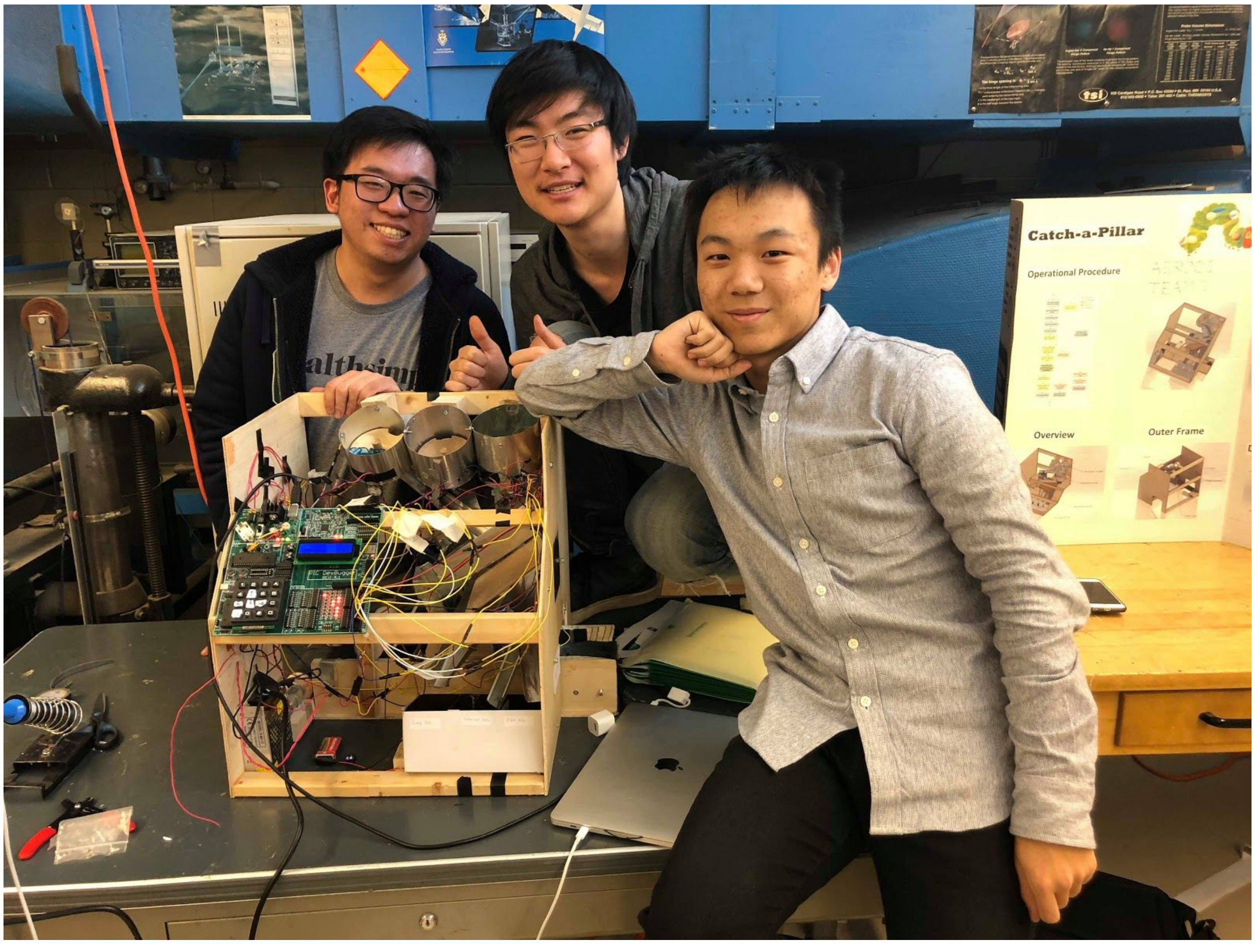
AER201: Engineering Design

# The Catch-a-Pilla




**Team 7**

**Zili Ge 1003213845**
**Lihan (James) Jin 1003284233**
**Yuan Hong (Bill) Sun 1003039838**

**PRA0101 | Teaching Assistant: Joachim Sarr | Instructor: M. Reza Emami**

## Acknowledgements

AER201—Engineering Design is a challenging, time-consuming, but rewarding experience for second year Engineering Science students. It is a stage where prior knowledge learned in various foundational courses in the past two years, and different skills and talents from aspiring EngSci students, converge to produce an amazing product.

Being the first truly hands-on engineering design project for most of the EngSci students, there are many challenges along the way of constructing a functional machine. Even as a team, we would not have achieved our final goals if we had not received assistance, recommendations, or encouragement from the following individuals. Here, we would like to thank:

*Professor Reza Emami, for using his engineering experience and intuition to develop this amazing course to challenge our minds and guide us through this project*

*The teaching assistants, for organizing the useful subsystem lectures to give us a good foundation to start with*

*Our TA Joachim, for his continuous and kind support of our team and great advice during weekly practicals*

*Colin Harry, the machine shop supervisor, for his help in developing our electromechanical subsystem and using the right tools in construction*

*Our fellow teams, for promoting a healthy and cooperative working environment and sharing strategies and advice in overcoming common problems*

*The Internet, for providing us answers to various questions, as well as reference designs, sample codes, circuits, and suggestions for improvement*

## Abstract

This report documents a proposed solution to the issue of ineffective pill sorting by nursing homes as put forth by the Request For Proposal document. The client, a nursing home, struggles with manually organizing and delivering their patients' weekly medications; so there is a need to develop an autonomous pill packaging machine that is inexpensive, compact, and efficient compared to existing solutions such as industrial pill machines or personal home-use pill dispensers.

The machine is required to take in a custom 14-compartment weekly pill organizer, as well as three types of pills: spherical, long, and flat shaped. The user inputs a set of operating instructions into the machine, which includes the prescription, frequency, and days of the week. The machine then loads the pills needed into the pillbox, and returns it to the user with its lids closed. It also stores and counts any extra pills, and returns them to the user.

Throughout this course, the design team is responsible for designing and fabricating a functional machine prototype as a proof-of-concept. Currently, at the time this report was written, the machine prototype produced by the design team can perform some of the tasks outlined in the RFP, but it still needs significant improvement in order to be fully functional.

For the design team, this is truly their first hands-on engineering design project. The design challenges posed by this problem offered an excellent learning experience, where the team members learned more about engineering design processes, organization, time management, and budgeting.

# Table of Contents

## Notations, Abbreviations, and Terms

| Abbreviation | Definition |
| --- | --- |
| A | Amperes (current measuring unit) |
| N | Newtons (force measuring unit) |
| V | Volts (voltage measuring unit) |
| W | Watts (power measuring unit) |
| AC | Alternating current |
| DC | Direct current |
| IR | Infrared, invisible light waves with wavelength longer than nominal red light |
| N*m | Newton meter (torque measuring unit) |
| kg*cm | Kilogram centimeter (torque measuring unit) |
| AHP | Analytical Hierarchy Method of decision making |
| CAD | Computer-aided design; in this report it stands for 3D models of the machine prototype using SolidWorks, a CAD software |
| LED | Light emitting diode, an electronic device that lights up when electricity passes through it |
| MCU | Microcontroller unit |
| OCR | Optical character recognition |
| PWM | Pulse width modulation |
| PCB | Printed circuit board |
| PIC | Peripheral Interface Controller; in this report it stands for the PIC18F4620 DevBugger board (one of the two microcontroller boards used) |
| PVC | Polyvinyl chloride; a widely used type of synthetic plastic polymer |
| RFP | Request for Proposal; a document outlining project requirements |
| RGB | Red-Green-Blue (color model based on additive color primaries) |
| RTC | Real Time Clock (on the PIC board) |

## Term Definitions

- **Arduino Nano:** One of the two microcontroller boards used.
- **Client:** Any nursing home that desires to purchase, install, and operate the machine.
- **Component:**
  - Any piece of material that is either continuous (not put together by joinery; or the joinery is permanently bound using glue, nails, screws, or solder), or electronic elements that come in a pre-made package that includes all of its subcomponents.
  - Examples: Pill loading compartment, conveyor belt not including motors, motors and actuators, PIC DevBugger board.
- **Degree of motion:**
  - A specific type of motion carried out by a component or subsystem, defined by 1. Being rotational or linear, and 2. The plane in which the motion occurs.
  - Example: rotating disk in the xy plane. However, if this disk may also move along the z-axis, then two degrees of motion are involved.
- **Error:**
  - Any one misplaced pill (whether in the wrong pill box compartment, or in the wrong reservoir for retrieval)
  - Jamming of the pills or the pillbox
  - Any miscounting of the number of pills
  - Damage of the pills or the pillbox
- **Machine:** Abbreviation for the pill-boxing machine.
- **Subsystem:**
  - A set of interconnected components that perform a specific function that is critical to the machine's operation.
  - Not: "microcontroller", "circuits and sensors", or "electromechanical"; a subsystem may involve all three of these roles.
  - Example: pill dispenser subsystem, which is comprised of pill loading compartments, rotating disks with pill slots, axles, and servos operated by PIC18F4620 MCU, and whose purpose is to dispense pills one at a time with precision and reliability.
- **User:**
  - Any personnel involved in the operation of the machine; who may perform at least one of the following tasks:
    - Loading / retrieving the pill box
    - Loading / retrieving the pills
    - Entering operation parameters
    - Conducting repair/replacement of components (defined below)

# 1. Introduction

## 1.1. Summary

A nursing home requires a solution for autonomously packing three different types of medications into a 14-well weekly pill organizer [1]. The RFP calls for the design and implementation of a pill-boxing machine. The user deposits three distinctive types of pills into three containers, which, upon the input of a sequence by the caretaker, autonomously packages the pills to those specifications into the pillbox, and returns all the remaining pills in an organized fashion [1]. As per the RFP, the cost of the machine cannot exceed $230 Canadian dollars [1].

## 1.2. Statement of Need

The Request for Proposal states that a nursing home needs to autonomously package the weekly medications of its residents. Since the repetitive movements of the slow process of manual pill sorting and loading can lead to fatigue-related chronic illness, the entire process should be automated to maximize efficiency and employee safety [1]. In addition, manually sorting and placing the pills into the pillboxes would expose the pills to bacteria on the hands and in the air, which may be harmful to the health of those taking the pills later on; automatic pill-boxing should reduce the chances of bacterial contact, thus, less chance of illness for the patient.

The medication comes in three types (shapes): the spherical-shaped pill, the flat-shaped round pill, and the long pill. There is some variability in mass and size of each type of pills. With the pill boxing machine, the workers at the nursing home would manually enter the prescription into the machine through an interface, load the three different types of pills into specified collectors (assuming the user puts enough of each type of pills to satisfy the prescription), and insert the pillbox into the designated slot. Within 3 minutes, the machine would need to place the appropriate number of each type of pills into the pillbox, and return the pillbox with all the lids closed and locked. The remaining pills would need to be sorted and placed in collection bins where the user can easily retrieve them by type. The display would show the number of each type of pills remaining. For safety purposes, the machine must also have an easily-accessible emergency stop button. [1]

## 1.3. Goal

The goal of this project is to design and fabricate the proof-of-concept prototype of a machine that can correctly and efficiently package various daily pills in a medication box based on the given instructions from the Request for Proposal.

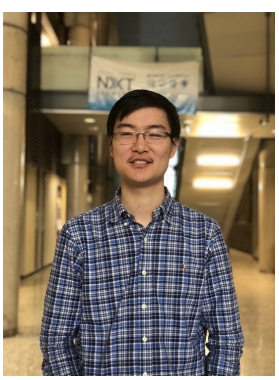
## 1.4. Stakeholders

1. Nursing homes and nursing home workers: Nursing homes need to provide prescription drugs weekly for some of its residents. The nursing home workers need to organize the pills into pillboxes efficiently and correctly so they can be delivered to the residents.
2. Nursing home residents who take prescription/medication: Some nursing home residents need to take prescription drugs. They need their pills correctly sorted into pillboxes and delivered to them.
3. Design Team: The team desires to satisfy the client requirements for the machine, and hopes to gain design experience and learning from the design and construction of this machine.
4. Professor M. Reza Emami & Joachim Sarr: The course instructor and teaching assistant desire the design team to demonstrate an understanding of the course material, as well as teamwork and coordination in a thorough design process.

## 1.5. Team Members, Roles, and Expertise

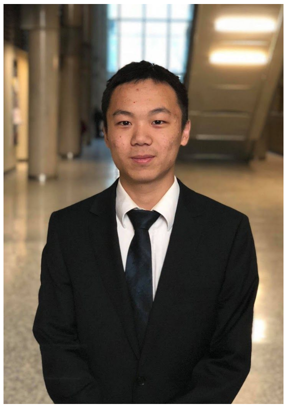
**Zili Ge - Circuits and Sensors**

I am a second-year engineering science student who is passionate about sensor design, signal processing and electrical engineering. As someone who has worked with biomedical engineers and nursing home caretakers alike throughout my time at the University of Toronto, I am acutely aware of the logistics in medicine delivery. I believe that this is an area that is sorely in need of an improvement, and I wish to be the one who makes that happen. As the circuits and sensors member, I am responsible for the design of all circuit and safety components, in addition to soldering and wiring.



**Lihan (James) Jin - Microcontroller**

I am an analytical and detail-oriented person who enjoys being engaged in complex design problems. I strongly believe in quality over quantity, and seek to create solutions that are elegant and robust to address pressing needs in society. Through various courses and completion of a summer project, I gained proficiency in object-oriented programming. Thus, my role this project is to create an efficient and reliable packaging algorithm, interface with various sensors and actuators, and to create an intuitive and user-friendly machine interface.



**Yuan Hong (Bill) Sun - Electromechanical**
Experienced in mechanical and mechatronics design, I enjoy solving multidisciplinary problems through engineering design. Through my expertise, I hope to create a solution that is simple in complexity, user-friendly, safe, reliable, and environmental-friendly. By solving this problem, I hope to make a difference in the community by helping out certain groups of people in need.

As the electromechanical member, I am in charge of designing, selecting and constructing the physical parts of the machine, including the frame, all the various non-electrical components in the subsystems, and all the supporting structures. Through calculations, modelling, and testing, I help determine the specifications required for choosing the appropriate materials, size of members, and actuators. By optimizing the size, weight, and operation time of the machine, I ensure that the prototype serves its best performance.

# 2. Perspectives

## 2.1. Theory

The following equations from classical mechanics were useful for calculations related to structural integrity of the machine, as well as for choosing the actuators:

- Force: $F = m \times a$
- Moment: $M = F \times d$
- Moment of inertia: $\tau = I \times A$
- Torque: $T = F \times r \times \sin(\boldsymbol{\theta})$

In circuits design, some basic principles were used to determine the appropriate tolerances for choosing different components, such as resistors, diodes, transistors, and chips. These include:

- Ohm's Law: $V = I \times R$
- Power Law: $P = V \times I$
- Resistance in series: $R_{eq} = \Sigma R_i$
- Resistance in parallel: $R_{eq} = 1 / (\Sigma 1/R_i)$
- Kirchhoff's Current Law: $I_{total} = \Sigma I_i$ for all branches at a node
- Kirchhoff's Voltage Law: $V_{total} = \Sigma V_i$ for all components around a loop

## 2.2. History

Although many types of pill-packaging machines have been designed in the past, they all fell into two opposite ends of a spectrum. On one end are heavy, industrial packaging machines that can sort, bottle, and label various capsules, powders, and liquids for distribution in the Pharmaceuticals industry. For instance, the CVC-1220 Multi-Channel Tablet Counter (Fig. 2.1) made by Busch Machinery [2] guarantees 100% accuracy, and packages between 1 to 9998 pills in a single package. It is made of stainless steel, and has various mechanisms for controlling dust levels. It is capable of packaging 50 bottles per minute, and can be placed in production lines with capping and labelling machines for complete system. On the other end of the spectrum are personal pill dispensers that dispense a single patient's description on a timely basis (these will be explored in 2.3).

Clearly, there exists a gap for intermediate-sized machines that are inexpensive yet reliable which can package medication for a moderate number of patients in hospitals and nursing homes. The prototype discussed in this report is a proof-of-concept design for a machine that can autonomously package pills in a reasonable amount of time with minimal nurse oversight and no errors in such a setting.

Figure 2.1: CVC-1220 Multi-Channel Tablet Counter

## 2.3. Surveys

### 2.3.1. Literature Survey

32 million individuals in the United States alone take three or more pills daily. However, up to 75% of individuals do not adhere strictly to their regimen in some way [3]. For patient medication, it is vitally important to ensure that pills are taken in the right order and at the

appropriate times. This can be a difficult task, especially pertaining to individuals that are on a regimen of multiple medications.

Our prototype is designed to address this concern. The process of pill sorting is streamlined through automation, freeing up more time for nurses and personal support workers. In addition, the area of pill administration has been identified as an area for further optimization. It was found that up to 40% of pills are wrongly administered in a clinical setting. Out of this 40%, half of all errors occurred during drug administration, with the most common mistake being that the pill is taken too early or too late. By sorting into discrete time steps, this prototype makes it easier for healthcare workers to properly sort the pills by the time in which they should be taken. For patients on a time-sensitive pill regimen, this is of vital importance [4][5].

### 2.3.2. Market Survey

A market survey was conducted, and it was that there has been efforts to develop automated pill dispensing mechanisms in homes. An example would be the startup Pillo (Fig. 2.2a) [6][7], which addresses the individual or family segment of the market of automated pill dispensing. However, this category of pills dispensers cannot satisfy the purpose of organizing multiple prescriptions into pill boxes for a nursing home, given the requirements defined by the Request for Proposal. Firstly, at a price of $449 US dollars, it is far too expensive, exceeding the cost constraint of $230 Canadian dollars. Additionally, this dispensing device is only specific to a particular patient's prescription, and cannot be used to package medication for multiple patients at a hospital or nursing home.


Figure 2.2a: The pillo dispensing machine

### 2.3.3. Patent Survey

The US 7359765 B2 patent [8], shown in the figures above, is an electronic pill dispenser. This device, like the Pillo, serves to remind patients when they need to take medication, and prevents accidental overdoses by controlling the number of pills dispensed each time. Although the function of this device is different from that of the requested pill-boxing machine, it gave inspiration for a mechanism that can consistently dispense one pill at a time. As shown in Fig. 2.2b, the pills are caught by a wheel with slots, and precisely one pill can be released by turning the wheel 180 degrees. This design lead to many of the brainstormed ideas that involve rotating disks/wheels with holes tailored to a specific type of pill, including the selected design, as will be discussed in section 4.2.3.

