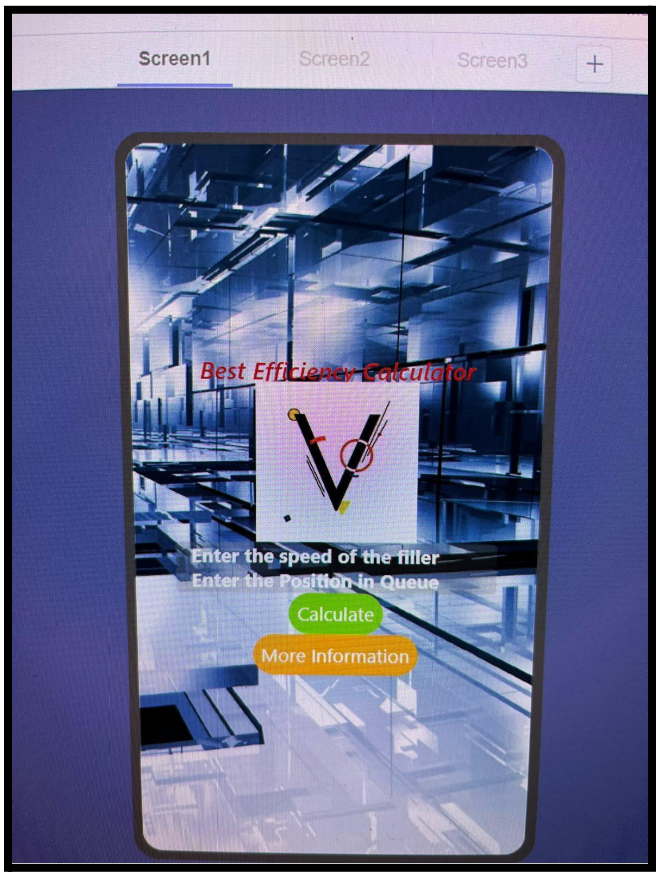

Fig 1.0

Prototype Test Plan

Testing for Prototype III:

1. **UI Visual Aesthetic Test**
2. **UI visual aids**
3. **UI usability features**
4. **Integrate physical prototype with app**

Test ID	Description of What was Done	How it was Tested	Duration of Test
1.	User interface was made more aesthetically pleasing through the use of background images (see fig 1.2 below) Various colors were chosen that were pleasing to the eye and created an improved UX.	We had many images to choose from, with various color combinations and designs. We asked random people which one looked more appealing to the eye, and more importantly, which ones looked more user friendly, thus creating a better UX.	Approx 30 min to choose a design. Amount of time asking people was not measured.
2.	To improve the app, various	The testing process for this	Approx 30 min. As

	visual aids were incorporated into the design of the app. As you can see in Fig 1.1 and Fig 1.2 below, the buttons added are color coded to be easy to see, and visually appealing. This testing process was done similarly to ID 1, with the same goal to improve UX.	step was almost identical to step 1. Since both are related to a visual aspect, the testing was done as a complete system, and not two separate systems with their own design. For future designs, if more buttons are added, more testing will be done that will evaluate not only the color scheme, but the placement and location of each button and feature of the app to improve UX.	aforementioned, time taken to conduct tests by asking people was not recorded.
3.	Various usability features to improve UX were added to the app. They can all be seen in Figures 1.0, 1.1 and 1.2	Since this aspect of testing relates more to the functionality of the app as opposed to the visual aspect, a different testing	30 min to 1 hour.

	respectively. In Fig 1.0, the option for more information was added. This gives the user more information in case of confusion. In Figure 1.1, the add data button was added, giving the user more control and flexibility when using the app. In Figure 1.2, one can see the save and update feature. This greatly improves the app's capabilities, functionality as well as usability. All these new add ons not only improve UX but also improve functionality. In the future, more buttons for UX	system than the two steps seen above were required. We gave the app to random people, and told them the basic premise of the purpose of the app. We then asked them to input random numbers. The testing arose when we wanted to see if any confusion arose, or navigational problems. Few seemed to arise except when an illegal character was entered resulting in an error. However, this is more of a coding improvement as opposed to a UI.	

	will be added such as a help button to aid with navigation.		
4.	Long story short, we have not been able to fully integrate the hardware component to the app as of yet in one complete closed system. However, the user can use the hardware system to detect the speed, and manually enter the seen speed into the app that will then compute the correlation with the v-curve theory.	Testing was done by first testing the hardware component, by moving objects in front of it and seeing if it gave a reasonable speed. Then we would input that speed onto the app for further calculation. There were little to no problems with testing this method.	2 hours.

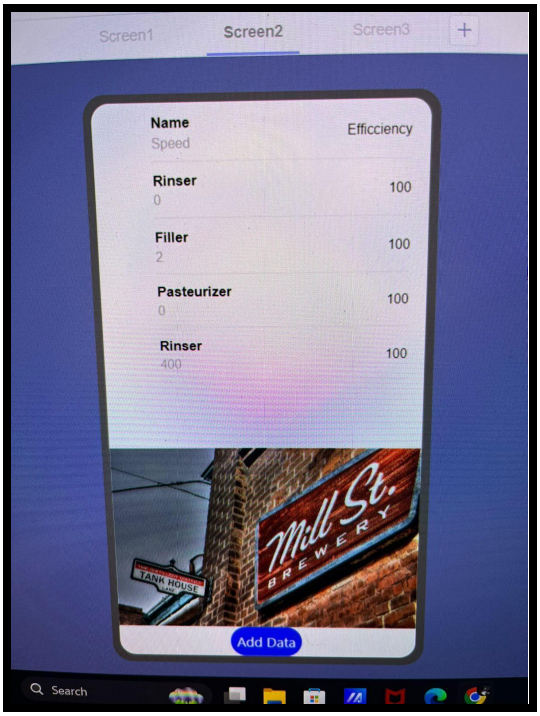

Fig 1.1 **Fig 1.2**

Prototype Analysis and Results

Feedback and Comments

Comment relating to difficulty in the efficiency calculation: Data provided by client differs greatly in speed and number of cans for each respective node. Thus, making the assumption for what is considered the most efficient speed for each respective node is difficult. A benchmark for efficiency is required, and due to the discrepancies in the client's data for each trial, the numbers can only be used as a sample, and not a benchmark. Therefore, even though it is evident that pure theoretical v-curve theory is not fully representative of the conveyor belt system, it is the only viable benchmark that can be used to calculate efficiency. More data would be helpful to set a more accurate benchmark.

Comment relating to visual representation of assumed efficiency calculation as explained above: It is difficult to create a graph on the Thunkable software used. It is plausible to send the data to an outside source such as matLab or another application such as excel that will then graph the data, however this

is very inconvenient. Another method of graphing must be devised to make the design as functional as possible.

Feedback from Random user: Simple to use, little to no problems with the actual usage of the app. Problems arose, when suggestions based on numbers were not quite clear, and could be clearer. “More features would be nice”.

Updated Target Specifications: *Visual component (UX) was optimized greatly since Prototype II

Can Read and interpret data:

Read from the keyboard. Provides suggestions/ reflection. Accept different kinds of data.

Can do calculations:

Accepts input and provides the result. Calculates the current speed and the ideal speed.(using modern v-curve theory) Then compares the two speeds.

Has a simple, visually appealing, and easy to navigate UI,

Simple windows and instructions for the windows. Use different colors to show positive and

negative influences. Show with bold text and clear worlds. Various buttons added in key positions ensures that navigation of the app is as smooth as possible with little to no confusion.