EE175AB Final Report

Pill Sorter

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

Project Team Member(s)	Peter Chau Nicole Chin Ted O'Young
Date Submitted	June 13, 2023
Section Professor	Professor Roman Chomko
Revision	Final Revision
URL of Project YouTube Videos, Wiki/Webpage	https://www.youtube.com/watch?v=4GyP8P8I9jE https://github.com/OYoung07/EE175AB-Pill-Itemzier https://github.com/OYoung07/EE175AB-Hopper-Code https://youtu.be/RquZzned6Y0
Permanent Emails of all team members	pchau013@ucr.edu nchin014@ucr.edu toyo001@ucr.edu

Summary

This report presents the documentation and progress of the project. It includes the hardware, software, and embedded technicalities of the design.

The objective for our project is to sort pills into separate piles according to their color. (Ted)

Note:

- **Sections marked with * are required**
- **In each section, you must clearly identify which team member is responsible for which objectives, modules or tasks.**

Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
Final	This is the final version of the report	Everyone	3/20/2023	Everyone, yes

Table of Contents

1 * Executive Summary

The Pill Sorter is designed to sort three pills into piles based on their color. The Pill Sorter is designed for the pharmaceutical industry. The pills' colors include red, blue, and yellow. As a result, this will reduce labor costs and error. It functions in a controlled lighting and temperature environment with the aid of a technician.

Key features of the design include a modular, compact design with computer vision, a transportation mechanism, and a NodeMCU ESP8266 which is a processing unit for a pill dispensing machine and a rotating chute to sort. The modular design allows for easy repairs and replicability of the machine. The compact design allows the Pill Sorter to be portable and low cost. Computer Vision is used as it only requires a web camera. The transportation mechanism provides a location for the color camera to scan the color, while moving the pill to the chute. The pill dispensing machine and a rotating chute are controlled by an inexpensive low power processing unit, a NodeMCU ESP8266. These features allow the machine to be portable and low cost, while functioning accurately. Given these features, this means that the scope of our project includes software, systems and electromechanical design, and electronic circuit implementation.

After testing, the Pill Sorter consumes less than 60 watts excluding the computer and camera, which will help with thermal management. Additionally, the time to sort each pill is less than 65 seconds. Important achievements of the Pill Sorter are high volatility with an accuracy of 100% for all test cases while working for long periods of time with low maintenance.

2 * Introduction

2.1 * Design Objectives and System Overview

The Pill Sorter automatically sorts pills into separate piles based on their color. An intended application of the device is to mechanize pill sorting in the pharmaceutical industry. The machine is an inexpensive, portable, and accurate way to separate pills. It is limited to three colors, which include red, blue, and yellow. This design includes a dispensing and transportation mechanism, thermal management, computer vision, and a sorting device. Each component is crucial to the functionality of the entire system and as shown in figure 1.

Improve:

The system sorts pills by three colors:

Technical principles include computer vision, electromechanical design, electronic circuit implementation, and systems design. The computer vision contains a detection algorithm, which detects when a pill passes through a certain point on the transportation device and analyzes its color. After scanning the color, PySerial is utilized to send information to the electronic circuit implementation to determine the angle for the sorting device. Electromechanical design includes 3D printing and structure creation for dispensing and transporting the pills to the location where the computer vision scans the pills' color. Electronic circuit implementation executes the mechanical movement for the dispensing, transporting, sorting, and thermal management components. Each part working together serves as the systems design, which includes the dispensing mechanism, transportation device, computer vision, PySerial, and sorting apparatus.

This project is meaningful because it reduces the risk of error for important medicines. Its intended application is in an environment with controlled lighting and temperature such as a warehouse or a pharmaceutical facility with the aid of a technician.

Proponents of the pill sorter that relate to electrical engineering include programming algorithms utilizing computer vision, coding firmware for the microcontroller, electrical component implementation, and validation testing of the electromechanical design.

The overall goal of the project is to lower the cost of manufacturing medicine and reduce human errors in the sorting process. A high-level description of the Pill Sorter is to sort pills into separate piles according to their color. This does interact with external systems, primarily the operating technician. System issues pertain to the operating environment because the computer vision is very sensitive to the light within the vicinity.

The technical design objectives include:

- Sorting a pill in less than 65 seconds
- Using less than 60 watts of power
- Achieving an accuracy of 100%
- Portable/Ability to Dissemble

Task	Task Description	Person Responsible
Python / OpenCV	This is the computer vision that contains the detection algorithm and color analysis.	Ted

Hardware Implementation	This is to implement the hardware components such as the MCU, Stepper motors and drivers and power supplies, and coding the firmware that controls these components.	Nicole
Hardware Testing	This is to test the hardware implementation and make adjustments if needed.	Nicole
System Testing	This is the implementation of every component together.	Everyone
3D Printing / Structure Design	This is the structure for the dispensing, transportation, and sorting device	Peter
PySerial	This code provides communication between OpenCV and the MCU.	Peter

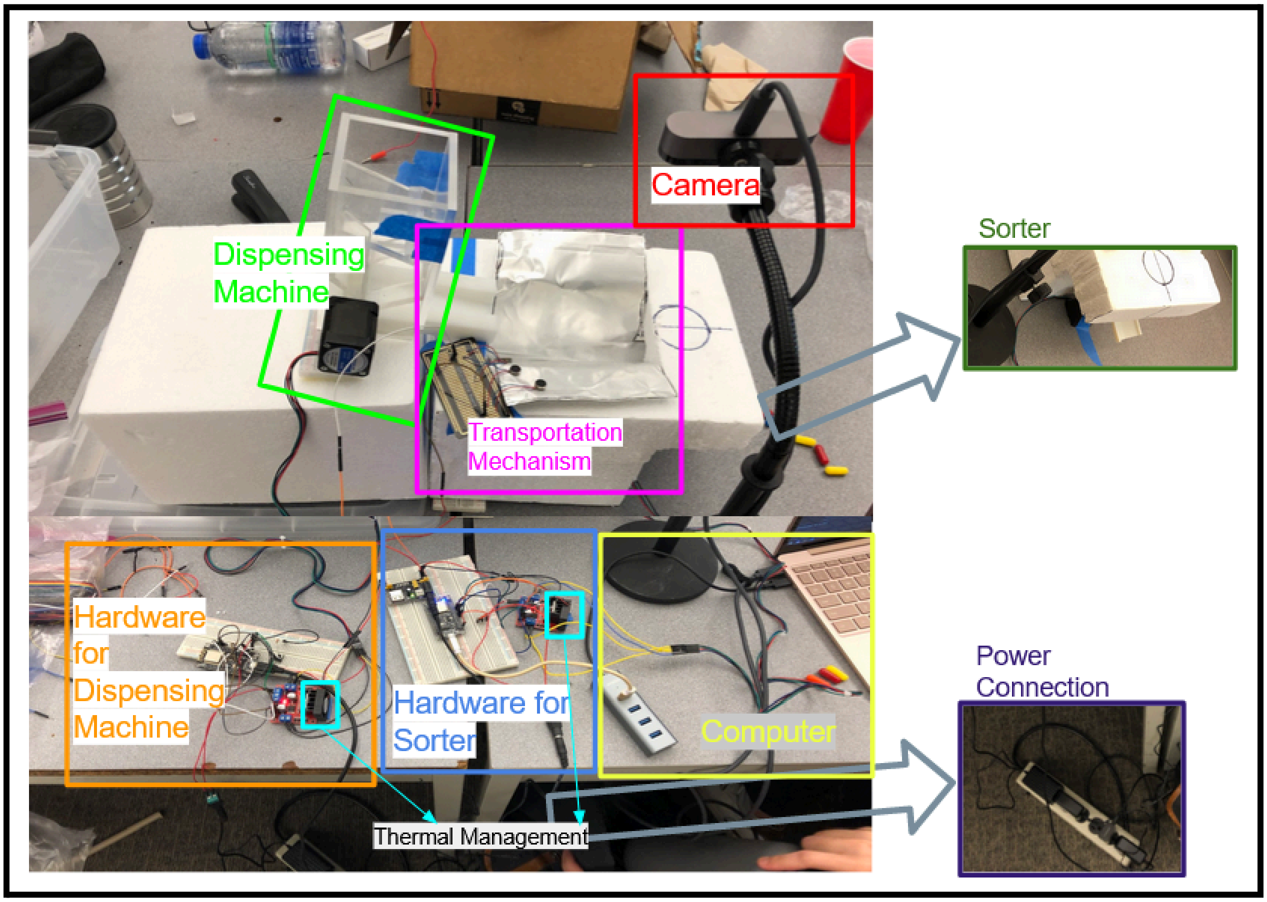

FIG. 1. System Overview

2.2 * Backgrounds and Prior Art

Two existing designs or products include the Capsule Counting Machine SBM-CM12L from Shineben Machinery [3], [4] and an M&M sorting machine based on color [2]. These formed the basis of the Pill Sorter project. The Shineben machine used both shaking platforms and converbelts to sort pills rapidly. This is where the idea of using a conveyor belt and a shaking platform to transport the pills comes from. On the other hand, the idea of sorting by color comes from the M&M sorting machine. The Pill Sorter consumes less power than the Capsule Counting Machine SBM-CM12L, as it uses 600 watts compared to the compact and portable Pill Sorter, which is 53 watts. Another advantage of the Pill Sorter is its low cost due to its material and parts required for the design. However, the shortcomings include speed and scale as it was not feasible to recreate their design given the time and budget constraints.

2.3 * Development Environment and Tools

Software:

- The computer vision was accomplished using Python language and OpenCV libraries such as numpy, serial, time, os, and cv2.
- The embedded systems portion and serial communication are for the sorting mechanism.