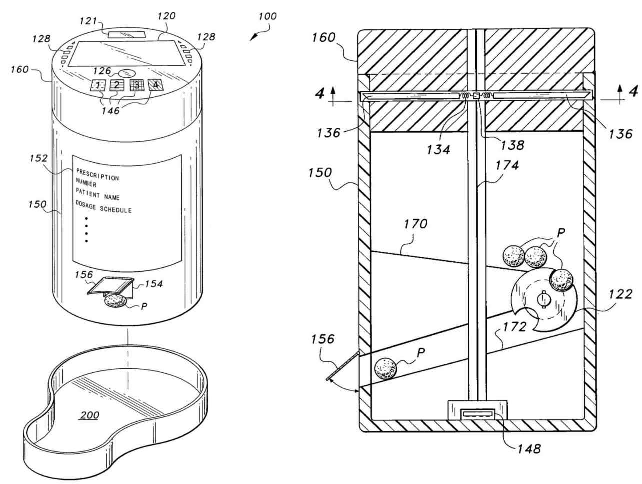



Figure 2.2b: The US 7359765 B2 patent

### 2.3.4. Idea Survey

Aside from patents and existing products, projects from a variety of miscellaneous sources also gave inspiration for various mechanisms. The video [9] shows a CAD simulation of a design for the pill dispenser (Fig. 2.3). This design gave the idea of using trapezoid-shaped funnels as well as to dispensing one pill at a time using two pieces of material with matching holes that slide over each other. The Skittles Sorting Machine [10] (Fig. 2.4) uses an RGB color sensor to determine the color of Skittles as they fall through a funnel, after which a servo is used to turn a channel that directs the Skittles to their respective containers. This project revealed a color sensor that can be used for determining box orientation. The rotating channel was also the precursor to the ‘pinball’ mechanism (6.1.5) used to direct falling pill to the right side of the box. Finally, a pill dispenser featured in another YouTube video [11] gave rise to the reliable yet simple cylindrical dispenser design (Fig. 2.5) that can dispense one pill at a time.

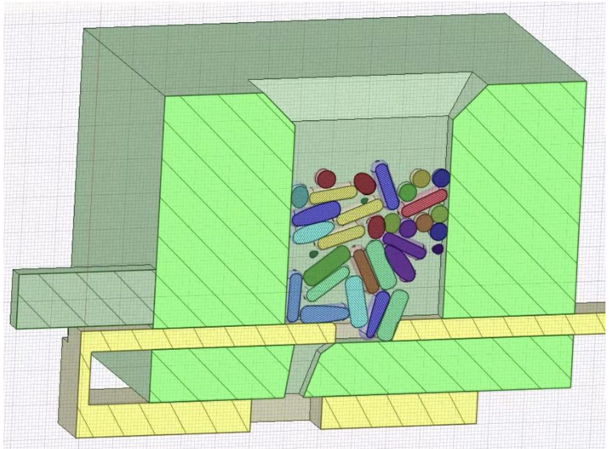


Figure 2.3: A pill dispenser design from video

Figure 2.4: The Skittles Sorting Machine

Figure 2.5: Another pill dispenser design from YouTube

# 3. Project Concept and Design Parameters

## 3.1. Functions and High Level Objectives

The design team developed the following set of design high level objectives for this machine based on the Request for Proposal and user needs. These high level objectives are listed in the order from the highest to lowest importance:

1. Safety: Protection and prevention from conditions that can cause danger to the user, the machine, or the surrounding environment.
2. Usability and User-friendliness: Little time and effort is needed to set up and calibrate the machine, and the machine is modular so that parts can be replaced or repaired easily.
3. Manufacturability and Complexity: Machine is simple in design and easy to construct.

4. Reliability and Efficiency: The completion of tasks using minimal time possible, and resulting in a minimum amount of errors. [1]
5. Compactness and Portability: The machine can be easily lifted and transported. [1]
6. Affordability: The price of the machine is affordable.
7. Reliability: The machine can withstand wear over repeated usage and mechanical stress, and functions consistently with a low failure rate. [1]
8. Elegance: Machine looks elegant, and operates quietly and smoothly. [1]

For more details about these high level objectives, please see Appendix A.

## 3.2. Objective Model

Based on the high level objectives defined above, an objective model with a set of interconnected metrics, constraints, criteria, and utility functions was developed. The full chart is shown in Table A below. Other requirements defined in the RFP are also considered for designing the machine. A full list of requirements as stated by the RFP is available in Appendix B.

## 3.3. Budget

As per the RFP, a maximum of $230 can be spent for all components in the entire prototype machine. However, to compensate for factors such as the cost of the project kit ($300), unused materials, material losses, and replacement for broken parts, we plan to spend less than $1,000 as a team for the entire project.

# 4. Problem Assessment and Subsystem Design

## 4.1. Division of the Problem

The following flowchart (Fig. 4.1) represents the team's interpretation of the functional procedure of the machine, and was derived from requirements and functionalities given in the Request for Proposals.

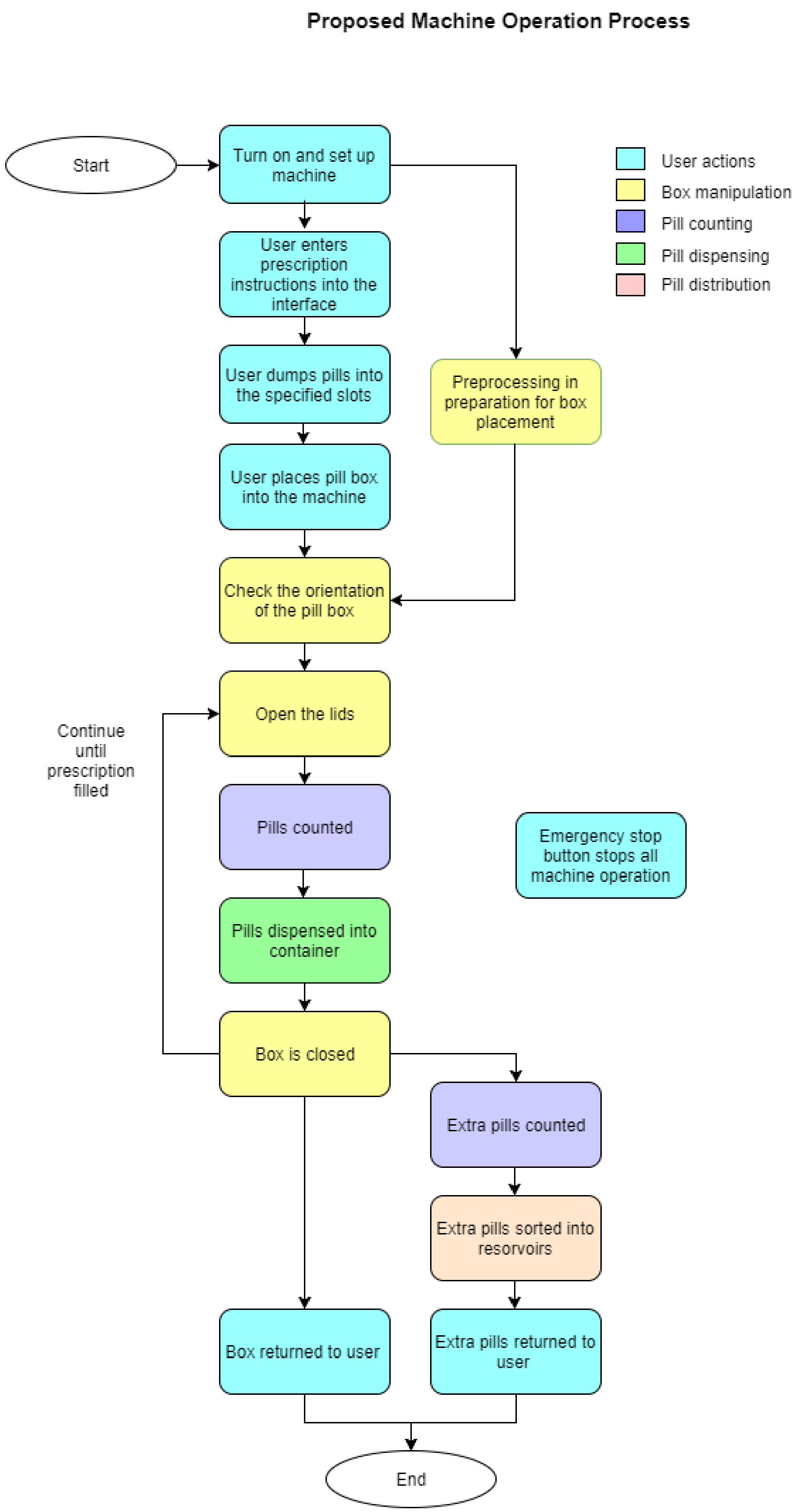



Figure 4.1: Machine operation flowchart as interpreted from the RFP

From the proposed operational process flowchart above, the main functionalities of the machine were identified as the following (not including user actions):

- Box manipulation
- Pill counting
- Pill dispensing
- Pill distribution

**Box manipulation** consists of the process of determining the orientation (direction) of the pillbox, opening the lids for pill dispensing, positioning the pillbox with the precision required to dispense the pills into their respective containers, in addition to the process of closing the lid post dispensing. The box must be accessible to the operator upon completion.

**Pill counting** is an autonomous process whereby the quantity of pills the user inputs into the machine is determined, with excess pills dispensed into an easily accessible location.

**Pill dispensing** is the process that begins as soon as the user dispenses the pills into the collector bins, and the pills then drop down one at a time into the pill distribution mechanism.

**Pill distribution** is a combination of all the processes involved in guiding the pills into different areas after they are dropped down from the dispenser. This includes different compartments of the pillbox, as well as the three collection reservoirs for extra pills.

The design process, justification, and decision making with regards to components and subsystems to address these functionalities are summarized in sections 4.2-4.5. Note that these are the original designs discussed in the Proposal. Relevant sections of this document has been included in Appendix C. Design changes made during integration and debugging as well as their justifications are detailed in Section 4.6.

## 4.2. Box Manipulation - Original Design

The process of pillbox manipulation can be roughly broken down into 4 sub-processes:

- **Checking box orientation**, which distinguishes between the two possible orientations of the box once its placed onto the conveyor belt.
- **Box movement**, the positioning of the pillbox relative to the pill droppers (or vise versa) in order for the pills to drop into the correct box compartments.
- **Opening the lids** so that the pills can be loaded into the pillbox compartments,
- **Closing the lids** after the operation is complete so the pillbox can be returned to the user.

### 4.2.1. Checking Box Orientation

As stated in the RFP, there exists the need to differentiate between the two orientations of the box (Sunday-end-first and Monday-end-first) as it was assumed that the user places the box on the conveyor without checking its orientation. The only physical differentiating features between the two sides are the lid colors in addition to the text on the board. Reading text requires OCR, which involves a camera, and this was deemed to be beyond the budget. On the other hand, the colors of the lids can be differentiated using an RGB sensor. The Adafruit TCS34725 sensor (Fig. 4.2; see Appendix E for datasheet extracts) was selected. This sensor provides the necessary functionality as it detects the intensity of red and blue light. It has an IR blocking filter to increase its sensitivity to visible spectrum light, which is beneficial considering the part-transparency of the lids. In addition, it has an on-chip LED to illuminate the object of interest, which makes it idea for use in the dim-lit interior of the machine. Finally, its cost of $9 make it appropriate for the tight budget.

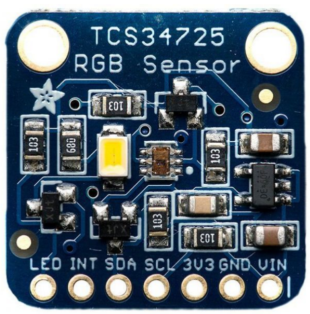



Figure 4.2: the Adafruit TCS34725 RGB Sensor

### 4.2.2. Box Movement

The machine must be precise enough to dispense pills into each individual compartment, and to be robust considering uncertainties involved with the user. It was identified that either the box itself needs to be moved or the dispensers must be positioned over the box. The box movement mechanism was designed with the goals of minimizing the number of degrees of motion, minimizing size, and maximizing reliability/accuracy (see Appendix C for further explanation and justification).

Through AHP and utility-based analysis, it was determined that a single-conveyor system was most appropriate for controlling box movement (please refer to Appendix C for the detailed AHP analysis and other solutions considered). The system is shown in Fig. 4.3 below. This solution prevailed over others as with only one motor, it had far less components than other solutions. It had only a single degree of motion since the box was only being moved forward and backward. Additionally, its geometry was simple - involving essentially two rollers and a piece of fabric, it can easily be manufactured. To assist the box in overcoming resistance as it moves through a pusher block is attached to the conveyor to push it through. To ensure that it stops at the correct positions, contact sensors, consisting of strips of aluminum, were attached to the sides of the conveyor belt, in positions such that when wires stretched across the pusher block make contact with them, a particular row of the box is situated directly below the dispensing system.

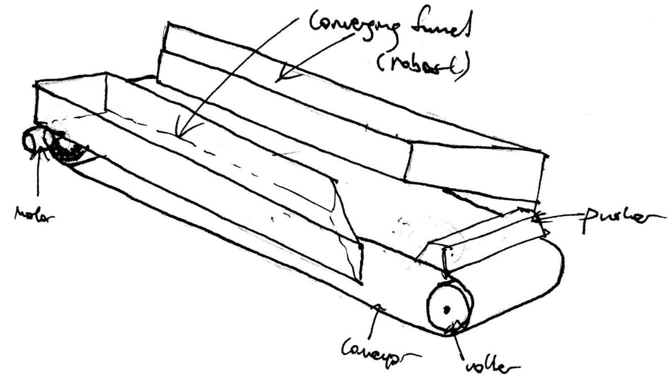



Figure 4.3: The selected box movement mechanism: the conveyor belt with pusher block

### 4.2.3. Opening the Lids

As per the RFP, the user would clip open all the lids of the pillbox before placing it inside the machine. At rest position, there would be a small gap, ~5 mm at the widest, between the lid and the box structure where a mechanism can reach inside and open it.

From the utility-based analysis, it was found that the double wedge design (Fig. 4.4) was the preferred lid-opening mechanism. The wedges are installed on both sides of the conveyor belt, and would wedge the lids open as the box move through. This mechanism was chosen as it was entirely mechanical, had few components, and was easy to manufacture.

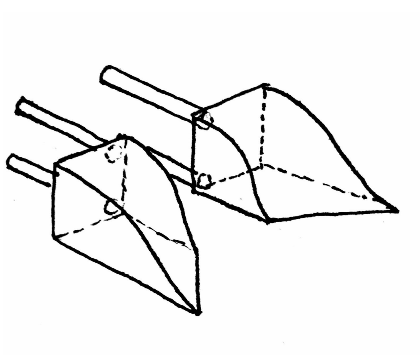


Figure 4.4: The double wedge design used to wedge open the lids of the pillbox

### 4.2.4. Closing the Lids

As per the RFP, the lids of the pillbox need to be locked in place when it is returned to the user. Through testing it was found that in order to effectively close the lids, a force must be applied in both the vertical (downwards on the box) and lateral direction (toward the central axis of the box). Without the lateral component, an extremely large vertical force was required (see Appendix C for testing results), and this may damage the box. Therefore, designs such as a single roller, a flat plate, and a plain wooden block that apply only downwards force are ruled out. The selected solution consisted of a wooden block that has been carved to apply force in both directions, as well as series of levers to turn the rotational motion of a DC motor into vertical motion (Fig. 4.5). Although this structure is relatively complex, it was deemed to be more reliable.

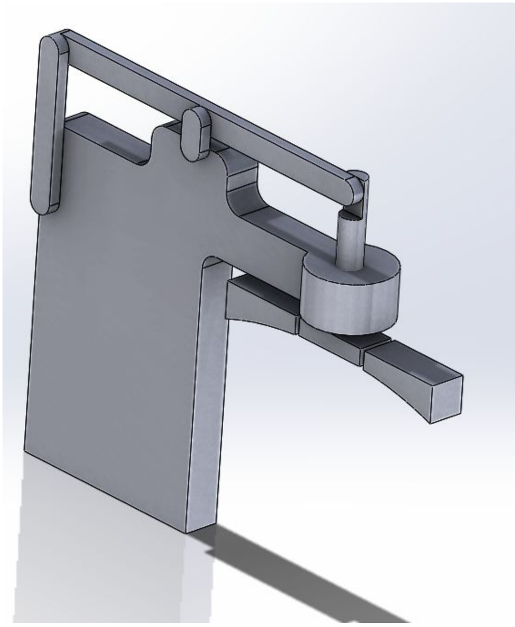


Figure 4.5: The original lid-closing mechanism, consisting of levers, a motor (not visible here), a carefully carved block of wood to apply both vertical and horizontal forces, and a supporting structure.

## 4.3. Pill Counting - Original Design

As per the RFP, the machine needs to count the remaining number of pills for each of the three types. A pressure sensor (Fig. 4.6) was considered where the impulse from a falling pill would cause two wires to touch, producing an interrupt signal. This sensor was found to be unable to detect the ellipsoid pills rolling on a flat surface; this was remedied by dropping the pills from a height. This design was chosen for its cost-effectiveness and ease of manufacturing.

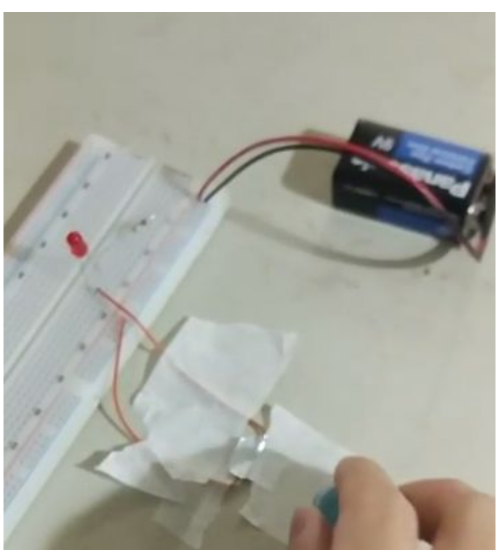


Figure 4.6: The original pressure sensor, where a circuit is completed if a pill falls onto a piece of aluminum.

## 4.4. Pill Dispensing - Original Design

As per the RFP, there can be three separate compartments for the user to dispense three different types of pills. These dispensers must drop the pills one at a time into the pill-distribution system (4.5) to be loaded into the pillbox. At first a funnel design was considered; however these were discarded because making the openings too small presented a jamming problem, while making them very large made it possible for more than 1 pill to fit through at once. The design objectives for this mechanism were minimizing the number of components, minimizing geometric complexity, minimizing size, and maximizing success rate as determined through prototyping or online sources. The chosen design was a cylinder with a rotating base, which contained a hole just large enough to fit a single pill. Though somewhat geometrically complex, this design prevailed in reliability, as seen through YouTube videos [11]. Below is a drawing of the design (Fig. 4.7).

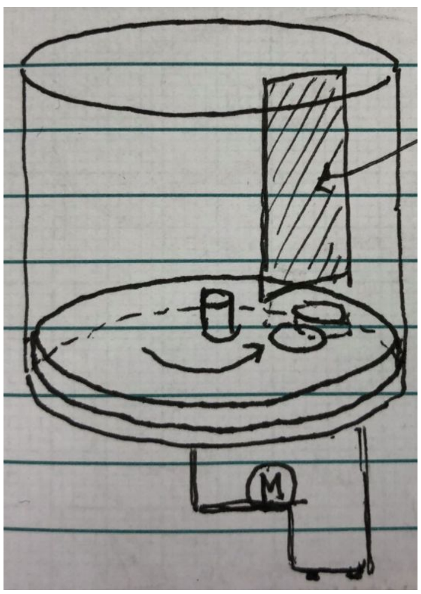



Figure 4.7: Drawing of the pill-dispensing cylinder. The motor attached to the bottom turns a disk, which contains a hold that can hold 1 pill at a time. When this hole lines up with a similar hole on the bottom, non-rotating disk, the pill is dispensed.

## 4.5. Pill Distribution - Original Design

Based on the definition of this subsystem, the pill distribution subsystem has two components: the component which guides the pills into the correct pillbox compartment, and the component that guides the remaining pills into the collection reservoir (although the two parts can be combined). The design criteria were again geometric complexity, number of components, error rate, and size. Degrees of motion was also considered as it was desirable to minimize the number of electronic actuators used. Through AHP and Utility-based analysis (see Appendix C), the pill distribution funnel with a ‘pinball’ mechanism to direct pills to either the morning or afternoon side was chosen. To release pills into the collections reservoirs for remaining pills, a servo is used to slide the baseplate of the funnel open, after which pills would drop into collection channels (please refer to Fig. 4.18). This design had an edge over others considered (Appendix C) due to its relatively simple geometry and manufacturability, and is shown in Fig. 4.8 below.

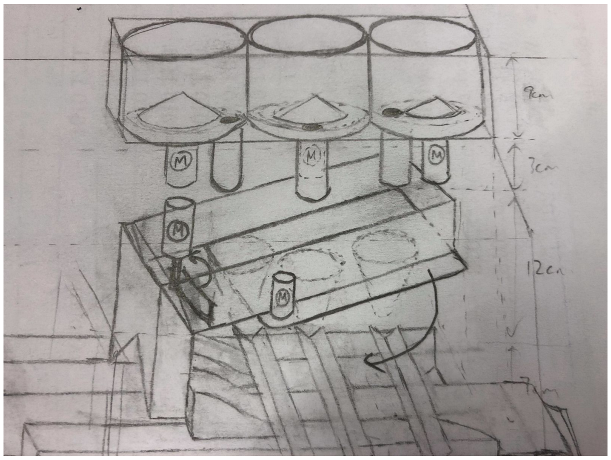


Figure 4.8: The original design for the pill distribution funnel. It is the central structure in this image.

## 4.6. Design Modifications

Since the Proposal Report, several design changes have been made to the functions mentioned above. The schematics for the original design from the Proposal are shown below (Fig. 4.9), but for more details about the originally proposed specifications of the machine, please refer to Section 5 of the Proposal document [12]

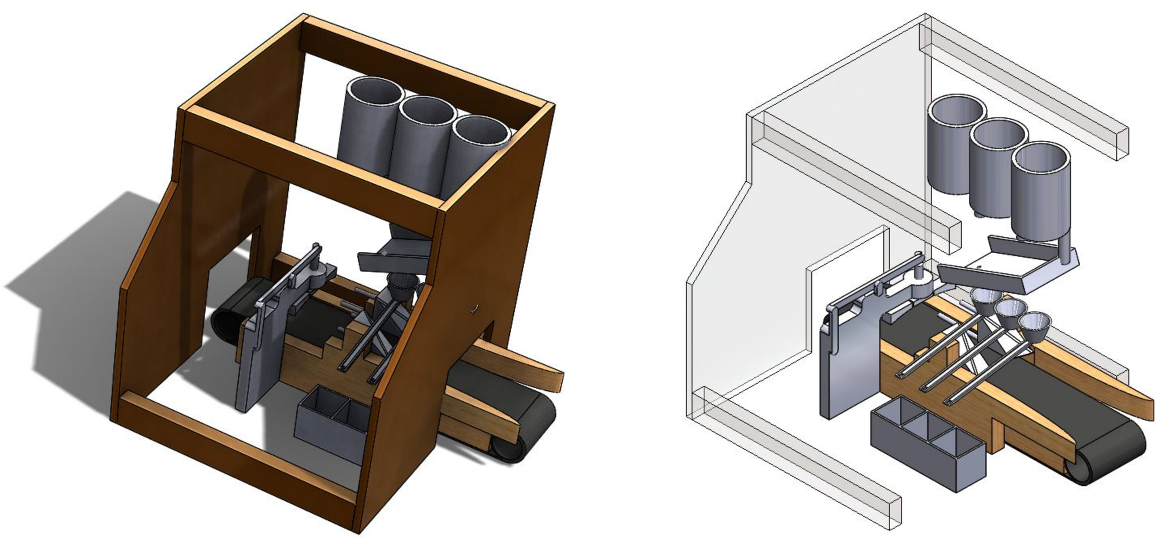


Figure 4.9: CAD views of the original design, as detailed in the Proposal.

The following iterations and design changes have been made:

### 4.6.1. Frame Structure

Instead of using just 4 cross beams, 6 were used to improve structural integrity (Fig. 4.10). Also, the beam on the top front was removed since it was obstructing the user accessing the dispenser cylinders, as well as increasing the difficulty of making repairs. A smaller beam was added near the top end of the slanted edge of the frame walls to create a slanted surface between the two adjacent beams, where the PIC board would be mounted. The angled mounting of the PIC board is easily accessible for the user standing in front of the machine.

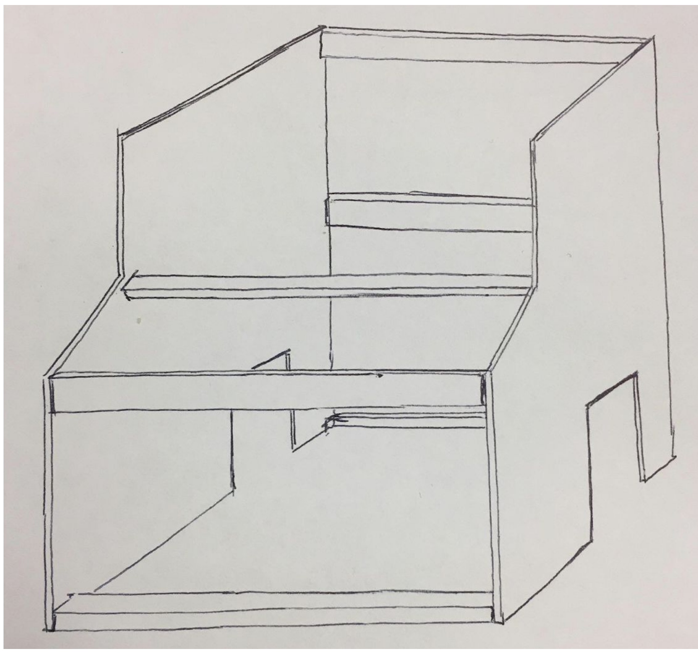


Figure 4.10: Simple sketch of the new frame design for the machine.

In addition, the height of the frame was increased from 42 cm to 45.5 cm (45 cm planned) to account for the possibility of parts not fitting inside as planned.

### 4.6.2. Box manipulation—Closing the Lids

Due to the complexity of constructing the pieces that compose the lid closing mechanism, as well as a tight budget, the original design was discarded. In the revised design, a wooden plate slanted at 20 degrees from vertical was attached to the back of the pill distribution funnel (Fig. 4.11). This is to lower the height of the lids right after they pass through the pill dispenser. If the lids are not lowered before the pillbox reaches the roller, they would hit the roller and potentially break, and stop the progression of the pillbox (which were the reasons why the single roller design was discarded).

Immediately behind the plate, a cylindrical roller made out of PVC pipe was mounted carefully onto two stands across the conveyor belt structure, with a metal rod going through the centre to hold everything together and to allow for the roller to spin freely (Fig. 4.11). The height of the lowest point of the roller allows just enough clearance for the pillbox to barely fit under; which, from repeated testing, supplies enough force to close all the lids as the box traverses under, and does not provide too much friction which stops the pill box from moving forward on the conveyor.

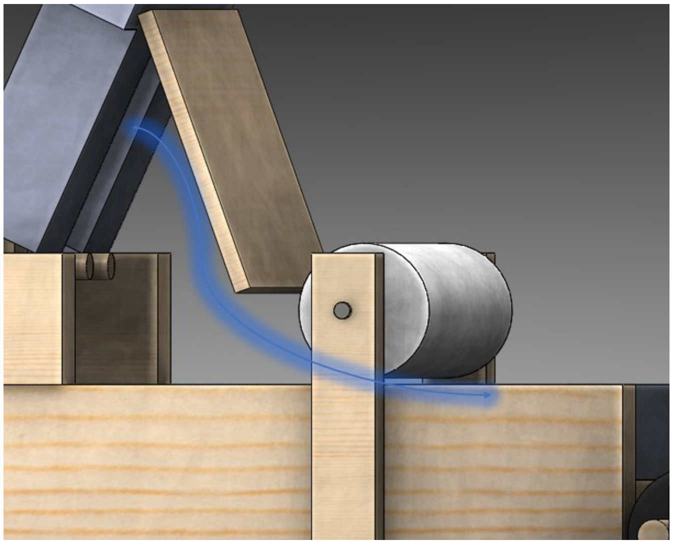
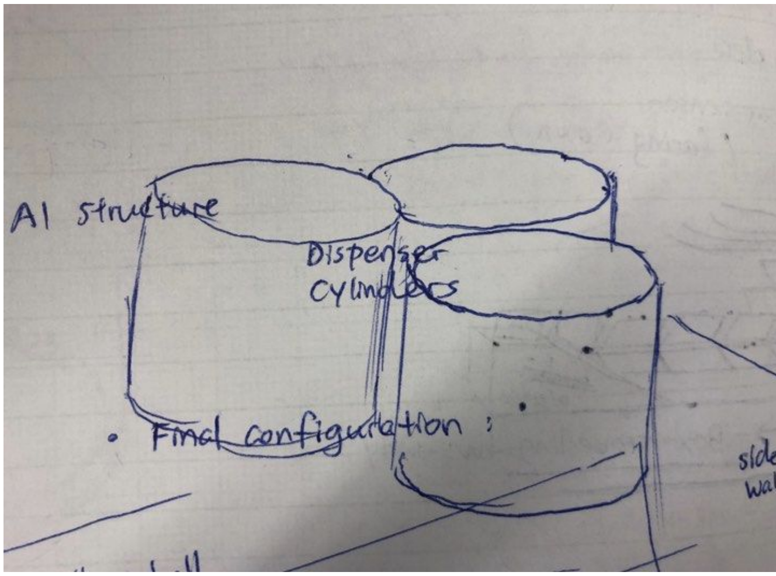


Figure 4.11: The revised lid-closing mechanism, with a wooden plate to lower the lids and a roller to snap them shut. The blue line is the approximate path followed by the top edges of the lids.

**4.6.3. Pill Dispensing**

The orientation of the dispensing cylinders, as well as the interior design of the dispensing cylinders, were significantly modified over two iterations to improve reliability.

On the initial iteration, first, to allow the pills to drop down onto the pill distribution funnel below, the orientation of the cylinders were changed from a single file line along the back side of the frame to a triangular formation (Fig. 4.12). This was done as the funnel was built to be too small for the original design, and it was an easier option to reconstructing the funnel, which would also lead to repositioning of various other components. Second, due to the small size of the cylinders, the inverted-cone at the center of each cylinder was removed to prevent potential jamming problems. Instead, they were tilted approximately 20 degrees so that the pills would gather in one area to be picked up by the hole in the rotating disk and so the wouldn't clog the hole at the bottom of the cylinder. Third, to prevent multiple pills to fall through the hole during each rotation, a barrier, made out of aluminum was placed over the opening (Fig. 4.13). Below the barrier, there is only enough clearance to allow for only one pill to go through at a time. The holes on the rotating disk were also significantly modified to ensure that only one pill can sit inside each time. For the long pills in particular, the holes were oriented such that the pills would not stand up vertically and jam the disks when it hits the barrier.



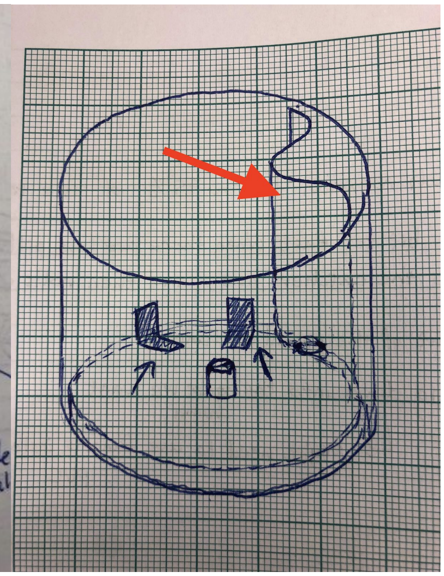



Figure 4.12 (left): An updated, triangular conformation of the dispenser cylinders.

Figure 4.13 (right): The structure pointed to by the red arrow is the aluminum barrier, designed to prevent pills from clogging the hole.

In the second iteration, it was found that for the long pills and flat pills, having a barrier surrounding the bottom opening creates a lot of issues. The pills sometimes stand up vertically, and when it hits the barrier, it acts like a lock and the entire disk jams. The same thing can happen when multiple pills get carried by the disk and tries to fit under the barrier at once. It was therefore decided that the barriers should be removed for the cylinders for long pills and flat pills. To ensure that only one pill falls through at a time, the cylinder were re-positioned such that the openings were at the highest point, and the rate of rotation was slowed down. In addition, to prevent two long pills or flat pills from stacking on each other or standing up vertically, the angle of the cylinders were increased from roughly 20 degrees to around 45 degrees (Fig. 4.14). The rotating disks were also made a lot thinner.

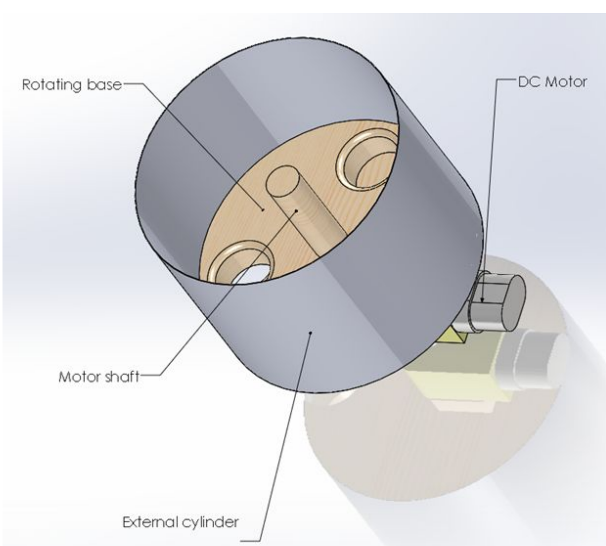



Figure 4.14: The new, approximate tilt of the dispenser cylinders.

On the third iteration, after addition of the IR sensors (4.6.5), two additional holes were cut into the rotating disks of each cylinder (Fig. 4.15). Having to read analog voltage from these sensors, it became impractical to have all three cylinders rotating at once (this extends into the microcontroller subsystem, and will be further discussed in section 6.3.3.4), which may lead to the operation taking longer than the time constraint of 3 minutes (constraint s in Appendix B). The additional holes increased the dispensing rate from each cylinder threefold, which made up for this disadvantage. Moreover, the shapes of the holes for the flat pills cylinder was modified such that pills inside them are oriented vertically (Fig. 4.15). This eliminated the possibility of two pills sitting inside one hole . Additionally, the positioning of the cylinders was changed back to a linear formation (Fig. 4.16) as this made it easier to install guiding channels (Section 4.6.4) and to make repairs to the cylinders.

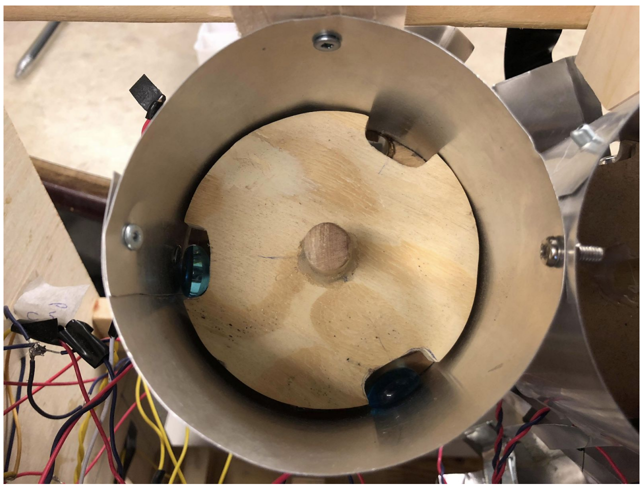


Figure 4.15: The modified design of the cylinders, with 3 holes cut into the rotating disk instead of 1. Also seen is that the long pills in the holes are oriented vertically; this was found to be more reliable.