Hardware:

- The hardware development environment for the coding ESP8266 is the Arduino IDE with the ESP8266 add-on. ESP826 is used for controlling both the dispensing and transportation mechanism.
- DC Fans for thermal management.
- Shaking motors for transporting the pills.
- Both the DC fans and shaking motors were connected to the same power supply.
- 3D printed the dispensing system and chute for the sorting mechanism.

Test Equipment:

- Keysight DC Power supply to test stepper motors.
- Agilent Multimeter to test voltage for various hardware parts.

2.4 * Related Documents and Supporting Materials

The project complies and uses USB standards. This can be seen with the other sections that describe the setup and details [4].

2.5 * Definitions and Acronyms

USB - Universal Serial Bus

3 * Design Considerations

3.1 * Realistic Constraints

Time, cost, portability, type of pills, time to sort, and originality constrained the project. Originality limited the project's progress, demanding double the testing time. Namely, the dispensing system demanded various prototypes and testing before its final implementation, which went through three prototypes. The shaker system demanded 10 weeks to develop, evolving from a non-functioning conveyor belt. Furthermore, the computer vision's limited scope forced us to use three significantly different colors to avoid aliasing. Although we entertained lasers and a graphical interface after finishing, it interfered with our already working prototype. In terms of the pills, the system only works on the same type of pills (same size and shape but different color).

3.2 System Environment and External Interfaces

In terms of hardware, the system requires three power outlets for the computer, sorting mechanism, and dispensing apparatus. USB is used to connect the hardware and software on the computer. The software functions with libraries and protocols such as OpenCV, numpy, PySerial, and time. The Pill Sorter interfaces with a trained technician and is also sensitive to the lighting and temperature of its environment. As a result, its use case is in a pharmaceutical manufacturing facility.

3.3 * Industry Standards

The industry standard involved in the project includes the Universal Serial Bus (USB) standard. This standard established a specification for cables to provide a consistent connection and communication between separate devices. The project used this by having two main connections to the computer. The first included a Logitech Brio, a camera plugged into the laptop's USB 3.0 port, using a USB-C to USB-A cable. The second connection is a Micro USB to USB connection for the ESP8266 microcontroller. According to IEEE's documentation on USB port and power delivery, a USB 3.0 port can only supply a maximum of 0.9 A at 5 V. This did not have a large impact on the design because the camera did not provide a significant amount of voltage output. Logitech's datasheet does not give any specific measurements, but it was below 0.9A at 5V [4].

3.4 * Knowledge and Skills

The knowledge and skills of each member is listed below:

Peter:

- Prior knowledge:
  - Embedded systems
  - Serial communication
  - Classical mechanics
- Learned knowledge
  - Soldering
  - 3D printing
  - Acrylic manufacturing

Nicole:

- Prior Knowledge
    - C and C++
    - Logic Design
    - Introduction to Embedded systems
    - Sensing and Actuation for Embedded Systems
    - Solid-State Electronics
    - Circuit Analysis
    - Soldering
- Learned Knowledge
    - Power Management
    - Thermal Management
    - Using ESP8266 NodeMCU

Ted:

- Prior knowledge:
    - C and C++
    - Logic Design
    - Introduction to Embedded Systems
    - Sensing and Actuation for Embedded Systems
- Learned knowledge:
    - Python
    - OpenCV
    - Numpy
    - Soldering

3.5 * Budget and Cost Analysis

Testing and feasibility took the most of the budget. Our testing cost soared to about $2500, purchasing various materials, backup components, and consulting services. Oftentimes, materials work in the short run but fail in the long run, and the development team lacked the skill to work with certain components. For instance, we tried foam core but failed to learn how to work with it in time. Consequently, we used Styrofoam and a hot knife to mold the desired structure. Furthermore, although we ordered a batch of power adapters, not all of them met specifications. Since mechanical engineering exceeded our team member's skills, we required consulting services from Peter's personal network with fees of 20 USD to eliminate impractical designs. However, the final material cost ended up at $140.70.

3.6 * Safety

Within testing, the motor driver presented a heat hazard reaching 40 degrees Celsius, but introducing a motor fan and lowering the amperage reduced that risk. Having a cool surrounding environment of 19 degrees Celsius also reduced the risk.

Salt pills were used instead of real pills because they contain biologically hazardous materials which could leak upon breaking in the dispensing system. After serious testing, we determined that the dispensing system was safe, and even in the event of failure, a pharmaceutical professional would know how to manage it. In short, the risk to benefit ratio provided justification for using the system.

3.7 Performance, Security, Quality, Reliability, Aesthetics etc.

The performance and quality demanded the foremost attention due to the rigid requirements. In short, the design needed to work with one hundred percent accuracy. Aesthetics demanded no consideration. Although considerations about aesthetics occurred, time forced us to stick with the prototype.

3.8 * Documentation

Weekly progress reports contained the project's documentation. The weekly reports would serve as technical documentation including procedures, design notes, changes, and updates to the system. Through these reports, team members would communicate and comment on design changes and testing for parts of the system. The report contains pictures and videos of the tests, allowing members to see group progress. The team stored research and design notes in Google Drive. All code and changes were also reported in the same folder. Github houses all the finalized code. YouTube also has documented videos of the final project.

3.9 Risks and Volatile Areas

Real-time latency provided the most hazardous risk. If the system operates too quickly, the computer crashes and requires a hard reset. To mitigate, our system comes with a policy of sorting one pill at a time.

4 * Experiment Design and Feasibility Study

4.1 * Experiment Design

Conveyor Belt Experiment Design

Objective: This experiment tests if it was possible to transport pills on a conveyor belt with a design that did not have an entire rotating pole pulling the material.
Setup: The materials needed were two pieces of cardboard, a PVC pipe, two motors with wheels, and a paper belt. The two pieces of cardboard had a PVC pipe through it and two motors on each side of the cardboard pulling the paper belt as the motors rotated.
Procedure: Setup the materials to match the set up described above. Turn on the motors and observe.
Expected Results of the Experiment: The conveyor belt transport pills smoothly and consistently.
Responsible: Nicole Chin was responsible for designing this experiment and assembled the conveyor belt used for this test. Peter Chau purchased and cut the PVC pipe, and Ted O'Young cut a hole to provide a structure for rotation.

Cloth Conveyor Belt Experiment Design

Objective: This experiment is to test if using cloth instead of paper for the conveyor belt would work, as well as include the adjustments to the structure based on the results of the previous conveyor belt experiment. The test will be verified if the pills can be successfully transported.
Setup: The materials needed were two pieces of cardboard, a PVC pipe, two motors with wheels, and a paper belt. The two pieces of cardboard with the adjusted measurement had a PVC pipe through it and two motors of the cardboard pulling a cloth belt due to the motors rotation.
Procedure: Get the materials. Cut the cardboard and cloth. Sew the cloth into a belt. Setup the materials to match the set up described above. Turn on the motors and observe.
Expected Results of the Experiment: The conveyor belt should transport pills smoothly and consistently.
Responsible: Nicole Chin was responsible for designing this experiment. She also assembled, cut, and sewed the conveyor belt used for this test.

Motor Spinning an off Centered Circle

Objective: This experiment is to test if a motor with an off centered circle hitting paper or cloth will cause the material to shake and transport the pills.
Setup: Have a motor under a flexible piece of material, and in this case, both cloth and paper. The material served as a surface where pills are placed. The motor is spinning an off centered circle made of thick cardboard that hits the piece of material that causes the material to shake. This then shakes and transports the pills.
Procedure: Cut the materials according to their functions. Setup the materials to match the set up described above. Turn on the motors and observe.
Expected Results of the Experiment: The motor should hit and shake the surface and transport the pill.
Responsible: Nicole Chin was responsible for this test including test design, assembly, and getting materials.

Mini DC Vibration Motors on a Pathway

Objective: This experiment is to test if using DC Vibration Motors on a pathway are able to transport pills as this would require less material. Using the DC Vibration Motors would make it easier to have the pills in a single file line and control the speed due to its concave shape and angle.
Setup: The DC shakers attached itself to the aluminum pathway.
Procedure: At first, the DC shakers were attached to a funnel-shaped aluminum device and applied a 5 Volt power, and placed pills on it with an incline.
Expected Result of the Experiment: The pills will move downwards at a steady pace.
Responsible: Peter Chau was responsible for designing this test. Peter Chau found the shakers and glued them onto the sculpted aluminum.

Stepper motor 28BYJ-48 with ULN2003

Objective: This experiment is to test if using the 28BYJ-48 stepper motor would be able to move a channel that guides the pills in the correct direction.
Setup: The 28BYJ-48 (the stepper motor) is connected to the ULN2003 (motor driver), The motor driver is connected to the 5 V power supply and the logic pins are connected to the microcontroller. The motor is coded to turn and point the channel at 90 degrees, 180 degrees, and 0 degrees.
Procedure: Setup the materials to match the set up described above. Code the motor driver using the stepper.h library. Run the code and observe.
Expected Result of the Experiment: The stepper should move to three destinations accurately within 7 seconds.
Responsible: Nicole Chin is responsible for this test.

Stepper motor Nema 17 with L298N
Objective: The experiment is to test the Nema 17 stepper motor and if it is sufficient enough to move a channel that guides the pills in the correct direction. The stepper needs to move to three destinations accurately within a sufficient time frame.
Setup: The Nema 17 (the stepper motor) is connected to the L298N (motor driver). The motor driver is connected to the 12 V, 2 A power supply and the logic pins are connected to the microcontroller. The motor is coded to turn and point the channel at 90 degrees, 180 degrees, and 0 degrees.
Procedure: Setup the materials to match the set up described above. Code the motor driver using the stepper.h library. Run the code and observe.
Expected Result of the Experiment: The stepper should move to three destinations accurately within 7 seconds.
Responsible: Nicole Chin is responsible for this test.

Stepper motor Thermal Management
Objective: This experiment is to test if lowering the current in the supply, as well as including a fan near the heatsink would be sufficient for thermal management. The L298N when driving the Nema 17 would heat up to 50 degrees Celsius.
Setup: The power supply for the driver is lowered to a 12 V, 1 A AC adapter and a small 5 V fan was added near the heatsink of the driver.

Procedure: Setup the hardware just matching the previous test, but switch out the power supply. Run the code and observe. The test runs for 10 minutes.
Expected Result of the Experiment: The heat of the driver should remain under 40 degrees Celsius.
Responsible: Nicole Chin is responsible for this test.

Coordinated Hardware Test
Objective: This experiment is to test all the hardware components before adding it to the system, and to determine if the stepper motors respond with the lasers and switches.
Setup: The hardware includes a laser, a photoresistor, a switch, 5 V power supply, two DC fans, two steppers motors, and 2 L298Ns for the final machine connected to the 2 NodeMCU ESP8266 microcontrollers.
Procedure: The two Nema 17 are connected to the two L298N. L298Ns are connected to the 12 V 1 A power supply and the logic pins are connected to the two NodeMCU ESP8266 microcontrollers. The motor is coded to turn and point the channel at 90 degrees, 180 degrees, and 0 degrees depending on the input from the photoresistor and switch. The two DC fans directly connect to the 5 V power supply. Turn on and plug all of the power supply. Run the code, and test all possible with different input.
Expected Result of the Experiment: The hardware should work as a system and respond properly to the input from the switch and photoresistor.
Responsible: Nicole Chin is responsible for this test.

Color Algorithm Model
Objective: The objective is to obtain an accurate scan of the pills' hue, saturation, and brightness value (HSV). This information will determine the color using HSV boundaries adjusted for its current environment.
Setup: The hardware includes a laptop and a computer camera. The software is run on the laptop.
Procedure: Test the software in different lighting conditions and observe the outputs. Adjust the boundaries if the scans are inaccurate.
Expected Result of the Experiment: The reading of the pills' color is the same regardless of the system's surroundings.
Responsible: Ted O'Young is responsible for developing and testing this implementation of the project.

4.2 * Experiment Results, Data Analysis and Feasibility

Conveyor Belt Experiment Design
Results: The conveyor belt pulled the paper, but the cardboard needed to be raised 5.75 inches off the table, and the belt needed to be 25.2 inches long and 3.75 inches wide, as it dragged on the table. The paper belt got stuck as the material is too stiff. There was a small issue as the speed could not be controlled, but the conveyor belt continued to develop while other alternatives are being researched. Furthermore, the pills could not reliably move in a single file line. This experiment determined that it is not feasible to make a conveyor belt using the current design. Nicole Chin was responsible for observing and analyzing the test results.

Cloth Conveyor Belt Experiment Design
Results: The cloth conveyor belt ran more smoothly than the paper conveyor belt and it was able to transport the pills. In the event that the project needs a conveyor belt, this will be used. Since it is still using the same motors as the previous conveyor belt, the speed cannot be controlled, and other

alternatives are still being researched. This experiment determined that it is more feasible to use cloth than paper for the conveyor belt. Nicole Chin was responsible for observing and analyzing the test results.

Motor Spinning an off Centered Circle
Results: The motion of the uneven circle hitting the paper had no effect. The paper material is too rigid to shake the pills. However, when cloth is substituted for paper, the pills are capable of moving, but at an uneven rate. The pills moved at varied speeds throughout the surface, which caused the exit chute to be jammed. Other solutions to transport the pills need to be explored, as this proved to be unsuccessful. This experiment determined that it is not feasible. Nicole Chin was responsible for observing and analyzing the test results.

Using mini DC Vibration Motors on a pathway
Results: The DC Vibration Motors were able to transport the pills on the aluminum pathway at a sufficient speed when the pathway was angled at around 20 degrees from the table or platform. This showed that it is feasible to to use this method to transport the pills and that the final project will be using this method. This experiment determined that it is feasible to use DC vibration motors for the pathway. Peter Chau was responsible for observing and analyzing the test results.

Stepper motor 28BYJ-48 with ULN2003
Results: The stepper motor turned with moderate accuracy. It worked 80% of the time, but failed 20% of the time with 10 to 20 degrees of error. The speed of the stepper motor was also too slow for the purposes of the project, so other methods and parts will need to be researched. This experiment determined that it is not feasible to use the 28BYJ-48 stepper motor with a ULN2003 driver for the purposes of the project. Peter Chau and Nicole Chin were responsible for observing and analyzing the test results.

Stepper motor Nema 17 with L298N
Results: The stepper was able to turn to the three destinations accurately, and the speed of the stepper motor was sufficient for the project. This experiment determined that it is feasible to use the Nema 17 stepper motor with a L298N driver for the purposes of the project. Peter Chau and Nicole Chin were responsible for observing and analyzing the test results.

Stepper motor Thermal Management
Results: The L298N heatsink was able to stay below 35 Celuis after the stepper was running accurately for 10 minutes. This experiment determined that it is feasible to use a DC fan to help dissipate heat. Nicole was responsible for observing and analyzing the test results.

Coordinated Hardware Test
Results: All of the components work together and respond in a timely manner. This experiment proved that it is feasible to add this system to the final machine, but some of the parts such as the lasers were redundant. This experiment determined what hardware would be implemented in the final project. Nicole was responsible for observing and analyzing the test results

Color Algorithm Model
Results: Under lights with a strong yellow tint, the computer vision reads red pills as orange. On the other hand, if the lights were too dim, the yellow pills would have an accurate scan 5/7 times. Adjusting the

HSV boundaries would fix these two issues. Additionally, the computer scans the same color multiple times for the same pill. As a solution, time.sleep() was implemented to ensure sufficient time has elapsed between scans. Ted O'Young was responsible for observing and analyzing the test results.

5 * Architecture and High Level Design

5.1 * System Architecture and Design

The system consists of four subsystems, demonstrated in Figure 5.1. The hopper mechanism, the shaker mechanism, the computer vision, and the sorter operate in tandem to receive, scan, and sort a pill per command signal. The hopper system consists of a stepper motor dispensing one pill per every ninety degree rotation with a push of a button. The shaker mechanism consists of an aluminum cut valley sloped between 15 to 20 degrees with four miniature DC voltage sources, operating at all times. The computer vision consists of a camera which sends information continuously to a computer to recognize the color of the pill on the shaker. Finally, the sorter consists of a 3D printed slide attached to a stepper motor which receives a signal from the laptop turning to a color-associated position. In total, the process goes in this sequence: user places one pill into the dispenser, presses a button to dispense onto the shaker which moves the pill downhill where the camera detects the color of the pill and sends a signal to the stepper motor to relocate to the correct position. Finally, the pill falls off the shaker and onto the sorter and slides down to the properly assigned batch.

To justify the architecture, minimal connectivity between subsystems allowed for robustness and consistency. Throughout development, uncoordinated timing within the system presented the problem of unpredictability. Each subsystem must behave harmoniously to navigate the pill from entry to destination. The hopper system acted as a separate system with independent hardware. For the sorter to react in time, the pill must move slowly enough for the computer vision system to read and to actuate the motor before the pill lands on the sorter. However, the system possesses a one pill at a time policy which the user must obey.

Furthermore, the latency and coordination of microcontrollers and the primary laptop demonstrated a cascading mis-timing hazard. Although in theory the microcontroller should act within a certain time frame, chaotic behavior of the primary laptop presented the risk of missing important deadlines which would then miscalibrate the state machines forcing the user to hard reset the system. In testing, the problem arises more than usual. To circumvent, the chosen slope of the shaker slowed down the pill's motion to a certified rate to guarantee functionality.

Ted O'Young designed and performed the computer vision feedback system. Peter Chau designed the serial communication, hopper mechanism, shaking mechanism, and sorting mechanism, also 3D printed the sorter and manufactured the acrylic hopper. Nicole Chin coded firmware, and she designed and implemented the hardware system and thermal management, integrating a fan to cool the hopper to a sustainable temperature.

In Figure 5.1, the block diagram demonstrates the four subsystems operating in tandem.