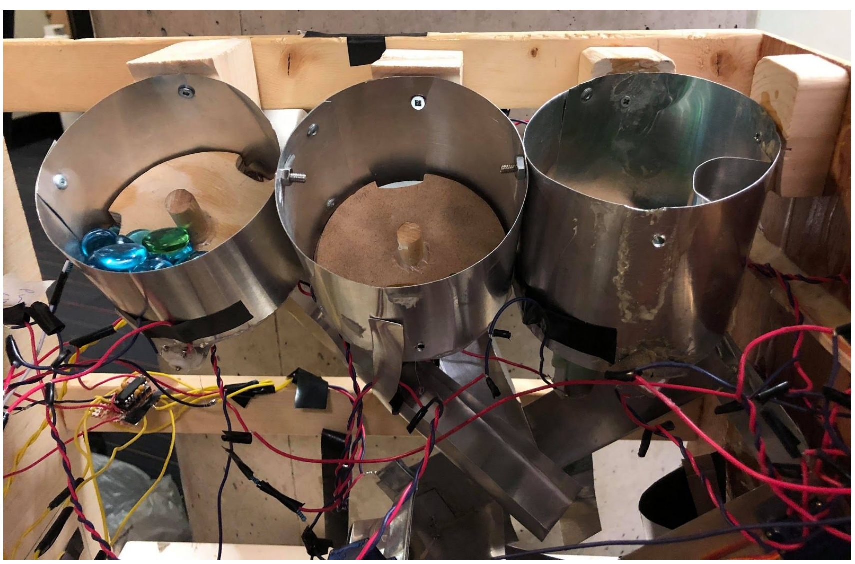


Figure 4.16: The final linear conformation of the dispensing cylinders.

### 4.6.4. Pill Distribution

The channels directing the extra pills to the collection reservoir were modified to accommodate for a greater range of error when pills were dropped down from the cylindrical dispensers above. It was found during testing that the original design (see Appendix D-5 for results), which is composed of a section of 1.5” PVC pipe and V-shaped channel, cannot capture most of the falling pills and direct them to the collection reservoirs, since the opening is too small. As a result, one of them (for the round pills) was modified such that a larger semi-circular funnel made out of aluminum sheets was attached on top to create a larger area of uncertainty with the wall of the machine frame (Fig. 4.17). For the flat and long pills, the V-channels were completely replaced with a trapezoid-shaped ramp, with the wider base right under the hole of the dispensing cylinder (Fig. 4.18). This eliminated the possibility that random bouncing of these pills due to their irregular and variable geometry causes them to land outside of the system.

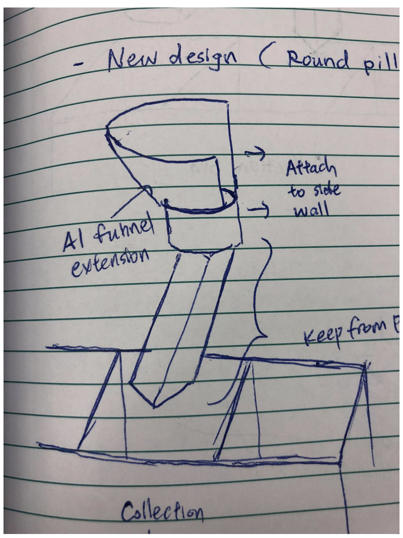



Figure 4.17: New design for the round pill distribution channel, with a large, semicircular funnel at the top.

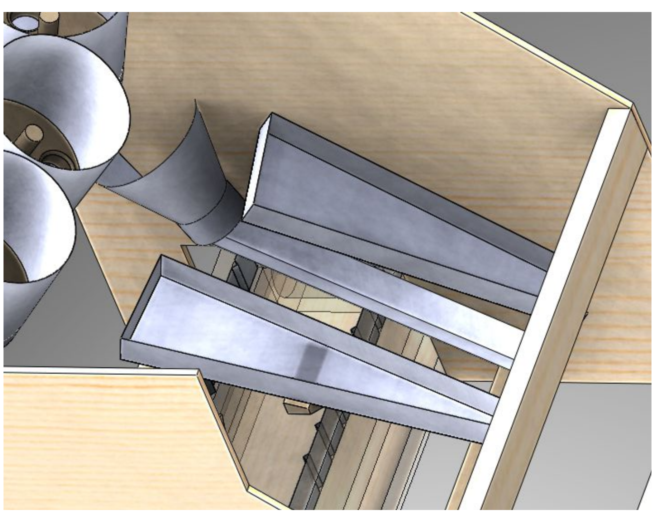


Figure 4.18: New trapezoidal design for the F and L pill distribution channels.

Additionally, as the pill distribution funnel was constructed to be too small for all three cylinders to fit directly over top, additional rectangle-shaped channels were constructed and glued to the structure to direct pills from the cylinders into this funnel (Fig. 4.19). This, again, was an easier option to taking the machine apart and reconstructing the funnel and reposition various other components.

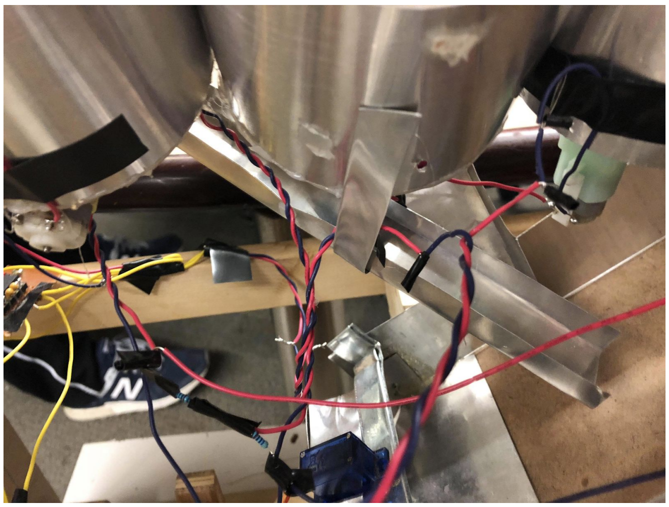


Figure 4.19: Additional channels constructed to direct pills from the dispensers to the distribution funnel.

**4.6.5. Pill Counting**

The original mechanism for pill counting was the contact sensor, which consisted of an aluminum sheet that would make contact with two fraying wires to complete a circuit for every 360 degree rotation of the cylinder (Fig. 4.20), signaling the microcontroller that pill has been dispensed. However, this design was unreliable as it was found through testing that when used for the flat and long pill cylinders, only 64% and 71% of the time did it successfully detect a full rotation, respectively (Appendix D-3). The cause of this is poor contact between the aluminum and the wires, which proved to be difficult to fix through simply repositioning the components. In addition, usage of these contact sensors meant that an additional set of contact sensors (4.3) must be installed on ramps (4.6.4) to count the number of pills left at the end. This was decided to be undesirable due to difficulties encountered in installing the conveyor belt contact sensor (4.6.6). Moreover, a full rotation did not always correspond to exactly one pill being dispensed since there can be multiple or no pills in the hole in the rotating disk. Thus a decision was made to discard this design.

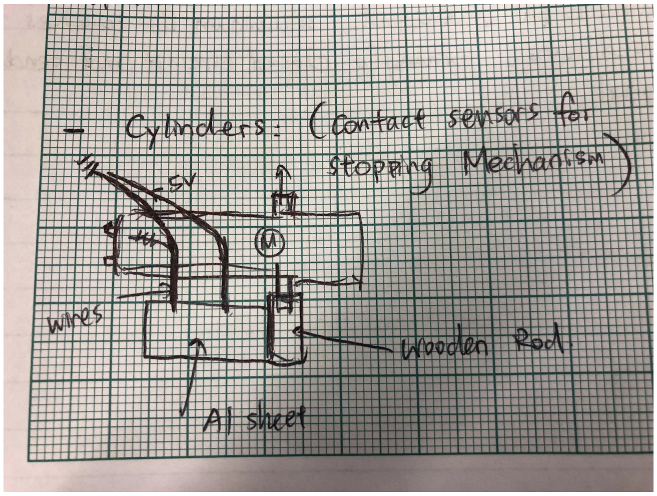



Figure 4.20: Drawing of the contact sensor used for the dispensers. The aluminum sheet makes contact with the two wires every 360 degree turn, sending an interrupt to the PIC.

The IR sensor consists of an emitter, which uses a 5V voltage source emits a strong IR signal, and a receiver, which picks up the signal and sends an analog voltage proportional to its intensity (Fig. 4.21). An object between the emitter and receiver causes a voltage drop, and this is used to detect when a pill drops. With use of IR sensor, pill counting is no longer dependent on the assumption that exactly one pill is dispensed for every 360 degree rotation of the cylinder; rather a pill is registered only when it does drop. This eliminates the bad-contact problem and allows greater room for error on the cylinders part. Through testing it was found that the sensors consistently detected the drop of a single pill over 95% of the time for all cylinders (100% for long pills). One disadvantage is that slight displacements of the emitter and receiver result in vastly different voltage readings. However, secure attachment with glue and re-determining voltage thresholds after moving the machine every time resolved this problem. Another disadvantage was that having to read analog signals, external interrupts could no longer be used (this will be further discussed in section 6.3.3.4), meaning that it was impractical to run all three cylinders simultaneously. The solution to this was discussed in section 4.6.3.

Figure 4.21: The new IR sensors used for pill detection and counting

### 4.6.6. Conveyor Belt

Two significant changes were made to the conveyor belt since the Proposal. First, it was found that when there exists too much friction, more than what the DC motor could provide, the belt would slip against the driver roller, despite the traction tape. Thus, it was decided that an extra support roller would be built right under the driver roller (Fig. 4.22). It would also be covered in traction tape like the other rollers. The conveyor belt would go through the space between the two rollers, which would make it a lot harder to slip due to increased friction between the two rollers.

Figure 4.22: The double-roller system. The fabric goes between the two rollers, and the friction keeps it from sliding.

Second, the shape of contact sensors on the conveyor belt structure and pusher block were altered multiple times. Initially, it was found impossible to put the contact sensors under the conveyor belt due to the difficulty of manufacturing, so the contact sensors were moved to the two sides. In order to stop the DC motor, the conductor on the pusher block needs to make good contact with the parallel contact sensors on both inner sides of the conveyor belt.

However, it was found through testing (refer Appendix D-6 for results) that this design was quite unreliable since oftentimes, wires attached to the pusher block does not have enough contact surface with the aluminum flaps of the contact sensors. This causes the conveyor belt to skip stops. Therefore, instead of using exposed wires, an aluminum ring-shaped contact surface was used on the pusher block (Fig. 4.23), which would theoretically be more flexible and durable. Yet, from further testing, this design was still quite unreliable and the conveyor still skips stops (refer to Appendix D-6 for results).

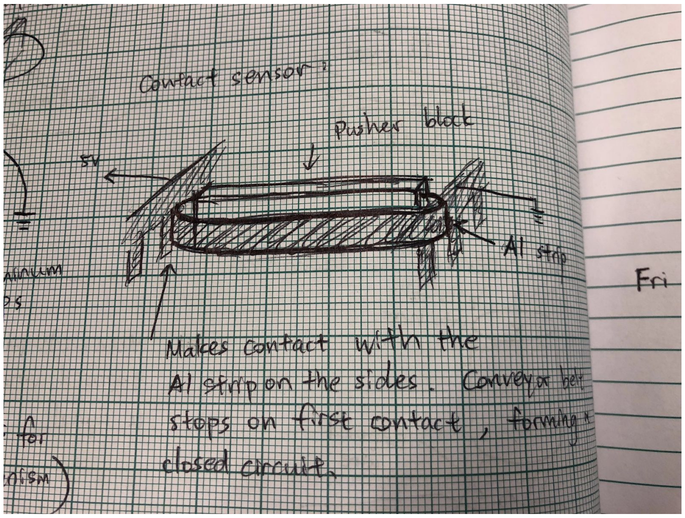



Figure 4.23: Replacing the single wire stretching across the pusher block with a aluminum strip wrapped around it.

To further improve the reliability of contact sensors, they were made into triangular shapes instead of aluminum flaps (Fig. 4.24). The triangular edges would allow contact to be made easier. However, the ring-shaped conductor on the pusher block sometimes still fail, especially after it deforms following several back and forth movements. As a result, a longer pusher block was used, with multiple wires attached to it and stripped at the edges (Fig. 4.25). The wires are multi-stranded, which provided adequate contact surface with the contact sensors.

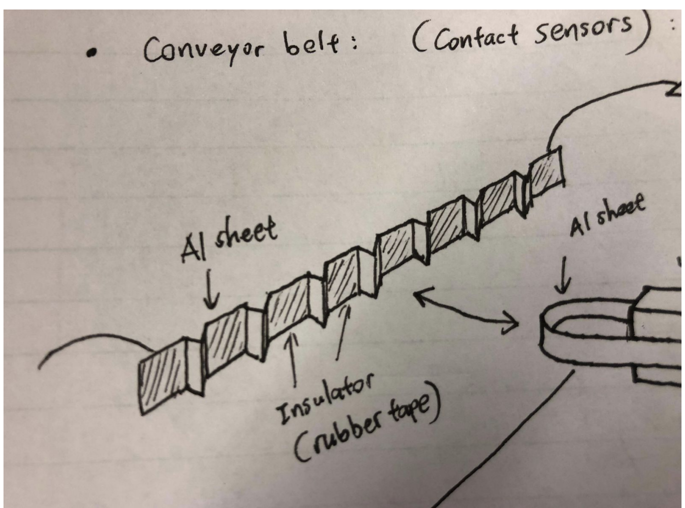



Figure 4.24: New shape for the contact sensors: strip of aluminum with triangular edges sticking out.

Figure 4.25: The new pusher block with multiple stranded wires stretched across it.

### 4.6.7. Power Supply

In order to reduce weight and cost, a smaller power supply was used (Fig. 4.26). It was originally calculated that the sum of all stall currents in addition to the power draw from the PIC and the arduino nano exceeded 5A; however, due to the fact that there is no point in operation where all motors are run simultaneously, there is no need for the larger power supply. We have instead opted for a 5A 12V supply instead, as this was able to provide the required power.

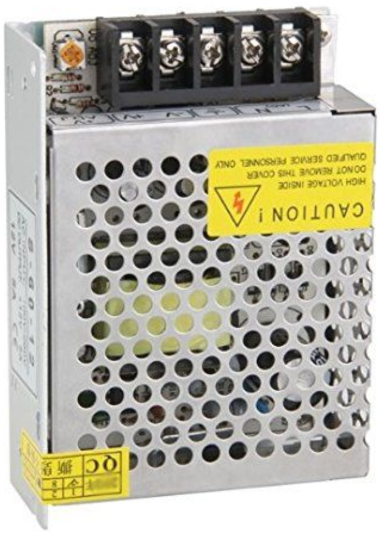
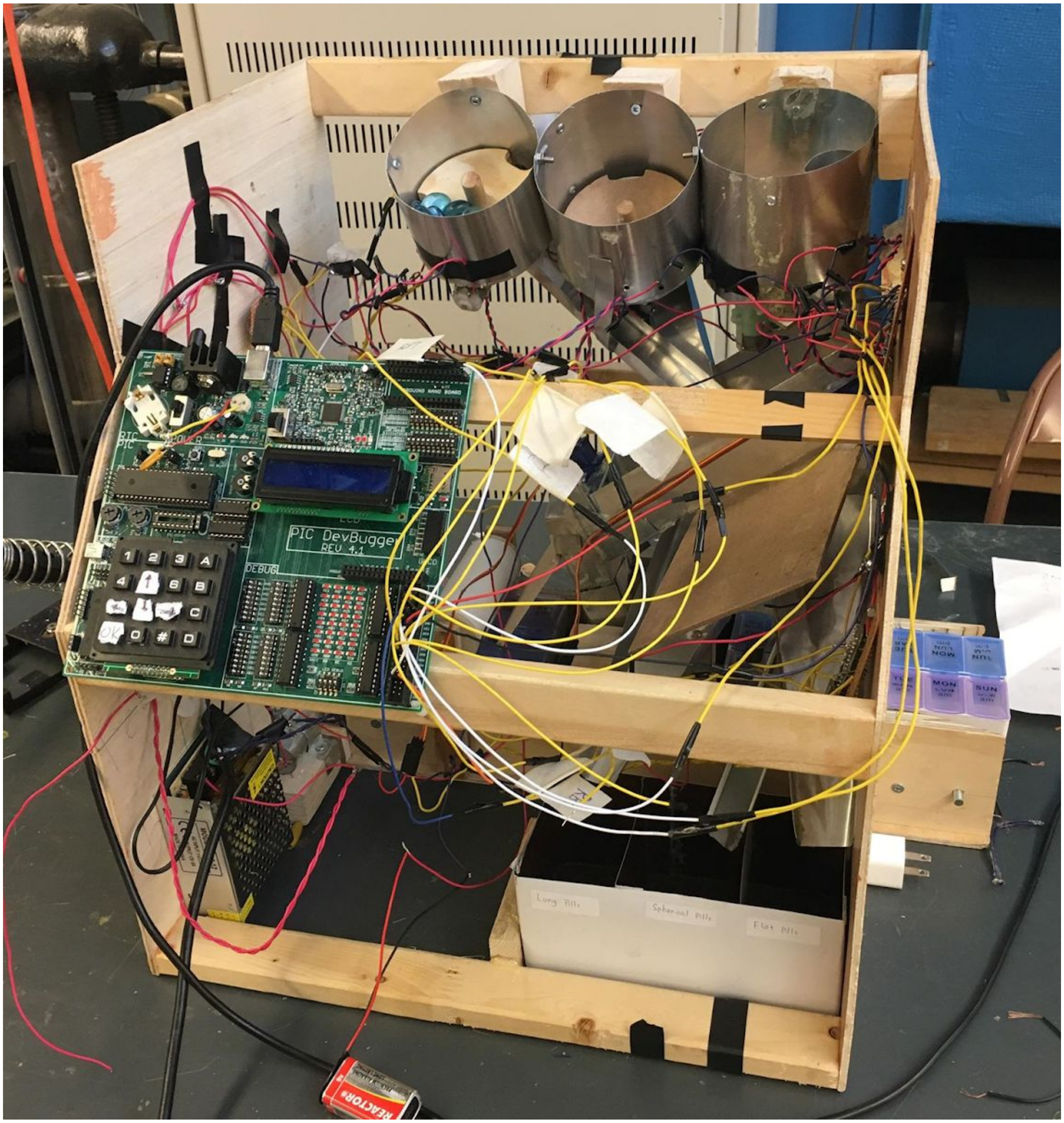



Figure 4.26: The power supply used

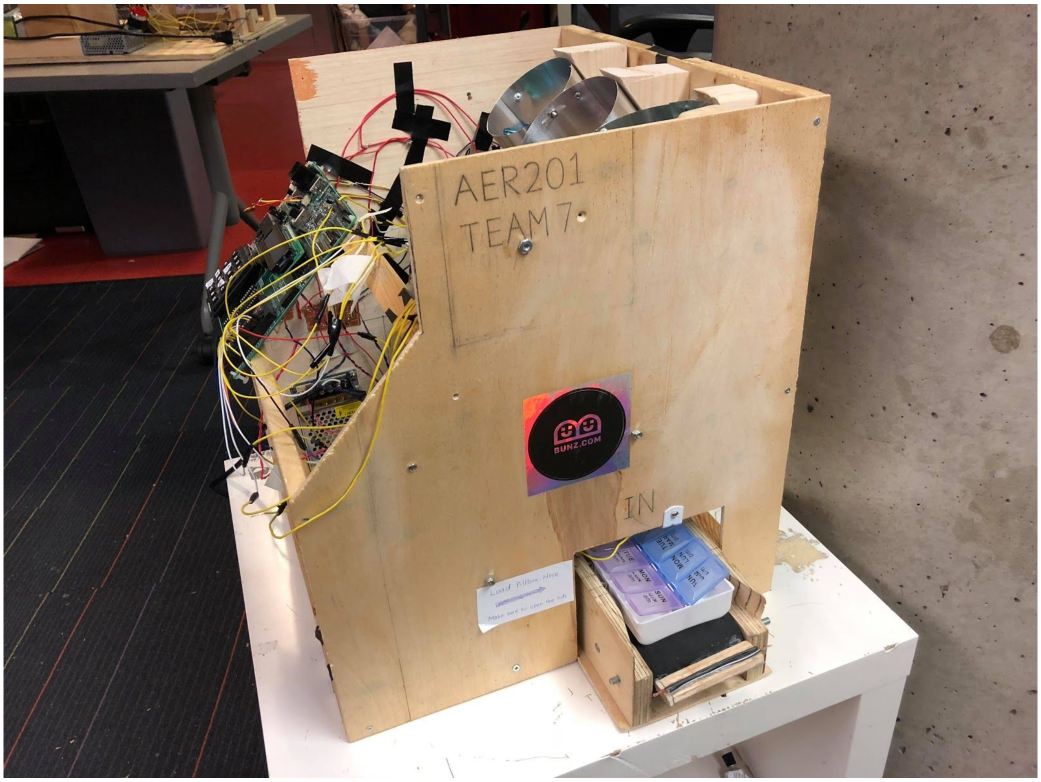
# 5. Solution: Description of Machine

## 5.1. Photos and Diagrams

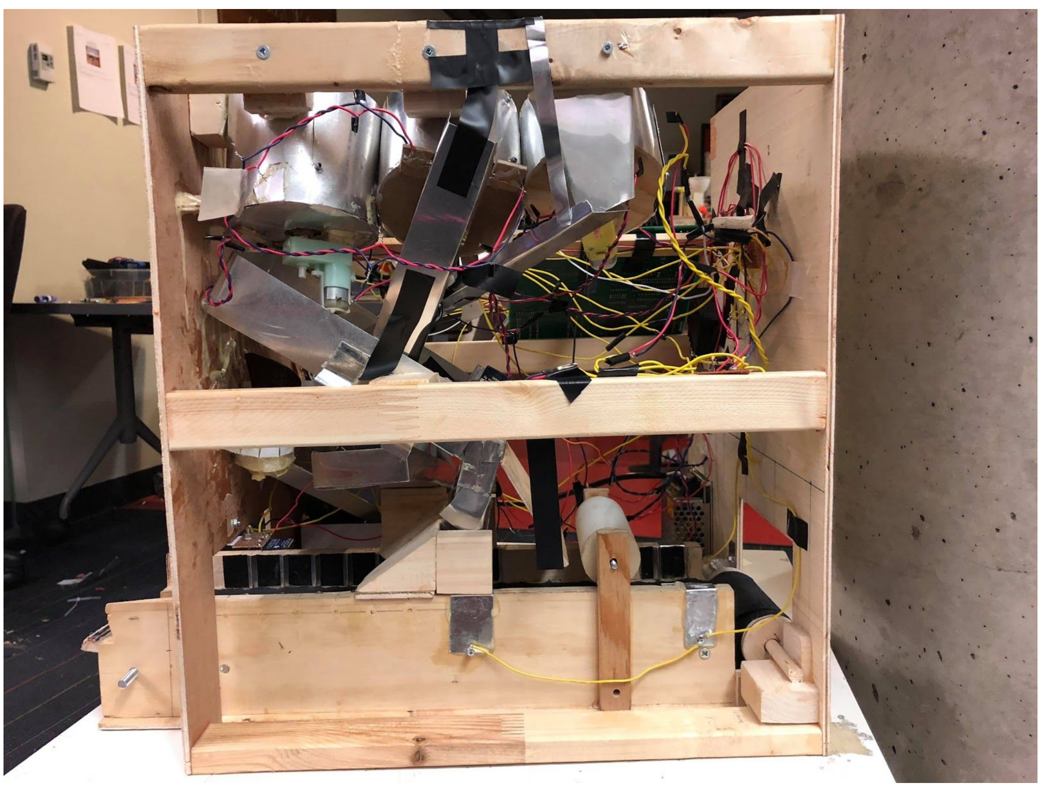
### Oblique View



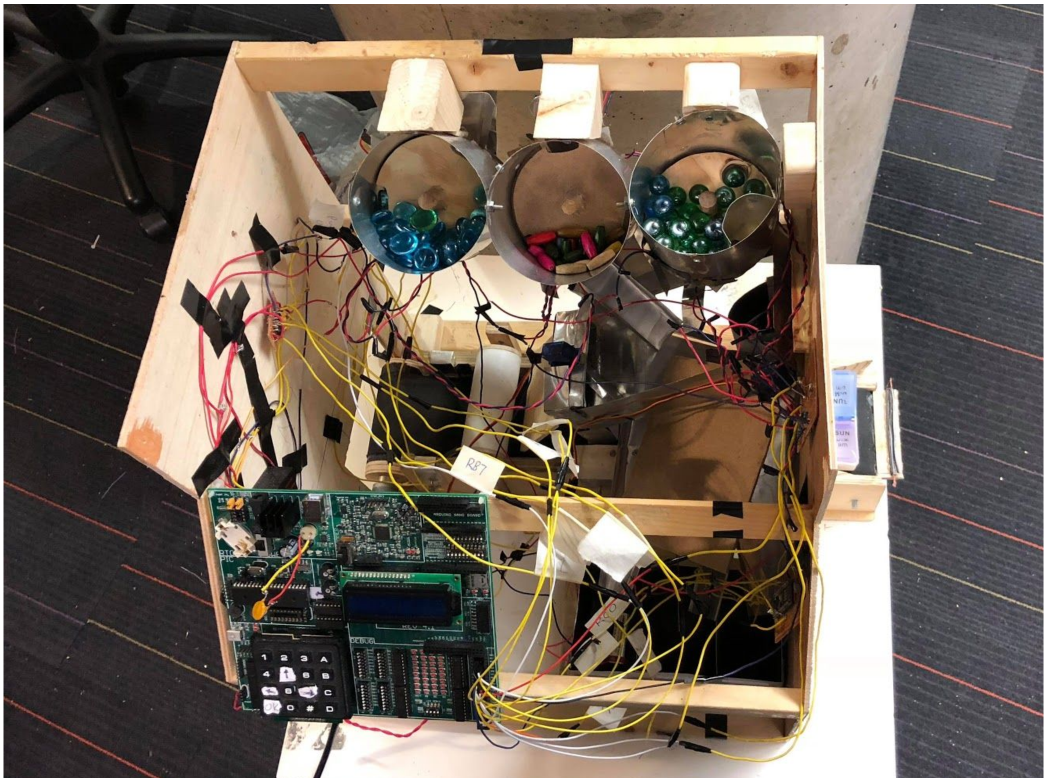
**Side View**



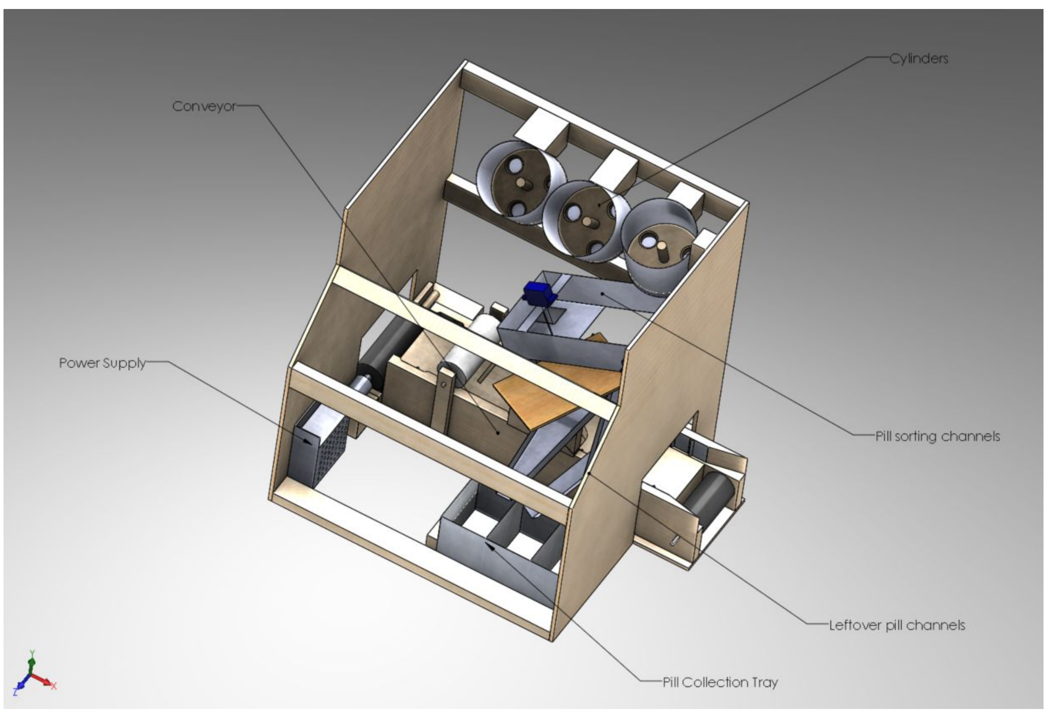

**Back View**



**Top View**



**Solidworks Model Overview**



## 5.2. General Specifications

These statistics below are some general specifications for the machine prototype. For a more detailed chart, please refer to Table B. All of these conditions satisfy the constraints outlined in the RFP.

| **Category** | | **Specification** | **Constraint in RFP [1]** |
|---|---|---|---|
| Dimensions | Length | 49.5 cm | ≤ 50 cm |
| | Width | 42.0 cm | ≤ 50 cm |
| | Height | 45.5 cm | ≤ 50 cm |
| Weight | | 5.14 kg | ≤ 6 kg |
| Total cost: | | $210.43 | ≤ $230.00 |
| Time takes to load pills, load pillbox, and enter prescription | | ~35 s | ≤ 1 minute (60 s) |
| Total runtime per operation* | | Up to 3 minutes and 6 s | ≤ 3 minutes (180 s) |

*Note: This is only the time elapsed between when the user have finished entering the prescription and pressed start to the point when the pillbox and remaining pills are available for retrieval on one standard operation (for one pillbox only).

## 5.3. Standard Operating Procedure

The following flowchart (Fig. 5.1) is an overview of the actual operating procedure of the machine, including everything from user interface to when the user picks up the box.

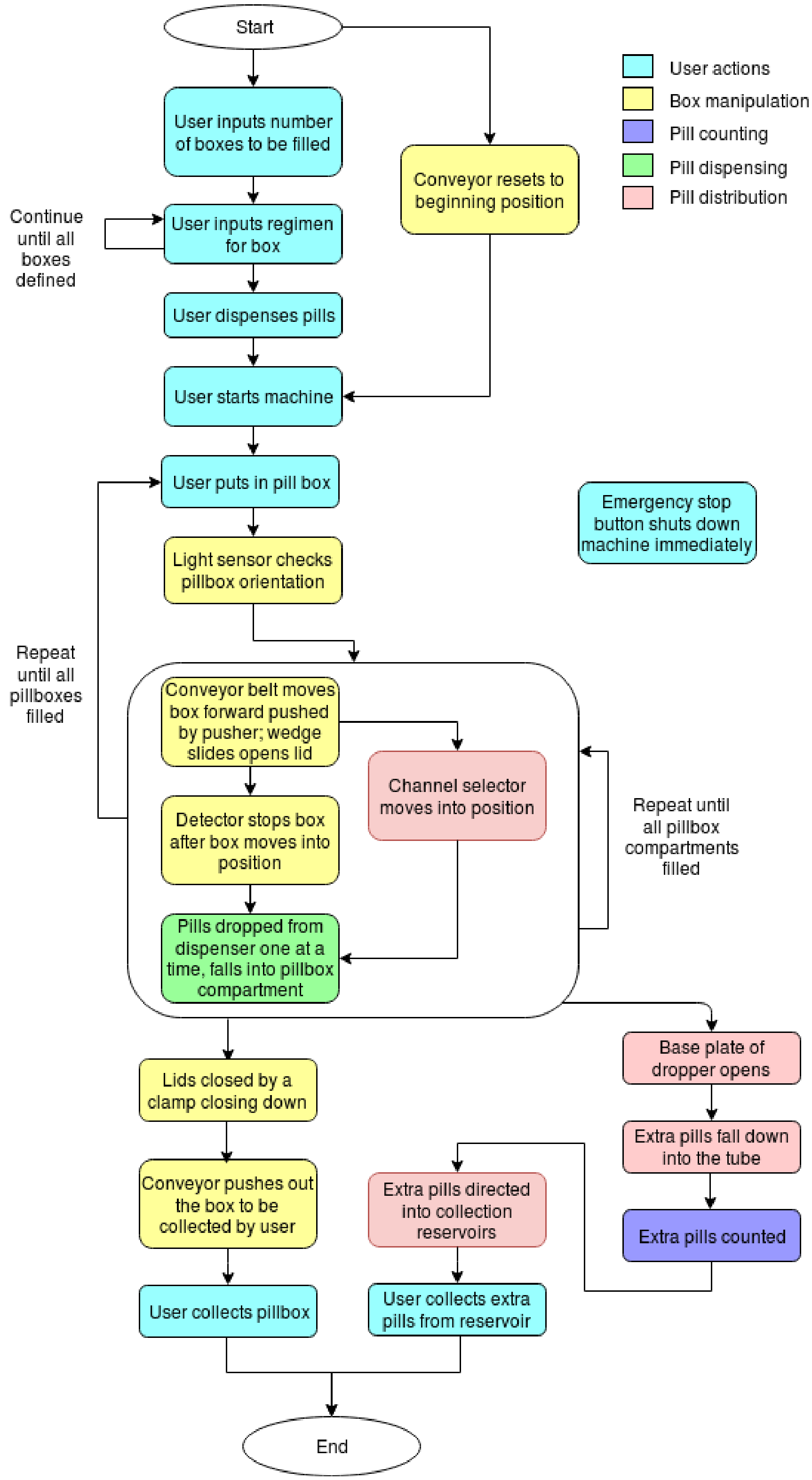



Figure 5.1: Actual flowchart representing the functional process of the machine.

### 5.3.1. Starting and Loading

The user first loads the three types of pills into the three cylindrical dispensers at the top right corner of the machine. The cylinders are labelled according to the type of pills they process: the long pills on the left, the spherical (round) pills in the middle at the corner, and the flat pills in the cylinder at the front.

For the pillbox, the user simply loads it into the machine by placing it along its length onto the conveyor belt from the right side (viewed from the front), and pushing it slightly inside (Fig. 5.2). The conveyor belt extends out of the frame on the right side for convenience. At this point, the pusher block (further described in section 6.1.2) would already be behind the box as its position is reset after every run, and be ready to push the box through the conveyor.


Fig. 5.2: The method of loading the box into the machine.

The main power switch is located on the left wall of the machine (as viewed from the front). This switch also serves as the emergency power switch. Upon switching on the machine, the LCD on the PIC DevBugger Board displays messages to signal the user to enter operating instructions.

### 5.3.2. User Interface

The user interacts with the machine via a 4x4 keypad on the PIC DevBugger board, and information including option selection, machine status, and process summary is communicated to the user through an LCD, also attached to the board. The user interface is further discussed in section 6.3.2, under the Microcontroller subsystem.

### 5.3.3. Box Movement

Once the user starts the operation, the RGB colour sensor located above the lids of the pillbox inside the entrance (Fig. 5.3) is read to determine the box orientation. Information is sent from the sensor to the Arduino then to the PIC board, which adjusts pill distribution directions if the box is loaded in the non-default direction (in which the blue side is closer to the user; section

6.3.3). The conveyor belt then starts moving forward, taking the pusher block with it, until it hits the first contact sensor and the PIC sends the stall command.

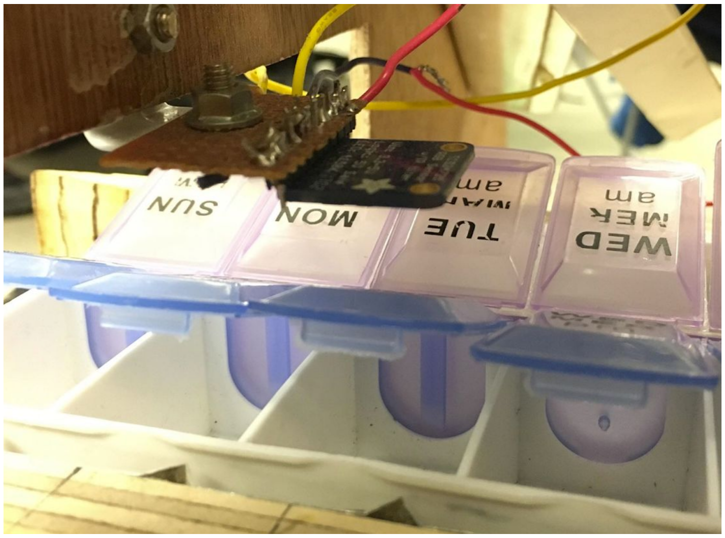


Figure 5.3: The location of the RGB color sensor within the machine.