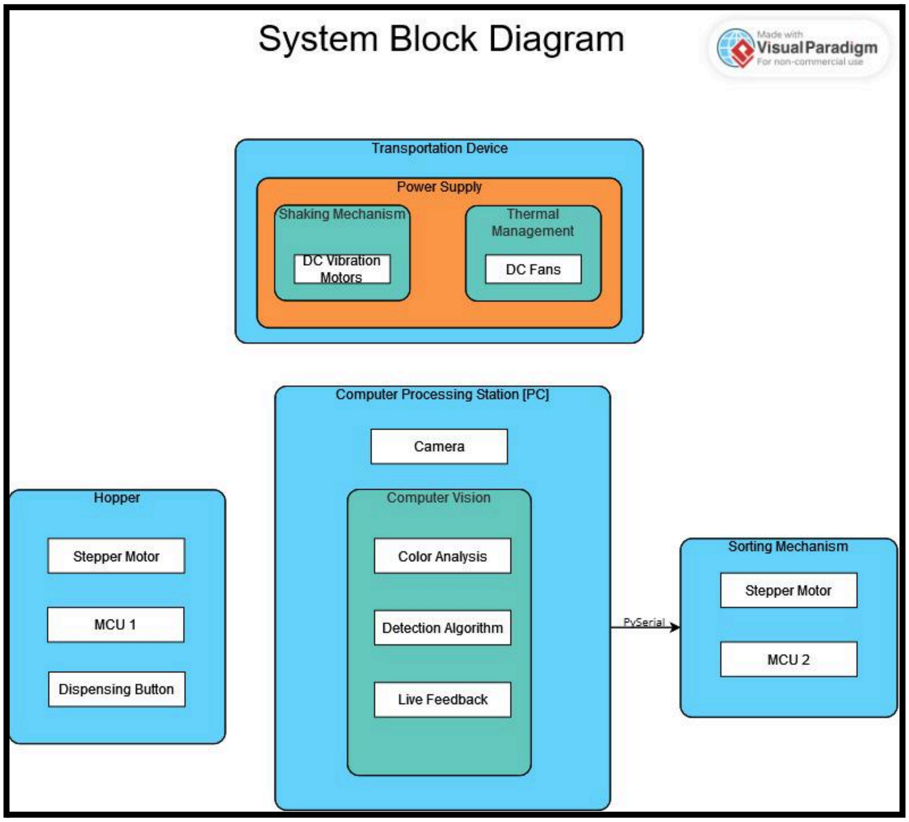

FIG. 2. System Block Diagram

5.2 * Hardware Architecture

In FIG. 3, the hardware architecture has three main parts. The sorter in the bottom left has a connection with the laptop and the NodeMCU ESP8266. The NodeMCU ESP8266 connects to the L298N. The L298N connects to the 12 V 1 A power supply and the Nema 17 stepper motor. The system on the top shows the hopper system hardware. The NodeMCU ESP8266 connects to a USB 5 V 0.9 A power supply and to the L298N. The L298N connects to the 12 V 2 A power supply and the Nema 17 stepper motor. The bottom system shows the DC vibration motors for the shaking mechanism and the two DC fans for thermal management. The DC vibration motors and DC fans are directly connected to a 5 V 700 mA power supply. The 5 V power supply is connected to a 12 V 1 A power supply.

Nicole is responsible for the hardware architecture.

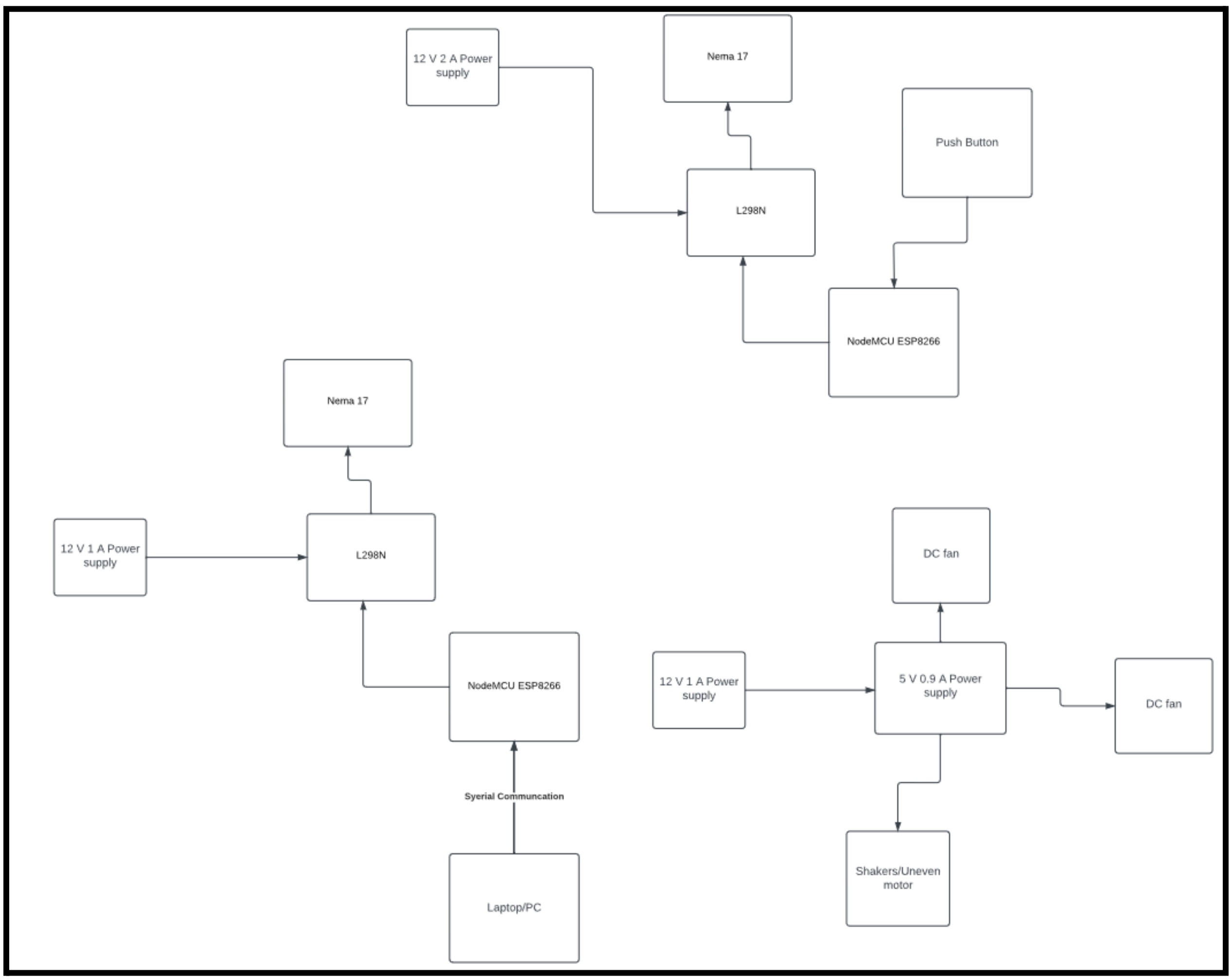

FIG. 3. Hardware Block Diagram

5.3 * Software Architecture (only required if your design includes software)

The software architecture of the system can be seen with FIG. 4. When the code compiles, it starts with the camera turning on to provide feedback to the technician. Afterwards, a detection algorithm is constantly polling a certain point on the shaking mechanism to determine if a pill is present. Once a pill passes through this point, the color analysis will determine the pill's HSV and send a PySerial command to the stepper motor. Once the stepper motor receives the instruction, it will rotate accordingly. After the stepper motor rotates, the detection algorithm runs again, and the cycle repeats. This shows how the software is capable of operating indefinitely, as reliability is a crucial factor for this system. This statement can be supported by our rigorous testing. When the project was presented during Open House, the system functioned flawlessly throughout the entire four hours.

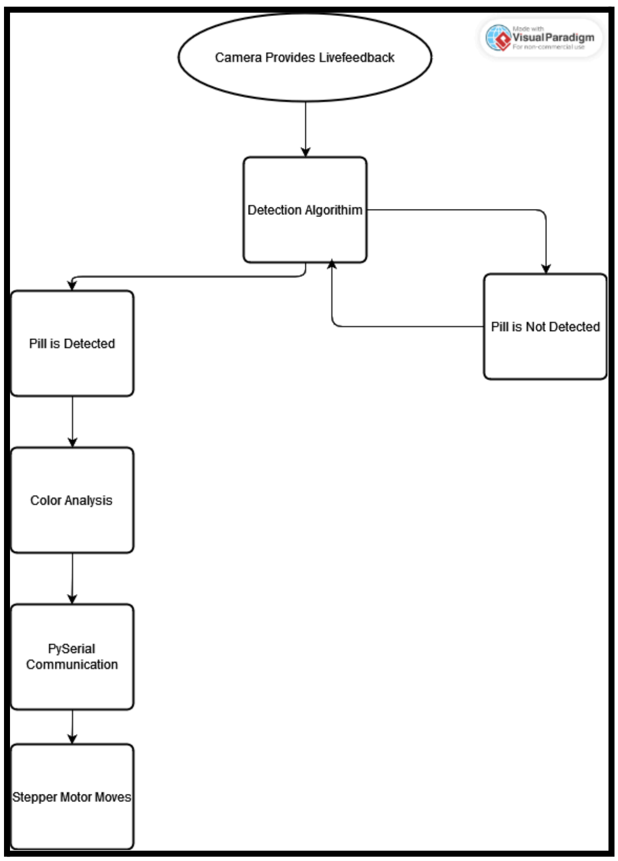

FIG. 4. Software Block Diagram

Ted O'Young performed the live feedback, detection algorithm, and color analysis. Ted O'Young researched, tested, and programmed the software from start to finish. Peter Chau performed the serial communication, where he researched timing protocols and tested the functionality of the mechanism.

5.4 * Rationale and Alternatives

OpenCV libraries were used as a recommendation from Professor Chomko and Chris for computer vision. After implementing the software and running tests, OpenCV was sufficient for the project's use case. However, if more time was given, TensorFlow libraries would have been a better alternative because it utilizes a Graphics Processing Unit (GPU) that would be able to handle the real-time calculations more efficiently than a Central Processing Unit (CPU) because of its parallelized firmware. This setup requires a NVIDIA GPU as TensorFlow operates using Compute Unified Device Architecture (CUDA). Given that the laptop used in the project does not have a GPU, a computer or server with one would be necessary. According to NVIDIA's Tensorflow Framework, CUDA 7.5 or CUDA 8.0 for Pascal GPUs are the minimum requirements [4]. There are two practical options to implement this: build a computer that fits these conditions or access UCR's Bender Server that runs four 2070 Supers in Scalable Link Interface (SLI). For the first option, the software would be capable of running locally, making it easier to develop and test. On the other hand, using UCR's server would be more cost effective, but has certain challenges. A request to install Tensorflow libraries needs to be sent to the administrator of Bender. Additionally, VIM, which is a text editor in Unix, would be the platform to develop the software. This may be problematic because the server's resources are shared with the engineering students accessing it. If another student compiles code that causes memory leaks on the server's GPUs, it will affect the performance and stability of the project.