**5.3.4. Pill Dispensing and Counting**

Each time after the pillbox stops, the three cylindrical dispensers (refer to Fig. 4.16) rotate to dispense the appropriate amount of each pill. Pill counting is performed by IR sensors, discussed in section 6.3.3.4, and directing of pills into the correct compartment of the box is performed by the pill distribution subsystem, described in section 6.1.5.

**5.3.5. Extra Pills and Ending**

Once the pillbox is fully loaded, the bottom of the collection funnel rotates 90 degrees to the open conformation (Fig. 5.4). The three cylinders then rotate continuously to unload all the remaining pills, which fall into collection channels (6.1.5) under the funnel (Fig. 5.5) and are counted by IR sensors. The three channels direct their respective pill type into the collection reservoir, which is an open-topped box with three compartments, one for each type of pill (Fig. 5.5). After all pills are dispensed, the cylinders stop rotating, the box is pushed out to be collected by the user, the conveyor is reset, and information about the run is displayed on the LCD.

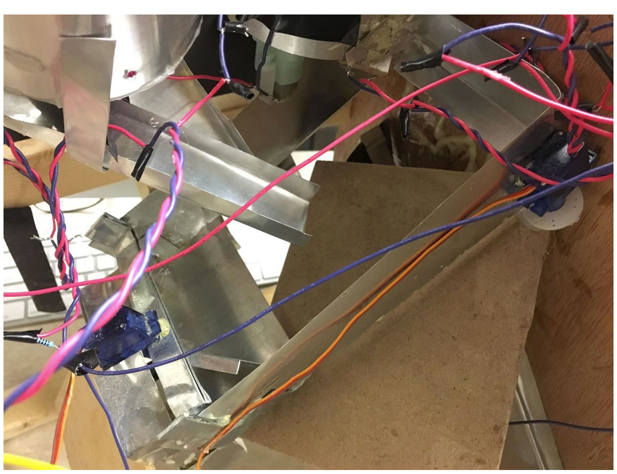


Figure 5.4: The base plate in the progress of turning to the open conformation

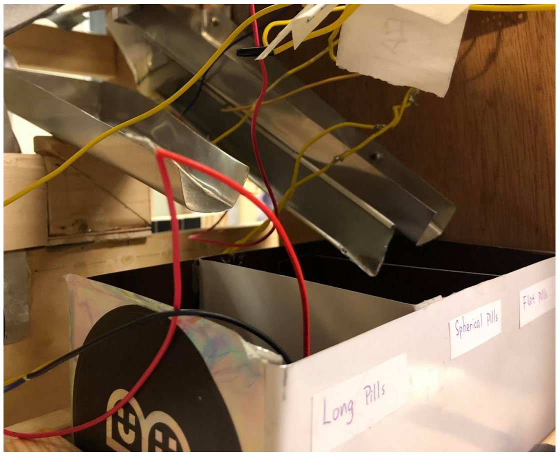



Figure 5.5: collection channels under the distribution funnel

## 5.4. Additional Features

### 5.4.1. Modularity and Easy Access

To improve the modularity of the machine, the machine has two separable main structures: the frame and the conveyor belt (Fig. 5.6). The structures can be separated by removing the conveyor belt first, and then lifting the frame straight up from the conveyor belt structure. To put the two parts back together, the user simply aligns the left end of the conveyor belt structure with the left opening of the frame (viewed from front), and the right opening with a notch cut near the right side of the conveyor belt structure, before pushing the frame down to lock it in place. However, before the structures are put together, the conveyor belt should be put through first, as it is impossible to install the conveyor belt once two structures are together. This feature would significantly improve the ease of making repairs or installing additional parts. A possible improvement would be to make the frame into two separate parts, with one wall in each part. Also, with no walls on the front, back, and top sides of the machine, repairs can be easily carried out.

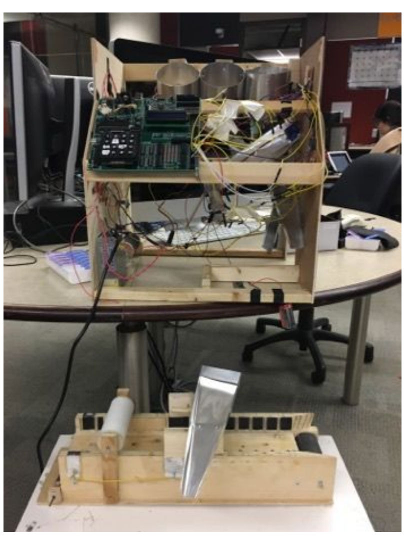


Figure 5.6: The two major components of the machine

### 5.4.2. Repeatability

The machine has an option to handle multiple pillboxes consecutively, given that they have the same prescription. In the user interface, there is a screen that allows the user to enter the number of pillboxes. If there is more than one box, after the previous pillbox is filled, instead of discharging all the remaining pills into the reservoir, the cylinders do not rotate, while the conveyor belt resets to the starting position. The user can then load another empty pillbox and press start on the control panel.

# 6. Solution: Subsystem Details

## 6.1. Electromechanical

### 6.1.1. Frame and Structure

This machine has a mostly wooden structure, made out of generally softwood and plywood. There are two main structures to this machine: the frame and the conveyor belt structure. As mentioned in Section 5.6.1, this feature renders the machine more modular and accessible.

The frame is composed of two identical wooden boards made out of 0.25” plywood, shaped strategically to allow for a slanted area to mount the PIC board, and have a opening at the bottom for the conveyor belt structure, as well as inserting and removing the pillbox (Fig. 6.1). The two wooden boards stand up vertically and are linked together with 6 cross beams (5 beams using 1×2 wood and another using 0.5×1 wood), each 40 cm long. They are generally placed evenly apart to distribute the forces. This structure encloses all the main components of the machine except for the conveyor belt. Also, on the bottom right of the frame are two extra pieces of wood that form an enclosure to place the removable collection reservoir for extra pills.

Attached to the frame are the power supply (on the lower left), the three dispenser cylinders (on the top right), the pill distribution mechanism (from the right wall), and the PIC board (slanted between two beams in the front left). They are attached using screws and nuts, as well as epoxy glue. In some cases, both are used in conjunction to prevent loosening.

The conveyor belt structure not only supports the conveyor belt (discussed in detail in the following section), but also serves as a structure to hold other components. Located at the bottom at the machine, it also performs the function of a base, which is relatively heavy and helps stabilize the overall machine.

The structure is composed of a 14 cm × 49 cm × 0.25” baseboard, two 9 cm × 45 cm × 0.5” vertical supports, and a 9 cm × 40 cm × 2” centre core (Fig. 6.2). At the two ends are two rollers made out of PVC pipe, and another supporting roller below the driver motor on the left (viewed from front). The conveyor structure opening on the right end is widened by removing a triangle on the inner sides of the vertical support boards to allow for easier loading of the pillbox.

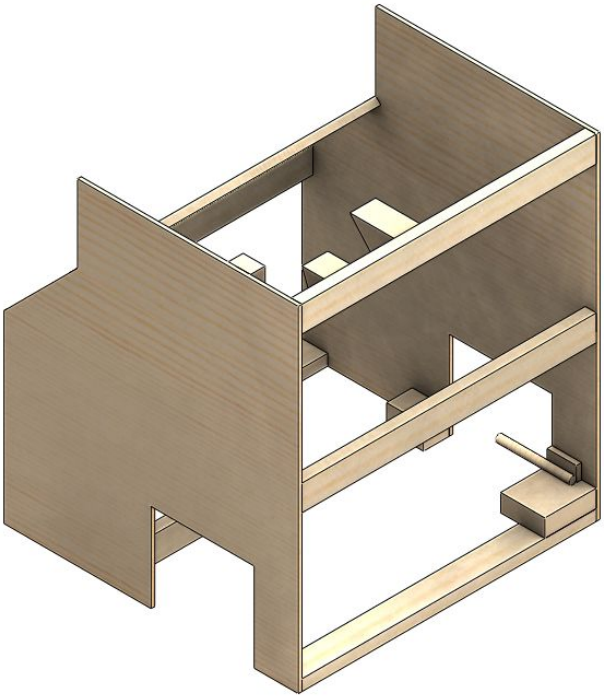


Figure 6.1: The outer structure of the machine

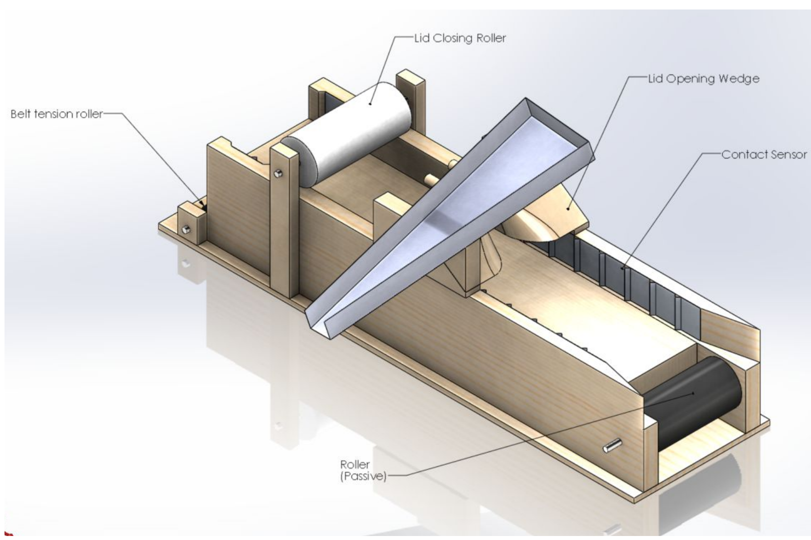



Figure 6.2: Structure of the conveyor belt

**6.1.2. Conveyor Belt**

A conveyor belt driven by a DC motor is used to move the pillbox relative to the stationary pill dispensing system. Since a lot of friction is generated as the box comes into contact with the lid-opening wedges (4.2.3) and lid-closing roller (4.6.2) (up to 2.25 kg*cm based on testing with a force-meter; see Appendix D-7 for results), a powerful motor is required. A DC motor with torque of 7.9 kg*cm was used. Also, since there is little friction between the bottom of the pillbox and the conveyor, a pusher block (Fig. 6.3), measuring 8.5cm × 1.5cm × 2.5cm is attached to push the pillbox through all these components. The pusher block is glued onto the conveyor belt.

Fig. 6.3: Image of the pusher block with the conveyor belt

In order to stop the DC motor that controls the position of the pillbox, contact sensors made out of 0.025” aluminum sheets were installed onto the side walls of the conveyor belt structure. On both sides, triangular edges extrude out at measured locations where the pillbox should stop (Fig. 6.2, Fig. 6.4). To form a complete circuit while in contact, strands of wires were installed onto the pusher block, with the wires exposed on the two sides. The exposed wires would come in contact simultaneously with the triangular edges on both sides. This sends a signal to the PIC board, which then interrupts the DC motor to stop running. There are 7 sets of triangular edges in total, corresponding to the 7 rows of the pillbox.

The conveyor belt itself is made out of a cotton-based fabric, which is quite durable, flexible, and hard to deform significantly. From when 11.21 lb force, which is much greater than the maximum force experienced by the conveyor during the process, is applied, the fabric only deformed by 3% (see Appendix D-8 for testing results). The lack of deformation is crucial since too much deformation would loosen the conveyor belt, which would result in less traction with the driver rollers.

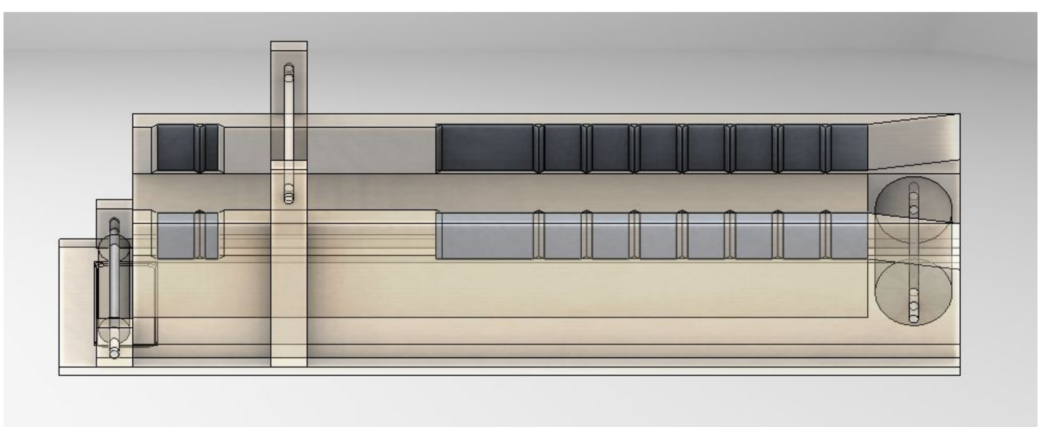


Figure 6.4: Another view of the conveyor structure, showing the contact sensors

The conveyor belt consists of three rollers: one driver roller on the left end (viewed from front) powered by the DC motor, one free roller on the right end, and one supporting roller under the driver roller to provide more traction (Fig. 6.5). The two main rollers are made using 1.5” PVC pipe, and all three rollers are coated with traction tape to prevent slippage from the belt.

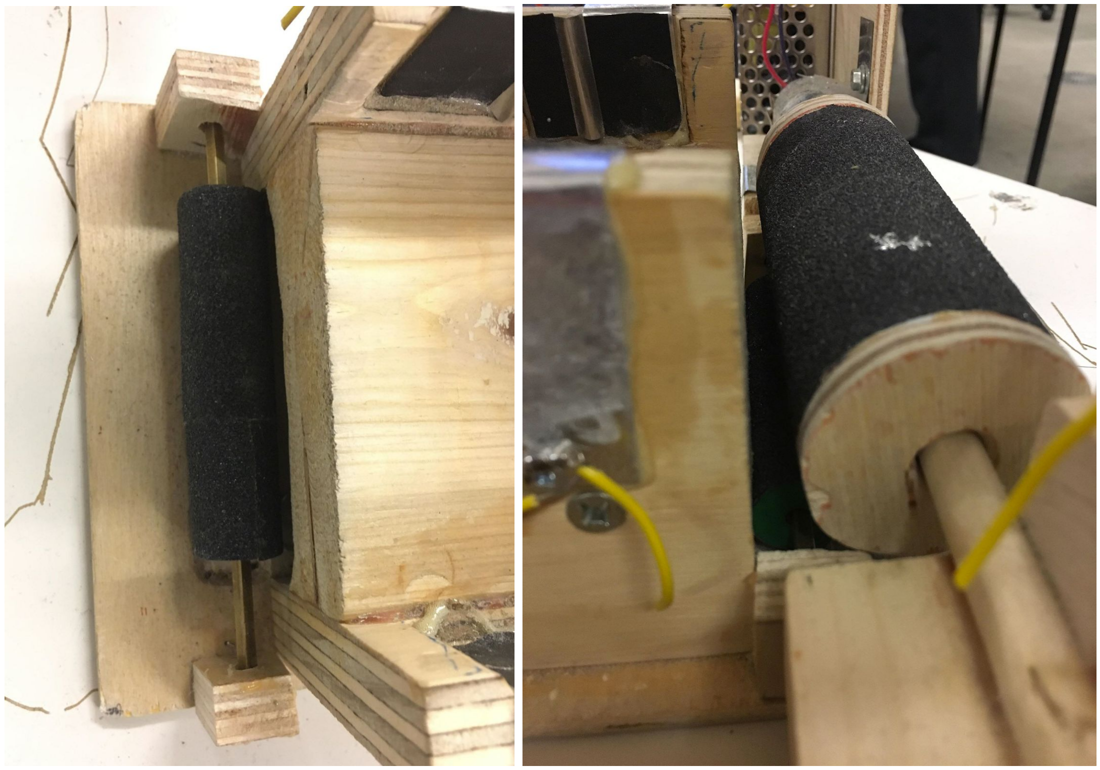


Figure 6.5: View of the rollers in the conveyor belt

The supporting roller is needed as it was found that when the conveyor belt experiences too much tension (over what the DC motor can supply), the driver roller simply slips with the conveyor belt, despite the added traction. To prevent this, a 0.5” roller was mounted just below the driver motor so they are in contact. The conveyor belt goes through the space between the two rollers and gets compressed, and therefore it would be a lot harder to slip. The wooden structure of the conveyor belt is discussed in Section 6.1.1.

### 6.1.3. Lid Opening and Closing Mechanisms

As mentioned in section 4.2.4, the lid opening mechanism consists of two wooden wedges mounted on the conveyor belt structure (Fig. 6.6). The wedges force the lid to be perpendicular to the pillbox for the pills to drop through. The sides where the lids touch the wedges are smoothed out to decrease friction as much as possible. Two wooden dowels extend out from the inner sides of the wedges to hold open the lids and to create enough space for loading the pills into the compartments.

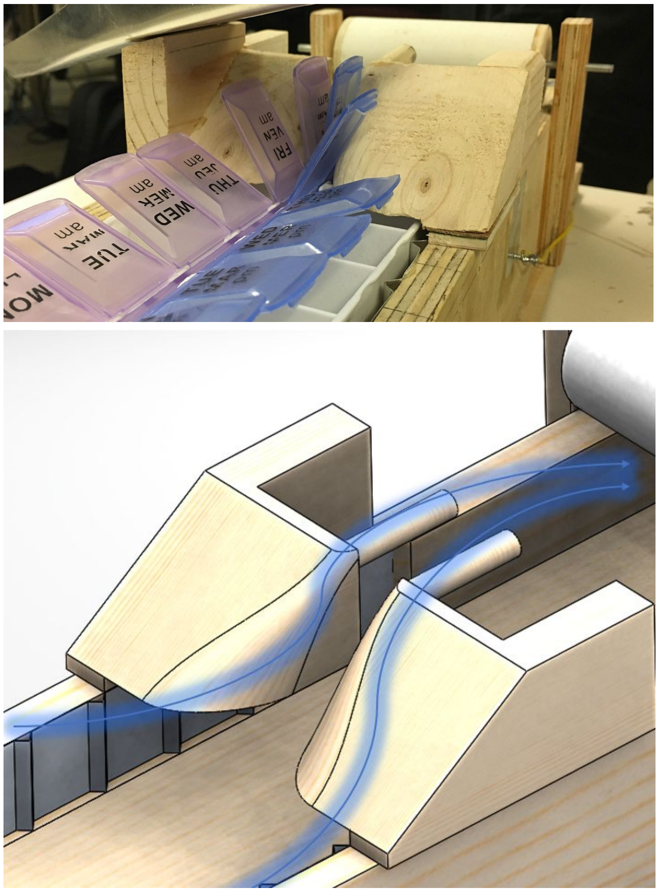


Figure 6.6: Different views of the lid-opening wedges. The blue lines in the second image represent the approximate paths taken by the top edges of the lids.