The hopper system presented the most variation due to the uncertain nature of the pill’s motion. Every design intended to dispense a select amount of pills in a single file line. At first, the design team tried a funnel with a spinning mouth, but it failed. The former held a greater number of pills, but became jammed with a large number of pills stacked like bullets. The design team settled with the latter to ensure functionality.

The conveyor system to move the pill went through two phases: the conveyor belt and the aluminum shaker. The former failed since the speed of the belt was unable to be adjusted, and the belt moved at an insufficient speed. Therefore, the conveyor belt’s motion did not work for a clean computer vision scan. The latter allowed for smooth, consistent, and controlled motion.

The computer vision software went through various versions before settling on a three-color recognition system. The control system consisted of a finite state machine which switched to a respective color upon processing the information from the camera. However, the accuracy of the system presented the greatest challenge. Although color on a computer consists of a linear combination of hue, saturation, and brightness, the kelvin intensity shifts the color of the pill. For instance, red became orange under certain lighting conditions. The design team tried to adjust the environment, but failed to deliver a consistent read. Consequently, the final design consisted of choosing three significantly different colored pills to guarantee accurate reads under all lighting conditions.

The camera choice presented little consideration since the algorithm operated on minimal resolution requirements. The design team used a 4K camera in Peter’s possession.

The shaker design used the principle of procession to vibrate the aluminum foil. Specifically, the thin material with the DC motors allowed the system to control the friction of the ramp depending on the system. In totality, the pill could “slide” upon a DC signal command.

In terms of the hardware, the 28BYJ-48 stepper motor with ULN2003 motor driver used only 5 V of power and the parts were inexpensive, but these parts were not fast or accurate enough. The Nema 17 stepper motor with L298N motor driver used 12 V 2 A. However, these alternative parts moved accurately and quickly, so the final hardware system uses the Nema 17 stepper motor with L298N motor driver. Due to the L298N motor driver heating up, the final design includes two DC fans to help with thermal management. Early development used the Arduino Uno, but the final hardware system used the NodeMCU ESP8266 microcontroller. The NodeMCU ESP8266 is smaller and costs less. Also, Professor Chomko recommended changing the Arduino Uno.

For the hardware system, the initial approach includes one microcontroller, a laser system, and a switch for the entire system. This approach complicates PySerial as this means that the microcontroller will have to receive information from PySerial, but also needs to give instruction to the computer because information can only be sent if the switch is on. This would mean that the microcontroller would need to receive information from two different places, and that the computer acts as a partial receiver. Due to this complexity, the final design has two microcontrollers with one receiving instructions from the computer. The other is its own system and has a push button that runs the hopper. Also, The push button replaced a switch, as it was smaller and more inexpensive. The laser system to detect a pill's location was taken out because of its redundancy in functionality, as the computer vision can detect when a pill passes a certain point on the transportation device. The final machine uses two NodeMCU ESP8266 microcontrollers and two L298N motor drivers.

6 Data Structures (include if used)

N/A

6.1 Internal software data structure

6.2 Global data structure

6.3 Temporary data structure

6.4 Database descriptions

7 * Low Level Design

7.1 **Module: Transportation and Thermal Management*

The hardware for the transportation and thermal management module is in Figure 7.1.

This module has a 5 V power supply that is connected to DC vibration motors and DC fans. The 5 V power supply is connected to a 12 V power source. Nicole Chin is responsible for putting this module together and for the DC fans. Peter Chau was responsible for finding the DC vibration motors.

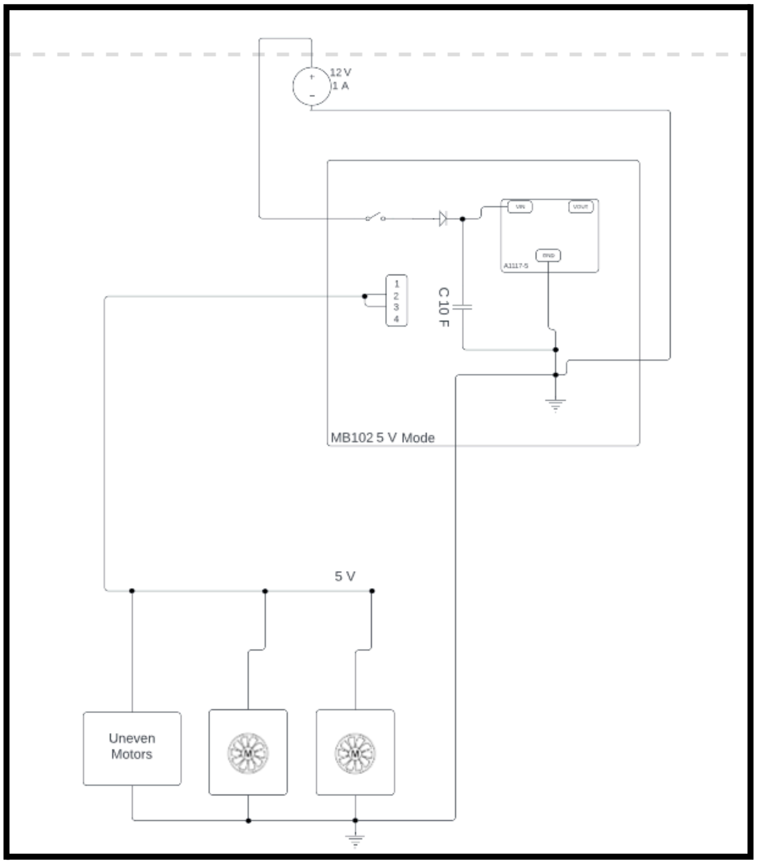

Figure 7.1. Hardware for Transportation and Thermal Management

7.1.1 Processing Narrative for *Transportation and Thermal Management*

The flowchart of the Transportation and Thermal Management is in Figure 7.2.

All the DC vibration motors and DC fans are directly connected to the 5 V power supply that has a pushbutton switch. To turn this module on, press the switch after the 12 V power source is connected. To turn it off, press the switch again.

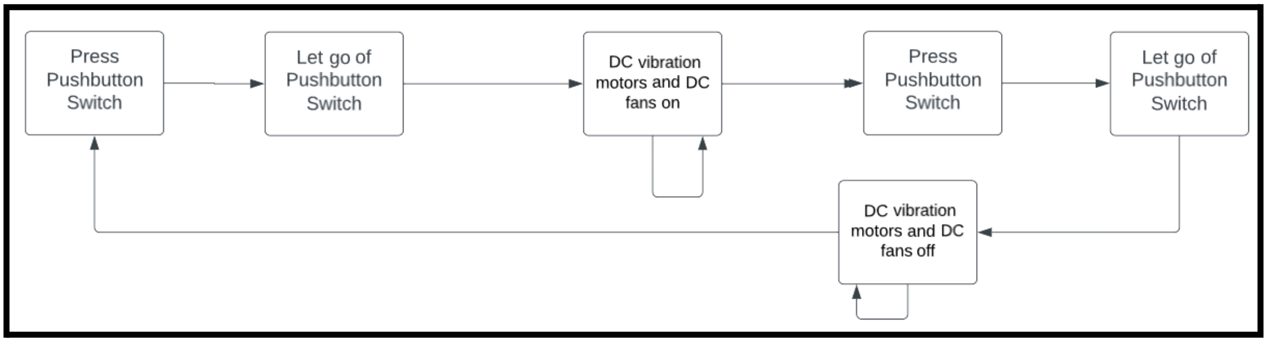

Figure 7.2. Flowchart for Transportation and Thermal Management

7.1.2 *Transportation and Thermal Management* Interface Description

This module receives pill from the hopper module, and provides a stable runway for a clean scan for the computer vision system in the sorting mechanism module. Ultimately, the transport module sends the pill to the sorter.

7.1.3 *Transportation and Thermal Management* Details

This module has a 5 V power supply that is directly connected to 2 DC fans and 5 DC vibration motors. The DC fans provide active cooling to prevent the microcontrollers from overheating. As for the 5 DC vibration motors, 3 were placed on the left side and 2 on the right. This allowed the aluminum foil structure to shake at a consistent and even rate. As a result, this led the pills to move in a ceaseless motion, which helped with the detection algorithm and color analysis. However, there are some limitations due to logistical constraints. As mentioned in section 5.4, if TensorFlow was utilized, then the computer vision would not depend on the shaking mechanism to transport the pills in an uninterrupted and consistent manner. The computer system would be computationally powerful enough to read pills that move at an erratic motion due to its much higher polling rate. The 5 V power supply takes in 12 V 1 A from a power source. This module performs sufficiently for the purposes of the sorter.

7.2 **Module: Hopper System*

The hardware for the Hopper system module is in Figure 7.3.

The module has a stepper motor, motor driver, a button, and a microcontroller. The hopper is made out of acrylic and 3D printed parts. The microcontroller is connected to a 5 V source through USB and the motor driver is connected to a 12 V 1 A source. Peter Chau was responsible for the acrylic and 3D printed parts, and Nicole is responsible for the hardware and firmware.

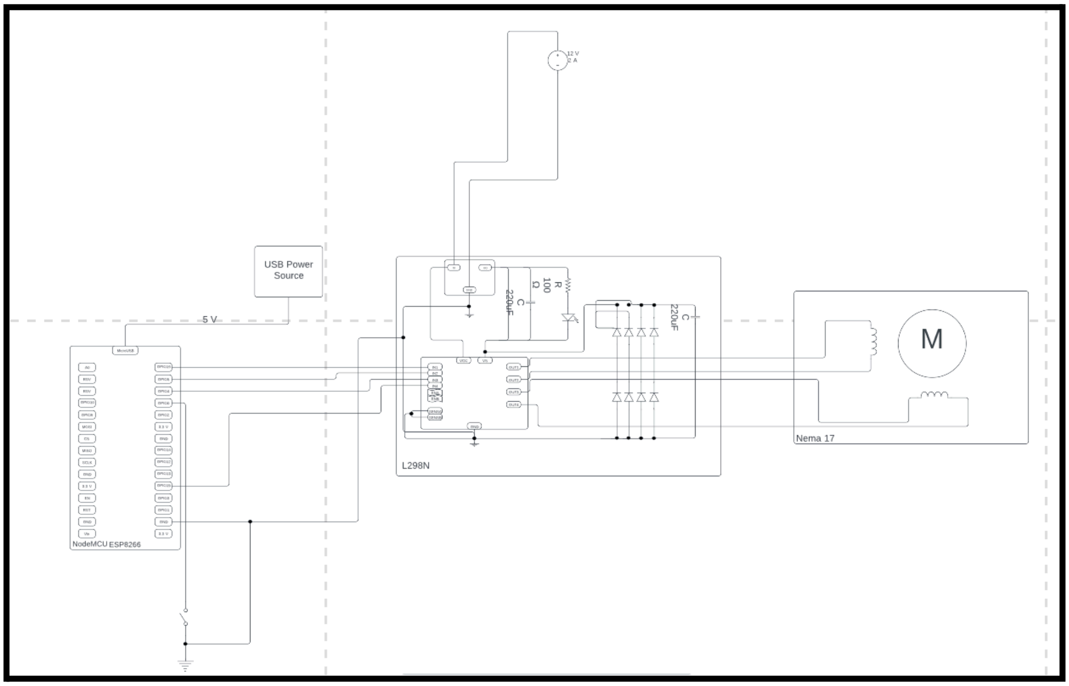

Figure 7.3. Hopper System

7.2.1 Processing Narrative for *Hopper System*

The flowchart of the Hopper System is in Figure 7.4.

The microcontroller should be connected to the 5 V source through USB and the motor driver should be connected to a 12 V 2 A source. The microcontroller receives a button-press signal to command the stepper motor to rotate ninety degrees.

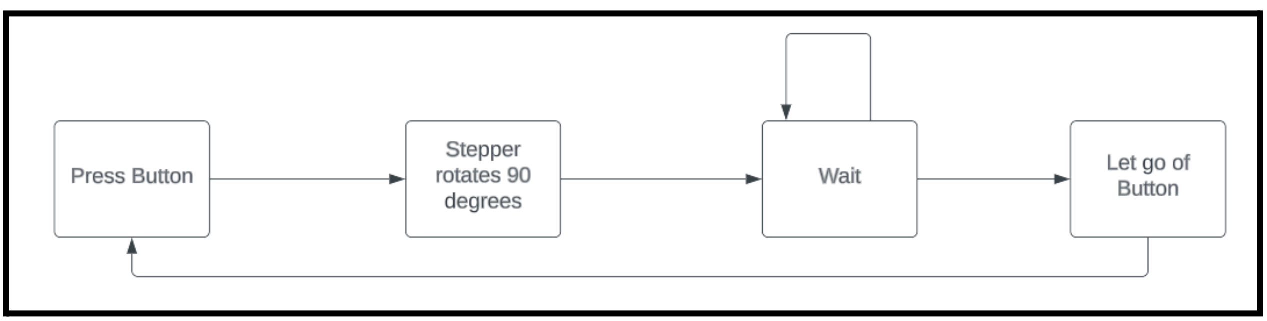

Figure 7.4. Program flowchart for Hopper system

7.2.2 *Hopper System* Interface Description

This module interfaces with the Transportation and Thermal Management module as the fans from the Thermal Management system cool down the motor driver in this module. The hopper system interfaces with the transportation system as it drops the pill in the transporting pathway.

7.2.3 *Hopper System* Details

This module uses the ESP8266 microcontroller, a button, a Nema 17 stepper motor, and a L298N motor driver. All of these parts meet the requirements for the project's needs, and the hopper performed sufficiently. The hopper has a limitation for the number of pills that can be loaded into the hopper. The hopper can not dispense when there are more than two pills loaded into the system.

7.3 **Module: Sorting Mechanism*

The hardware for the sorting mechanism is in Figure 7.5.

The hardware for the module consists of stepper motor, motor driver, a computer, camera, and a microcontroller. This is the module where the stepper sorts the colored pills. Ted O'Young is responsible for the computer vision code. Peter Chau was responsible for the PySerial code, and Nicole Chin was responsible for the hardware.

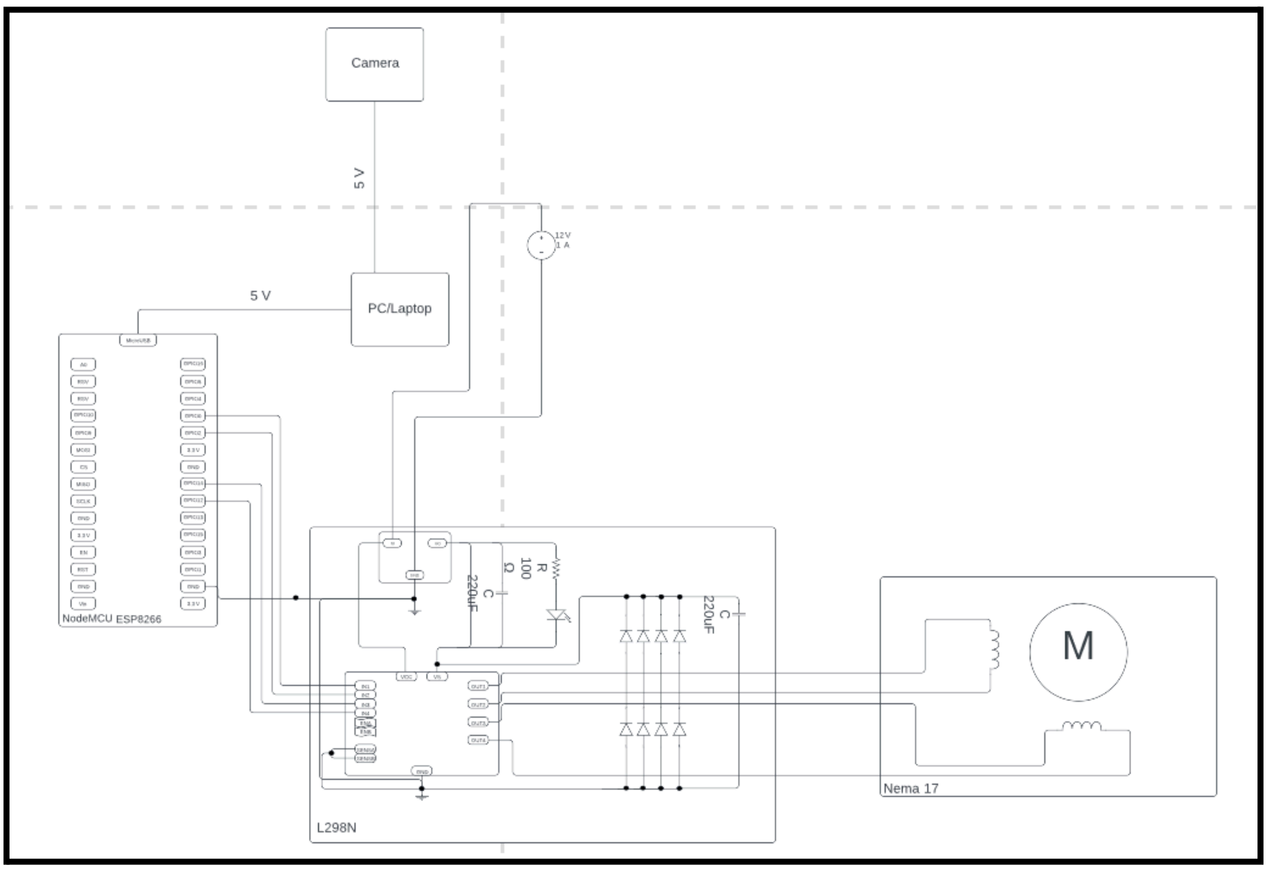

Figure 7.5. Sorting Mechanism Hardware

7.3.1 Processing Narrative for Sorting Mechanism

The flowchart of the sorting mechanism is in Figure 7.6.

The microcontroller and camera operate with the computer using USB protocol. The motor driver seen in FIG 9 uses a 12 V 1 A source. Once the camera scans the pill's color using the computer vision code, it communicates this information to the microcontroller using PySerial. Depending on the signal, the stepper motor of the sorting mechanism will rotate accordingly.