After the pills are loaded, the pillbox passes through a slanted plate (made out of a sheet of 0.25” plywood) that lowers the lids. The reason for these plates was explained in section 4.6.1. The box then passes under the roller (made out of 1.5” PVC pipe), which offers just enough compressive force to lock the lids, but not too much to generate excessive friction that would prevent the pillbox from moving (Fig. 6.7). It also has a hard surface, which, from testing, makes it easier to close the lids compared to a soft surface like rubber or sponge. This system ensures that all lids are closed and locked properly when the pillbox is returned to the user.

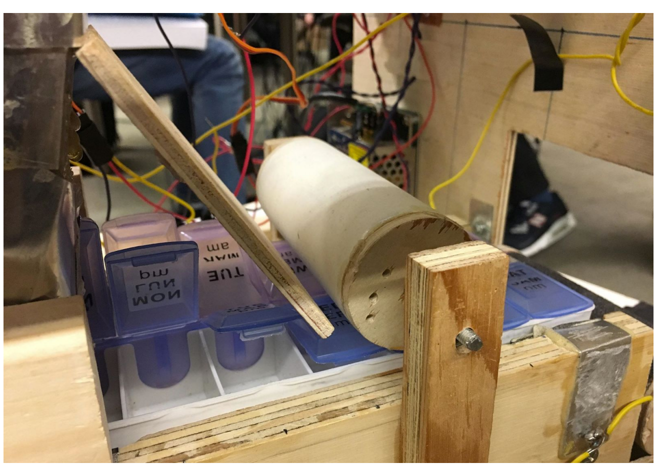


Figure 6.7: Image of the slanted plate and roller working together

### 6.1.4. Pill Dispensing Cylinders

The three cylinders on top of the machine allows the users to load the pills and then dispenses them one at a time to the channels below, which allows them to be counted one by one. Each cylinder is designed differently to better handle a certain type of pills based on its shape and other physical characteristics.

The general design of the cylinders include an outer shell made of 0.025” aluminum sheets. They are around 9 cm tall and 9 cm in diameter, which makes them large enough to handle up to 45 pills of each kind. The bottom of the cylinders has two parts: a lower stationary base that has an opening for the pills to fall out, and an upper rotating disk attached to a DC motor through a wooden dowel at the center, that has three evenly-spaced holes cut into them to allow one pill to sit inside. While the disk rotates, the pill would fall and sit inside one of the holes, get carried around to the opening on the bottom plate, and gets dropped. A pair of IR sensors located right beneath the opening registers after each pill falls through, and counts the number of pills. All three cylinders are attached to the frame using screws and custom-made wooden anchor blocks. In addition, for better control of the speed of rotation, a PWM signal is used to power the DC motors from the Arduino Nano. The non-electromechanical decisions are discussed later.

To minimize the possibility of error, the tilt of the cylinder, the thickness and the shape of the holes of the rotating disk, the shape and location of the opening of the lower base, and the interior structure of the cylinders, are designed differently to account for physical differences between the three types of pills.

For spherical pills, the disk is about 1 cm thick, and has circular holes cut into them that are not located by the edge (Fig. 6.8). To allow the pills to fall in easily, the side of the holes adjacent to the outer edge is rounded. As the spherical pills gather easily at the bottom, this cylinder is only tilted at about 15 degrees. Also, since the opening on the base was not at the highest point (as it is lined up exactly with a collection funnel below that catches extra pills), an aluminum barrier was built around the bottom opening to prevent two pills from falling through the hole at once.

For the flat pills, the disk is also about 1 cm thick; however, the holes cut into a slit shape instead of being circular, which would only allow the pills to stand up vertically instead of sitting flat (Fig. 6.8). The reason for this decision was discussed above in Section 4.6.2. Also, due to problems associated with having a barrier around the bottom opening (see 4.6.2), the bottom opening was moved to the highest position of the base to prevent multiple pills from falling through at once. The cylinder is also oriented at a much steeper angle (about 45 degrees from horizontal) to ensure that the pills slide vertically into the slits of the rotating disk.

The cylinder for the long pills have a similar orientation to that for the flat pills. The disk, however, is a lot thinner (only about 4 mm thick) (Fig. 6.8). The holes on the disks are shaped to carry the pill along its length, and dispense from the highest point of the base.

Figure 6.8: The rotating disk designs tailored to each pill type

As for choosing the motors, it was determined from testing that the motor should supply a minimum torque of 1.2 kg*cm (with a factor of safety of 2x). The test involved pulling a force meter attached to the rotating disk inside the cylinder shell, with 45 pills inside. The yellow motors from the AER201 project kit proved to provide just enough torque at 1.233 kg*cm, and was cheap enough ($5 each) that it can be used for all three cylinders. Having the same motor model decreases the complexity of construction, and allows for easier control.

### 6.1.5. Pill Distribution Mechanisms

The pill distribution mechanisms include two main parts: the slanted channel directing the pills to the pillbox (aka the distribution funnel) and the channels for directing extra pills to the collection reservoir.

The distribution funnel is essentially an open box, tilted at about 25 degrees from horizontal, that is located right below the three dispenser cylinders (Fig. 6.9). The lower end of the box narrows and splits into two channels going down, each one serving one lane of the pillbox as it moves through. In between, a servo controls an aluminum barrier that selects the channel, so pills would only enter one channel at a time (Fig. 6.10). The base of the box is also attached to a servo, which can rotate open once the pillbox is filled and the remaining pills need to be transported to the collection reservoir (Fig. 6.11, Fig. 6.12).

The general structure of the box is made out of 0.025” aluminum sheets, with the rotating base being a sheet of 0.25” particle board. Both servos are 180-degree servos with precise angle control (model: FS-90). The box is attached to the frame using epoxy glue and supported by another anchor block on the back side (viewed from front of machine). The two separate channels extend all the way down to just above the height of the pillbox body, so no pills can spill out when loading. They form a space in between so that the pillbox lids can go through in between. Also, the channels fit right into the space between the wedges, the protection barriers, and the two wooden dowels extending from the wedges.

Figure 6.9: Image of the distribution funnel located below the cylinders.

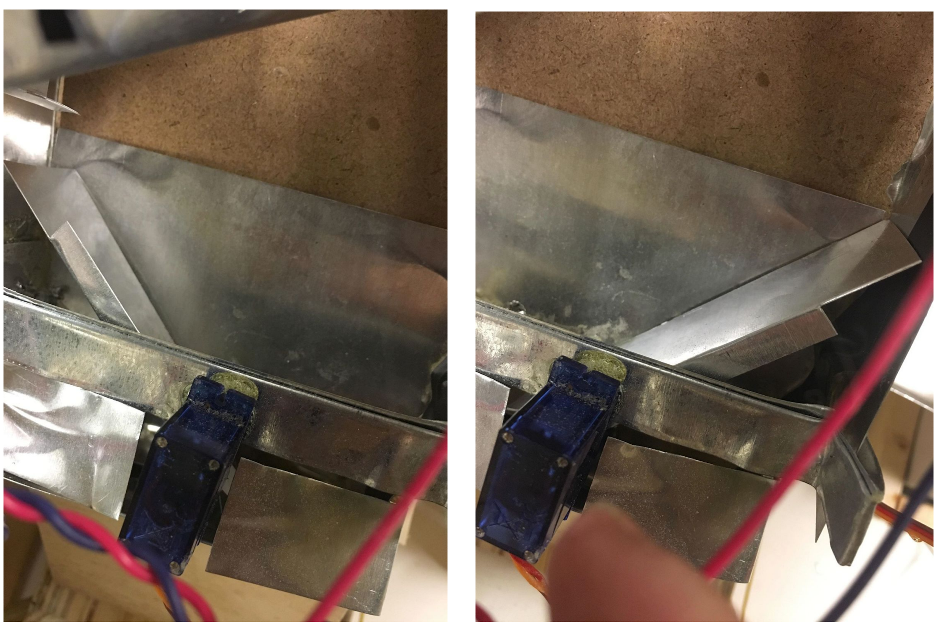


Figure 6.10: Two orientations of the servo controlling which side of the box the pills go to

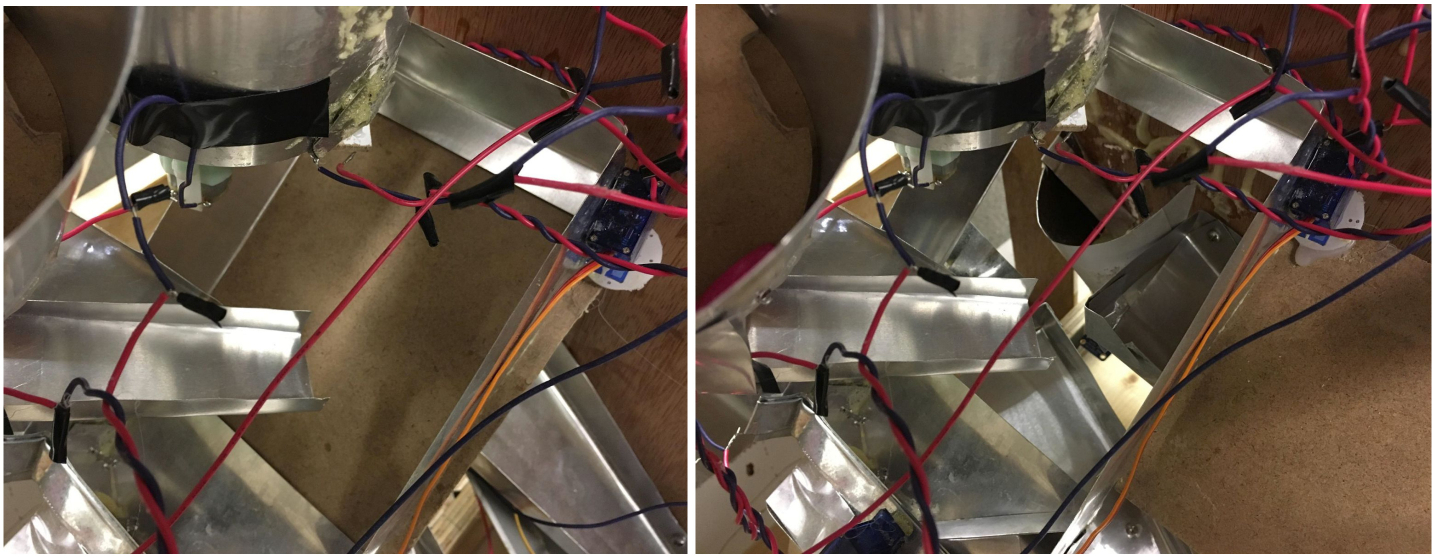


Figure 6.11: Images of the base plate of the distribution funnel. The left and right images show the closed and open conformations, respectively.

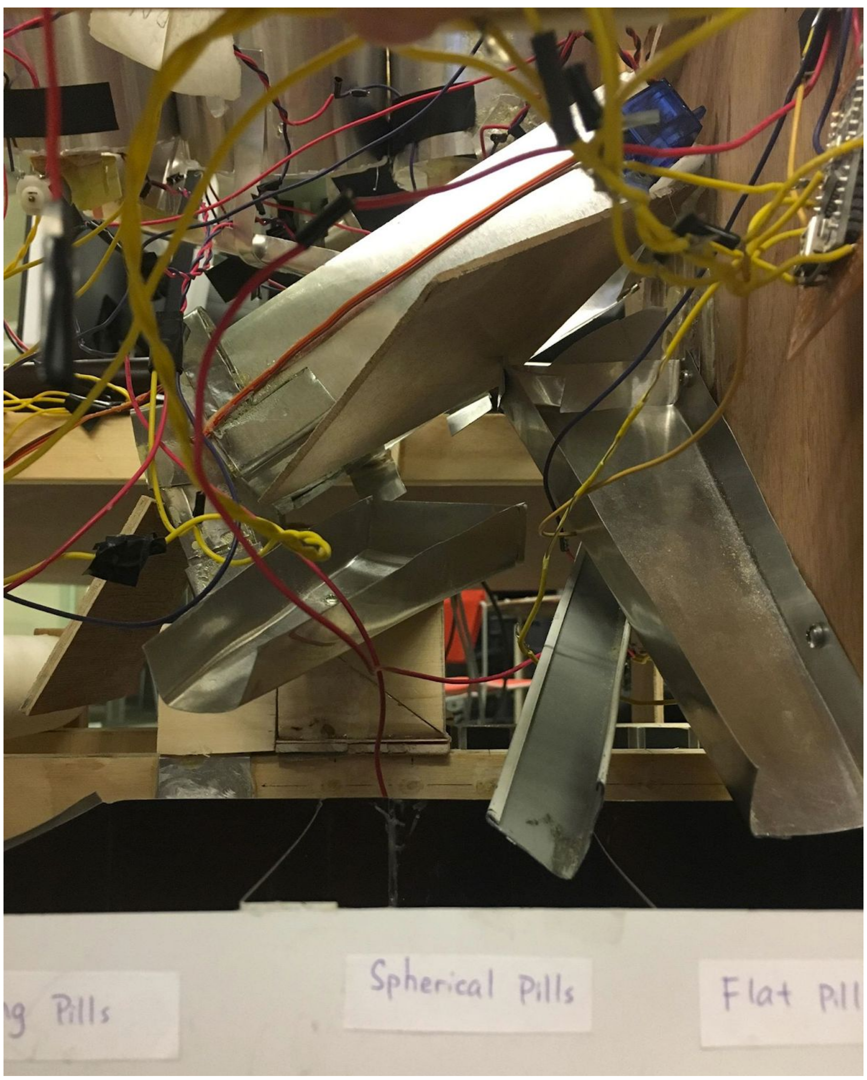



Fig. 6.12: An image of the collections channels and collection reservoir below the distribution funnels, with the base plate in the open conformation.

For extra pills after filling the box, the base of the box rotates open, and pills fall directly from the dispenser cylinders down onto the three separate channels, one for each type of pills, and get directed to the appropriate compartment of the collection reservoir (Fig. 6.12). All channels are set at an angle, and are attached to either the conveyor belt structure or the right wall of the frame.

For long pills and flat pills, the channels are made out of 0.025” aluminum sheets and shaped like a trapezoid with barriers on three sides (Fig. 6.12). The wider base (7 cm wide) is located under where the pills fall down and allows for an area of uncertainty. The narrow side (2-3 cm wide) opens to the collection reservoir. For spherical pills, the channel consists a short section of 1.5” PVC pipe, a V-shaped aluminum channel (1” wide), and a semi-circular funnel attached above the opening of the PVC pipe (7 cm diameter). The half funnel and the right wall of the frame forms a semi-circular area of uncertainty for the pills dropping down.

Finally, the collection reservoir for extra pills is made out of of 0.025” aluminum sheets, divided into three equal sections for the three types of pills. It sits right inside a space created by the frame, under the opening ends of the three collection channels, and can be easily removed from the machine to for the user to retrieve pills. It was tested to make sure that pills would not fly over the top of the reservoir coming out from the channels.

## 6.2. Circuits and Sensors

### 6.2.1. Overview

Power is supplied and regulated through two distinct rails, the 5V and 12V rails, through AC-DC and DC-DC converters. The 5V rail powers the motor ICs in addition to the arduino nano, the color sensor, the servo motors and the IR sensors. The servo is controlled through PWM on the arduino. Communication between the PIC and the arduino is done exclusively through digital pins, while the color sensor communicates with the arduino through SDA/SCL. Further schematics can be found in Figures 6.13 to 6.17 for further wiring and protocol.

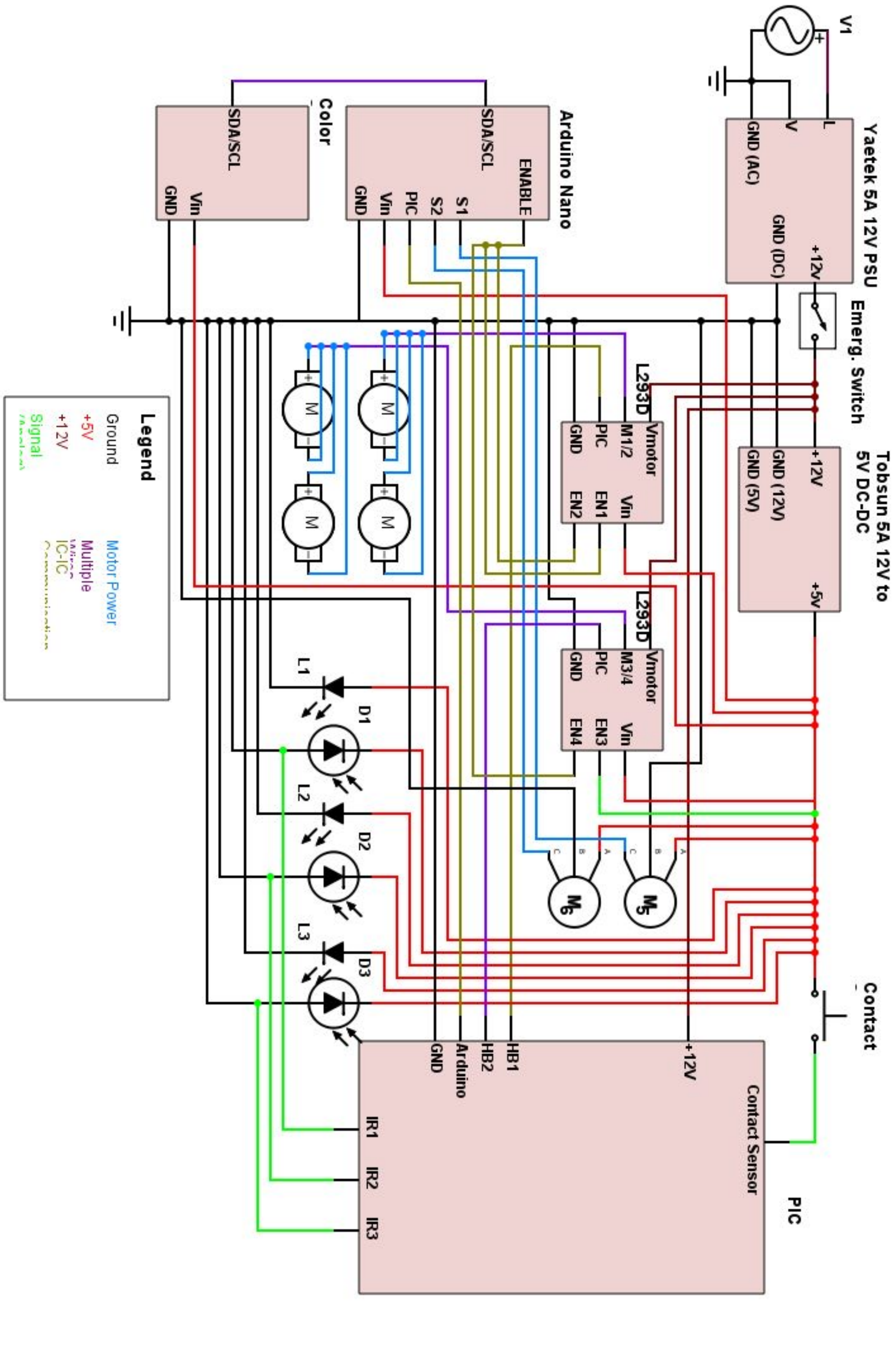



Figure 6.13: Circuit diagram for the entire machine

### 6.2.2. Motor Control

The DC motors are controlled through 2 ICs (L293D) (Fig. 6.14). The arduino connects to the enable bits on the 3 DC motors responsible for pill dispensing, allowing for PWM control of those three motors. The enable bit for the conveyor belt motor is connected to the 5V supply rail in order to provide a constant logical high.