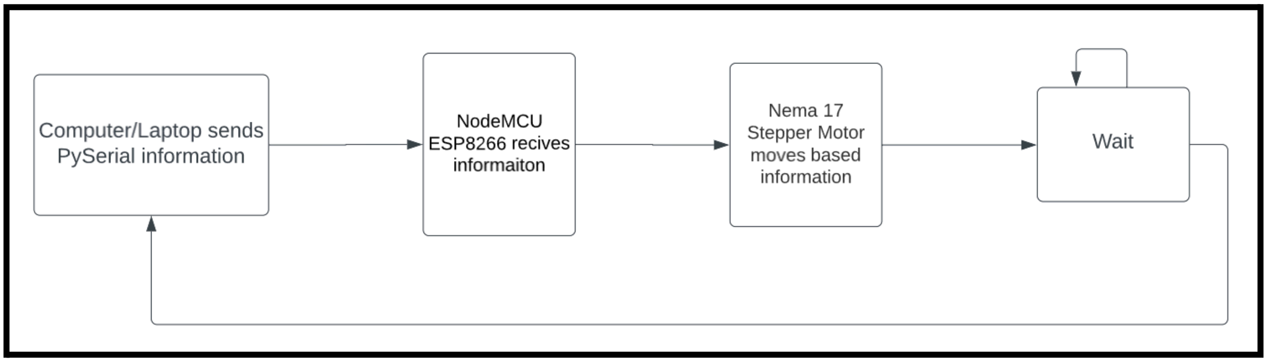

Figure 7.6. Flowchart for Sorting Mechanism

7.3.2 *Sorting Mechanism* Interface Description

This module interfaces with the Transportation and Thermal Management module as the pills move along the Transportation system and the camera scans the color once it reaches the detection algorithm. One of the DC fans from the Thermal Management system cools down the motor driver in this module.

7.3.3 *Sorting Mechanism* Processing Details

The module consists of a Nema 17 stepper motor, L298N motor driver, a computer, web camera, and the ESP8266 microcontroller. The color detection program uses OpenCV libraries written in Python. Once the program detects the color, it sends a signal to the sorter stepper motor to steer the pills to the correct pile. This module meets the requirements for the project's needs.

8 * Technical Problem Solving

8.1 * The Color Detection Problem

The computer vision had issues getting an accurate scan of the pill's color. Due to the kelvin of the surrounding, it would alter the hue, saturation, and brightness (HSV) of the computer's reading. For example, if the lighting was yellow, it would detect red pills as orange. This issue was identified by Ted O'Young.

8.2 * Solving the Color Detection Problem

When the pills initially were red, orange, and yellow pills, red scanned as orange and vice versa. To fix this, the design team painted the shaker mechanism green to calibrate the lighting. However, once the paint dried, there was too much friction between the pill and the surface, and it failed to provide smooth transportation. Consequently, the design team swapped orange for blue. With each pill having a vastly different HSV value, the computer accurately scanned them once the lower and upper bounds were fine tuned. Learning how to program computer vision software facilitated the problem solving. Ted O'Young resolved this issue.

8.3 * The Stepper Motor Accuracy Problem

The accuracy of the stepper motor would lose accuracy over time when given instructions by the PySerial communication for an unknown reason. When doing a system test, all team members identified the issue.

8.4 * Solving the Stepper Motor Accuracy Problem

This issue likely resulted from a slow build up of leftover signals into the program's cache which carried over into future commands, but resetting the microcontroller manually by pressing the reset button allowed the stepper motor to move accurately since it cleared the buildup. To fix this issue, a reset was coded into its software after each movement. In this process, group members learned about the command for a software reset, and to use the simplest solution as long as it works. Learning to examine the code's cache within a code allowed the design team to debug the code properly. The issue was resolved by Peter Chau.

8.5 * Logic Bug Problem

The code for the microcontroller that was reading the serial communication had a logic bug, and therefore, that stepper would not move. Ted O'Young identified the serial communication logic bug, and Nicole Chin identified the delay issue.

8.6 * Solving the Logic Bug Problem

The serial communication contained a logic bug. Although coded as a branch statement, it appeared to read the signal instantaneously due to the fast operating speed. There was also a small delay missing from the python code with computer vision as it was reading one long string instead of separate strings. The branch statement was changed to two branch statements with a three second delay. Learning to debug in an efficient manner helped to resolve this problem. All the team members identified the issue and solved it.

8.7 * Thermal Management Problem

The heatsink attached to the motor driver presented a hazard as it reached 40 degrees Celsius. This issue caused safety concerns as touching the heatsink on the motor driver can burn skin. The issue was identified by Nicole Chin.

8.8 * Solving the Thermal Management Problem

When the drivers ran in real time, they reached 40 degrees Celsius. Two fans were added to dissipate the heat from the heat sink and for one of the motors the current supply was lowered from 2 A to 1 A. The part maximum current draw is 2 A, but the stepper motor for the sorter did not need as much torque so that current for that one was lowered. For this issue, looking at how a computer is able to manage its heat was used as a basis for the design change. Lowering the temperature of the surrounding environment to 19 degrees Celsius helped with managing the heat. The motor drivers stayed below 35 degrees Celsius. Learning safety procedures and potential hazards avoided any further complications. Nicole identified and fixed the issue.

8.9 * The AC Adapter Problem

The stepper motor stopped working after reassembling the full machine. This motor was not responding correctly to the serial communication and all the team members identified the issue.

8.10 * Solving the AC Adapter Problem

The serial communication worked, but the stepper motor failed to respond to a command signal. This meant that the issue was hardware related. Rewiring did not resolve this issue, even after trying multiple variations. Both AC adapters had the same specifications, 12 V 1 A, but only one worked, meaning that the issue was a defective AC adapter. With this problem, the team learned to check everything as the AC adapter was not a suspected problem. Learning to be thorough and testing every possibility was necessary to solve this complication. The issue was identified by the team, and it was fixed by Nicole Chin.

9 User Interface Design

9.1 Application Control

The user has to press a button after loading into the hopper. There is also a GUI for the user to see. The screen is not interactive. The GUI only provided a live update on the amount of pills for each color, as well as a total count.

9.2 User Interface Screens

A screen transition diagram or table can optionally be created to illustrate the flow of control through the various screens.

This does not have to be actual screenshots since they have not been programmed. They can be powerpoint drawings or mockups created in Visual Basic or some other rapid GUI-building tool. Figure 9.1 shows the proposed GUI.

Figure 9.1. GUI

This GUI would provide a live update whenever a pill of a certain color would be detected. However, this was not shown off in the final demonstration due to computational limits. Whenever the GUI was working in tandem with the color analysis software, it would cause the live feedback to be very laggy.

10 * Test Plan

10.1 * Test Design

Sorter Hardwear (repeated for i = 20 times)
Objective/Function: This test is to see if the stepper motor for the sorter could consistently turn accurately when receiving PySerial information, which will be part of achieving the 100% accuracy for sorting the pills.
Experiment setup: The microcontroller sends a command signal to the stepper motor sorter, rotating 45 or 90 degrees in both directions.
Experiment procedure:

- Wire the stepper to the microcontroller
- Mark a starting place on the stepper using tape (the piece of tape serves as a reference point)
- Use the tape to see if the stepper motor correctly reaches the reference point to collect data
- Repeat the two steps below 20 times
    - Manually type and send the PySerial code information to the microcontroller
    - Compare the movement of the stepper to the taped reference points

Expected results: The stepper turns to the reference points accurately after manually sending the Pyserial information 20 times randomly in the 45 or 90 degrees clockwise or counterclockwise direction. Peter Chau, Nicole Chin, and Ted O'Young are responsible for the setup and procedure of this experiment.

Computer Vision Test and sending PySerial (repeated for i = 18 times)
Objective/Function: This test served to see if the computer vision software would send the correct PySerial communication instructions to the microcontroller and have the stepper respond. This contributes to achieving the 100% accuracy for sorting the pills.
Experiment setup: The camera, stepper motor for the sorter, and the machine with the pathway is set up.
Experiment procedure:

- Set up the systems mentioned in the experiment setup
- Drop each of the different colored pills to test all 9 cases twice
- Examine the results

Expected results: The computer vision would accurately recognize the pill color and send the correct instructions though PySerial communication. The stepper connected to the microcontroller would accurately turn to that color's pill. Ted O'Young is responsible for this experiment.

Hopper Test (repeat for i = 10 times)
Objective/Function: This test is to see if the hopper was working correctly, and if it is able to dispense pills
Experiment setup: The 3D/acrylic case is assembled, and the stepper and button is wired to another microcontroller.
Experiment procedure:

- Place and tilt the Hopper at 60 degrees
- Load a pill into the dispensing mechanism
- Press the button
- See if the pill dispensed

Expected results: The hopper is able to dispense the pill, and the stepper motor for the hopper turns accurately. Peter Chau and Nicole Chin are responsible for this experiment. Peter Chau designed and built the 3D/acrylic, and Nicole Chin coded and designed the hardware.

Final Test of Everything (repeated for i = 18)
Objective/Function: This test is to see if all systems of the machine would work together and accurately dispense and sort the pills. This is also to test the amount of time it takes to sort a pill. This contributes to achieving the 100% accuracy for sorting the pills.
Experiment setup: The hopper, camera, stepper motor for the sorter, and the machine with the pathway is set up.
Experiment procedure:

- Set up the systems mentioned in the experiment setup
- Drop the different colored pills in the hopper
- Test all 9 possible cases twice

Expected results: The system is able to sort the pills accurately into different piles, and the time to sort a pill is on average less than 65 seconds. Peter Chau, Nicole Chin, and Ted O'Young are responsible for this experiment.

10.2 * Bug Tracking

A document on the shared Google Drive recorded bug tracking.

Defect: Stepper motor for sorter not receiving PySerial information. Assigned to all members to fix and investigate
Defect: Stepper motor for sorter losing increasingly more accuracy over time. Assigned to all members to fix and investigate
Defect: Computer vision scanning red and orange incorrectly. Assigned to Ted O'Young to fix and investigate

10.3 * Quality Control

For quality control, all 9 possible test cases, which are based on the three different colored pills, were tested for the final project, and issues during the test were logged.

Test 1: Modules Sorter Hardwear
The initial issue was the logic error in the code, which was fixed as described in the technical problem solving section. After fixing the logic error in the code, the stepper would turn accurately the first time, but it would lose accuracy over time. This issue with the stepper motor was not reaching the reference points accurately because the turns were 0-5% away from the correct position. Over time the destination of the stepper would be completely off, as all the small errors added up. To fix this issue, a reset for the microcontroller was coded into its software after each movement.

10/20 Test Cases Failed Occurred on: February 25
Fixed on February 26, so 20/20 cases passed

Test 2: Computer Vision Test and sending PySerial

Given the lethal consequence for failure, quality control remained first priority. Initially, when the machine received red, orange, and yellow pills as an input, it scanned 8/9 test cases accurately. However, after several fixes aforementioned in the technical problem solving section, the design team concluded that the project works with 100% accuracy.

1/9 Test Case Failed Occurred on: March 4th

Fixed on March 8th, so passed 9/9 cases

10.4 * Identification of critical components

Serial communication acted as the most critical component since it linked and coordinated the system. Sections 8.1 and 8.2 identified the issues and explained the solution to the problems.

10.5 * Items Not Tested by the Experiments

No items remain untested.

11 * Test Report

11.1 * Test 1: Modules Sorter Hardwear (repeated for i=20 times)

1.Test results and Responsibility: All of the members set up, ran and observed this test together. The stepper motor turned when the PySerial was sent to the microcontroller, but the accuracy of the turns were about 0% - 5% away from the correct position. This is not an issue for our purposes at the beginning, but over time the destination of the stepper would be inaccurate, as all the small errors cumulatively added up.
2.Comparison: The stepper motor needed to turn to the reference points accurately each time, but there were small errors that added up. This made the stepper inadequate for the project.
3. Analysis: Since the motor did not match with the expected results, corrective actions needed to be taken in order to fix this issue.
4.Corrective actions: The stepper motor would turn accurately the first time. However the stepper motor lost accuracy over time. To fix this issue, a reset for the microcontroller was coded into its software after each movement. After this, the test was repeated again for another 20 times, and this reached the reference points with 100% accuracy.

11.2 * Test 2: Computer Vision Test and Sending PySerial (repeated for i=18 times)

1.Test results and Responsibility: All of the members set up, ran and observed this test together. This test was done with all of the members. The computer vision had difficulty recognizing red and orange, as the lighting was inconsistent. Because of this the stepper would not always change in response.
2. Comparison: This test showed that the color direction failed occasionally as it had difficulty recognizing the difference from red and orange. This was due to uncontrolled lighting, shadows, and reflections.
3. Analysis: The stepper correctly responded to color that the computer vision was recognizing, but it was not always recognizing the correct color. This was due to the close HSV (hue, saturation, and brightness value) and the machine being in an uncontrolled environment.
4. Corrective actions: The pill colors were changed to red, blue, and yellow due to their vastly different HSV values. The computer vision would not confuse the two colors anymore, and the stepper correctly responded to all of the possible 9 combinations of pill color. This meant that the computer vision and the PySerial communication was working accurately as well. This achieved 100% matching the technical objective.

11.3 *Test 3: Hopper Test (repeated for i=10 times)

1.Test results and Responsibility: The test was done by Nicole. The hopper was able to dispense the pills correctly. However, the pill would occasionally get stuck as there are many orientations at which the pill can land in the hopper.
2. Comparison: Overall the hopper matches the expected results. However, there is a mechanical issue with the orientations of the pills jamming the hopper.
3. Analysis: The hopper works and the stepper is able to turn 90 degrees when a button is pressed. The hopper system occasionally got jammed. The chances of the hopper jamming increases with the amount of pills loaded into the machine
4.Corrective actions: Due to time and cost constraints, loading less pills into the hopper at a time will prevent it from jamming, so loading no more than 2 pills at a time prevented this.

11.4 * Test 4: Final Test of Everything (repeated for i=18 times)

1.Test results and Responsibility: All of the members set up, ran and observed this test together. The system is able to sort all of the 9 test cases twice, accurately. The pills were sorted into three different piles, and it took about an average of 12 seconds to sort a pill.
2. Comparison: The system matched the expected result as it sorted all the pills with 100% accuracy and it took less than 65 seconds to sort a pill.
3. Analysis: The Pill Sorter is successful as it matches the technical objectives.
4. Corrective actions: No corrective actions were needed.

12 * Conclusion and Future Work

12.1 * Conclusion

The project achieved the objective of taking in three pills of different colors and sorting them, using electrical, mechanical, and computer engineering. The project successfully achieves all of the technical objectives written in Section 2. The team worked together to achieve the objectives of this project.

Peter Chau learned about mechanical design and manufacturing. Given that 3D printing and managing the transportation of the pills was a large portion of my work, it was also pertinent to the overall functionality of the project. He also learned serial communication through PySerial. Professionally, he learned how to manage team members and allocate tasks accordingly.

Nicole: I learned alot about hardware design and experiment design. I was able to apply what I learned from my internship which is hardware testing. I also refreshed my skills learned from my classes and also learned a bit about PySerial. Professionally, I can apply what I learned about test or experiment design to my job. In the future, I will have to write a design test for products, and this project gave me experience. Personally, I am more confident with communicating and presenting in front of others.

Ted: I learned how to use OpenCV for color and detection algorithms in computer vision. Personally, I feel more confident in my ability to create robust software. Professionally, troubleshooting and systems integration were skills that were heavily developed. Additionally, I took a step forward in terms of personal development.

12.2 Future Work

For future work, an improvement of the computer vision software is crucial. The design team sought to generalize our program to any kind of pills. This can be achieved by using TensorFlow and building a performance oriented computer, which was mentioned in Section 5.4. Additionally, if more time was given, creating a more solid structure would be considered for controlled lighting. The computer vision was sensitive to different lighting, reflections, and shadows. A cover would allow all colors to be recognized, including red and orange, which have close HSV values (hue, saturation, and brightness value). Different motor drivers need to be tested as the L298N motor driver has thermal issues. A possible option could be the TB6612FNG or the A4988 motor driver. Adding a switch for the whole machine and coding the PySerial communication adds marketability by being more user friendly.

12.3 * Acknowledgement

Peter Chau would like to personally thank his mother for all the help she provided along the way, by buying all the necessary materials.
The group would like to thank Professor Chomko and Christopher Eng for their general advising, assistance, and review.

13 * References

[1] R. Chomko. (2023). Contemporary Engineering Issues Senior Design [PowerPoint slides].

[2] "Arduino M&M Color Sorter." Instructables. https://www.instructables.com/Arduino-MM-Color-Sorter/. (accessed Oct. 3, 2023).

[3] Machinery, Shineben. "Automatic Sorting Machine for Pills Capsules Tablets High Speed Sorter Pill Sorting Machine". Youtube. https://www.youtube.com/watch?v=ctetCCCBZ_g. (accessed Jan. 15, 2023).

[4] "Tensorflow Framework & GPU Acceleration from Nvidia Data Center." NVIDIA. www.nvidia.com/en-sg/data-center/gpu-accelerated-applications/tensorflow/. (accessed Jan. 3, 2023).

[5] "Capsule Counting Machine SBM-CM12L." ShineBen. https://shineben.com/product-details/capsule-counting-machine-sbm-cm12l/. (accessed Jan. 3, 2023).

[6] "USB Port and Power Delivery: An Overview of USB Port Interoperabiliy" IEEE. https://ieeexplore.ieee.org/document/7138710/. (accessed March 15, 2023).

14 * Appendices

Presents information that supplements the design specification, including:

*** Appendix A:** Parts List

Name of Parts	Quantity	Buyer	Total Cost	Purpose
l298n	2	Peter	$6	Motor Driver
Nema 17 Stepper motor	2	Peter	$28	spins the hopper system and
DC fans	2	Nicole	$7	Cooling heatsink on motor driver
12V 1A ac adapter	2	Peter	$7.20	Powering one of motors
12V 2A ac adapter	1	Peter	$6.50	Powering one of motors
Barrel Connecter	2	Peter	$2	Connection from power to driver
Breadboard power supply	2	Peter	$0	Powering the hopper and sorting mechanism
Button	1	Ted	$0	For the hopper
Acrylic/3D Hooper Casing	1	Peter	$30	For the Hopper
3D slide	1	Peter	$15	Guides pills for sorting mechanism
Styrofoam Blocks	2	Peter	$25	Structure
Aluminum sheet	5	Peter	$14	Transportation of pills
Camera	1	Peter	$0	Computer Vision
Laptop/PC	1	Ted	$0	Computer Vision
Breadboard	3	Peter and Nicole	$0	Wiring
Wires	100	Peter and Nicole	$0	Connection
Cardboard for cover	1	Nicole	$0	Creating a controlled environment
Environment light	1	Ted	$32.28	Provided ambient light if necessary
DC Motors	4	Nicole	$12	Conveyor belt
Cloth	1	Nicole	$6	Conveyor belt

Total cost spent: $190.98
Total cost of final product: $140.70

*** Appendix B:** Equipment List

Name of Equipment	User	Total Cost	Purpose

Keysight DC Power Supply	Nicole	$0	Part testing
Agilent Multimeter	Nicole	$0	Part testing

* **Appendix C:** Software List

https://github.com/OYoung07/EE175AB-Pill-Itemzier

https://github.com/OYoung07/EE175AB-Hopper-Code