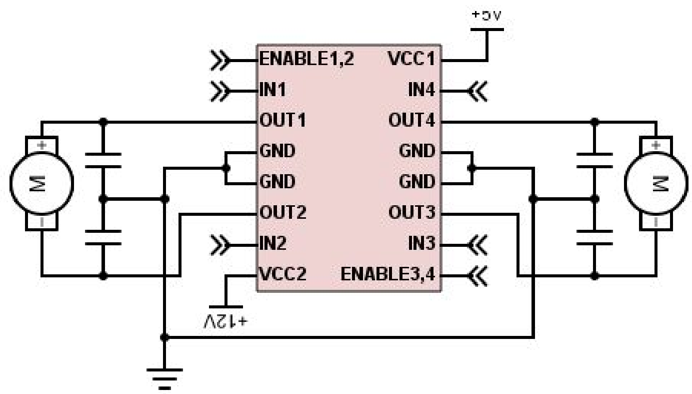



Figure 6.14: L293D H-bridge chips schematic

The servo motors are powered by the 5V supply rail, with PWM provided by the PIC.

| **Pin Assignments - L293D H-Bridge** | | |
|---|---|---|
| **Pin** | **Assignment (HB1)** | **Assignment (HB2)** |
| EN1,2 | [Insert PWM Arduino here] | [Insert PWM Arduino Here] |
| IN1 | PIC | PIC |
| IN2 | PIC | PIC |
| EN3,4 | [Insert PWM Arduino here] | +5V |
| IN3 | PIC | PIC |
| IN4 | PIC | PIC |

### 6.2.3. IR Sensors

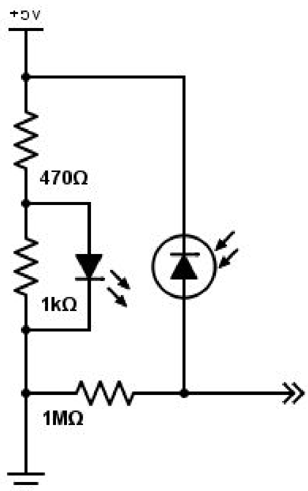



Figure 6.15: Circuit diagram for the IR sensors

The emitter-receiver pair is designed as shown in Figure 6.15. The emitter has a drop of 1.60V, which is regulated by a voltage divider. The receiver has a large resistor at the tip, where a wire runs to the PIC in order to determine the voltage drop across the resistor. A high light intensity provides a voltage drop of 1V, while darkness should see no voltage drop.

### 6.2.4. RGB Colour Sensor

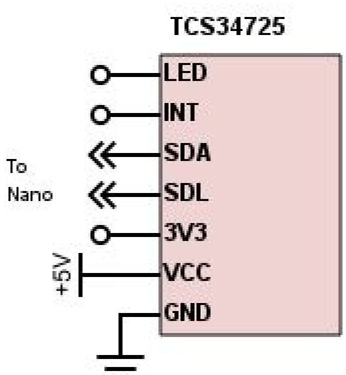



Fig. 6.16: Pins of the TCS34725 color sensor

The TCS34725 color sensor (Fig. 6.16) is designed to detect the orientation of  the box using thresholding on the Arduino (see Appendix E for datasheet extracts). The SDA and SCL outputs I2C to the Arduino Nano, which samples the RGB values every 150ms. The Nano receives intensity values for the Red, Green, Blue, and clear channels; a red value above a certain threshold triggers a logic high on an output pin on the Arduino Nano, which in turn alerts the PIC to the orientation of the box. Pin assignments and arduino code can be found in Appendix F.

### 6.2.5. Contact Sensor

The following is the circuit diagram (Fig. 6.17) for the contact sensor discussed in section 6.1.2.

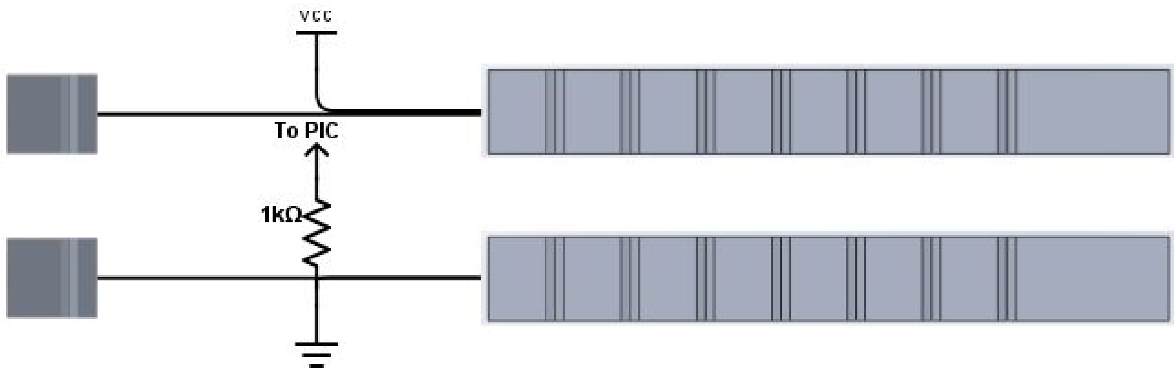



Figure 6.17: Circuit diagram for the conveyor contact sensor

## 6.3. Microcontroller

As described in the RFP, the procedure taken by the microcontroller includes: 1. Interacting with the user, where to user enters prescription and dosage pattern, 2. Control various actuators and sensors in order to package the pill box once the user presses ‘Start’, and 3. Return to a standby mode, and communicate to the user information such as entered options, run time, and numbers of remaining pills.

Two microcontroller boards are used in the machine: the PIC18f4620 and an Arduino Nano. The PIC is responsible for user interactions, controlling various DC motors, receiving and processing signals from all IR and contact sensors, and sending instructions to the Arduino, while the Arduino performs servo control, reads from the TCS34725 color sensor, processes this information and sends corresponding signals to the PIC. The arduino was used for two reasons. Firstly, servo control requires pulse width modulation (PWM), and doing this with the PIC will require the timer2 module. However, this module was already used for timing the machine operation and all other timers were occupied, and thus an Arduino board was necessary. Secondly, communication with the color sensor requires the $I^2C$ protocol, which was extremely tedious to implement with the PIC, while the Arduino library for doing so was already available online. The operational procedure of the microcontroller subsystem consists of three modes: Standby, User Interface, and Packaging. These will be discussed in the sections below.

### 6.3.1. Standby Mode

While in Standby mode, the PIC displays messages to the user via the LCD in a cycle. Each message is displayed for 1.5 seconds as this was found through testing to be a comfortable amount of time for the user to read a message. Timing of the messages is controlled by the timer1 module, which sends an interrupt every 1.5 seconds to change the message. The cycle of messages is as follows (Fig. 6.18):

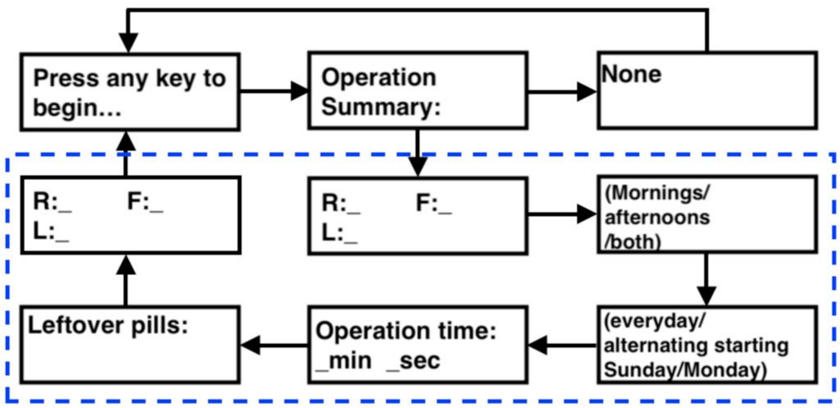
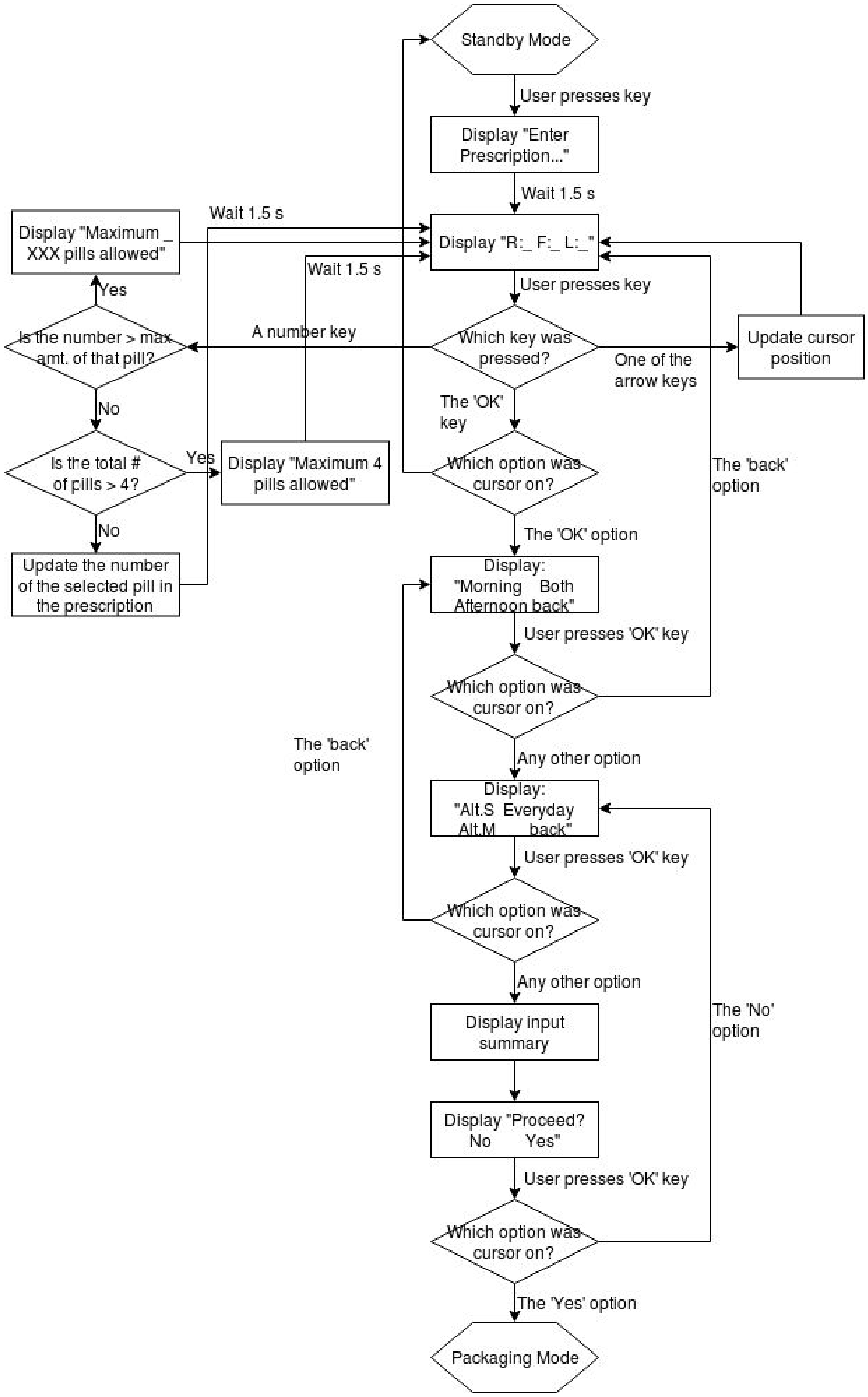



Figure 6.18: The sequence of messages while the microcontroller is in Standby Mode.

Note that the messages outlined in blue are only displayed if the machine has completed at least one run. The machine exits Standby mode when the user presses any key on the keypad, which generates an interrupt on the RB1 pin.

### 6.3.2. User Interface

In the User Interface mode, the user inputs and selects various operating parameters. On the keypad, the arrow keys are used to control the position of the cursor on the LCD and the ‘OK’ button is pressed to select the option that the cursor is located on. Note that these were originally number keys, but have been re-programmed and relabeled to provide these functionalities. In the first screen, the user is asked to enter the prescription. The arrow keys are used to select pill type, after which the user enters the number of pills using the remaining number keys . Selecting the ‘OK’ option brings the user to the next screen, where the user selects the intake times from ‘Morning’, ‘Afternoon’, and ‘Both’ . Selecting any of these options leads to the third screen, which displays the options ‘Alternate starting Sunday’, ‘Alternate starting Monday’, and ‘Everyday’. After selecting an option, the machine runs through a summary of the input parameters, before bringing the user to the final screen, which asks the user whether they would like to proceed . If the user selects ‘Yes’, the machine proceeds to the Packaging mode. At any step in this mode, selecting the ‘back’ option brings the user to the previous screen, where they may correct a previous selection. The flowchart below present this process in greater detail:




### 6.3.3 Packaging Mode

As an overview, the packaging process begins with running the conveyor belt motor until the wire on the pusher block (4.2.2) makes first contact with the conveyor belt contact sensor (6.2.5). Polling is used to determine when this happens. Next, data is taken from the color sensor to determine how the box is oriented. The default orientation is where the afternoon side is closer to the user than the morning side. If the box is oriented the other way, the PIC sends a signal to the arduino, which then runs the servo controlling the 'pinball' mechanism (5.3.4.) is run to change which way the pills are funneled. After this, the PIC sends pulsing signals to the dispenser DC motors to turn the cylinders. Polling is used with analog voltage readings from the IR sensors to determine when a pill drops, and a motor is stalled once the correct number of its respective pill has been dispensed. After the first row is loaded, the conveyor motor is run until the contact sensor is activated again, and the process is repeated 7 times. After all rows have been loaded, the PIC instructs the arduino to run the servo that opens the base of the dispenser funnel, before sending signals to the dispenser motors to dispense the remaining pills. Voltage drops from the IR sensors are used to count the pills. A timer variable is used which resets every time a pill drops. If the timer reaches 5 seconds without a pill dropping, it is an indication that the cylinder is empty, and the motor is stalled. Finally, the position of the conveyor pusher block is reset by running the conveyor DC motor in reverse for 4.9 seconds (calculation in Appendix D-9). The machine returns to Standby mode and displays process information to the user. The total packaging time in the worst case scenario where 4 pills are to be dispensed into all 14 compartments of the pill box is 2 min 30 s, which is less than the 3 minute time limit (constraint s in Appendix B; calculation in Appendix D-9).

#### 6.3.3.1. DC Motor Control

DC motors are controlled by sending signals to 2 pins on the L293D chips (6.2.2). For example, the round pill dispenser motor is controlled with pins RA5 and RA4. The following are signals sent to these pins to control the motor:

| Signal (RA5, RA4) | Effect |
|---|---|
| (1, 1) | Not used - burns chip |
| (1, 0) | Clockwise rotation |
| (0, 1) | Counterclockwise rotation |
| (0, 0) | Stall |

To make the dispenser motors turn at a slower pace, PWM was used. This is a technique of sending a square wave with various ratios between on-time and pulse period (duty cycle) in order to simulate an analog voltage [13]. For example, the round dispenser motor is powered with 12V, but using PWM on RA5 with a 50% duty cycle (on-time = ½ pulse period) makes it as if they are powered with 6V. Through testing it was found that a 12% duty cycle gave an appropriate rotation rate of approximately 30 rpm, and that any lower made it so that the motors no longer generated enough torque to turn the cylinders.

#### 6.3.3.2. Interfacing with TCS34725 Color Sensor

The TCS34725 provides digital values for sensed R, G, B, and Clear intensities and sends these to the Arduino. Two important settings were calibrated: integration time and gain. Integration time is the amount of time the sensor is given to read light intensities before sending a signal. A longer integration time means greater sensitivity and higher resolution of the returned value (see Appendix E for datasheet extracts). Gain can take values from 1 to 60, and is the amount by which the values are amplified. Through testing it was found that an integration time of 154 ms and 4x gain was most appropriate - the sensitivity was not too high such that values can change unpredictably while the returned values had high resolution which made it easy to determine thresholds. The Arduino communicates to the PIC through a wire between the D12 (Arduino) and RB6 (PIC) pins. Everytime the blue intensity exceeds a certain threshold, the Arduino writes '1' to D12, which tells the PIC that the box is placed in the non-default orientation. This threshold depends on external lighting conditions, and must be calibrated every time.

#### 6.3.3.3. Servo control

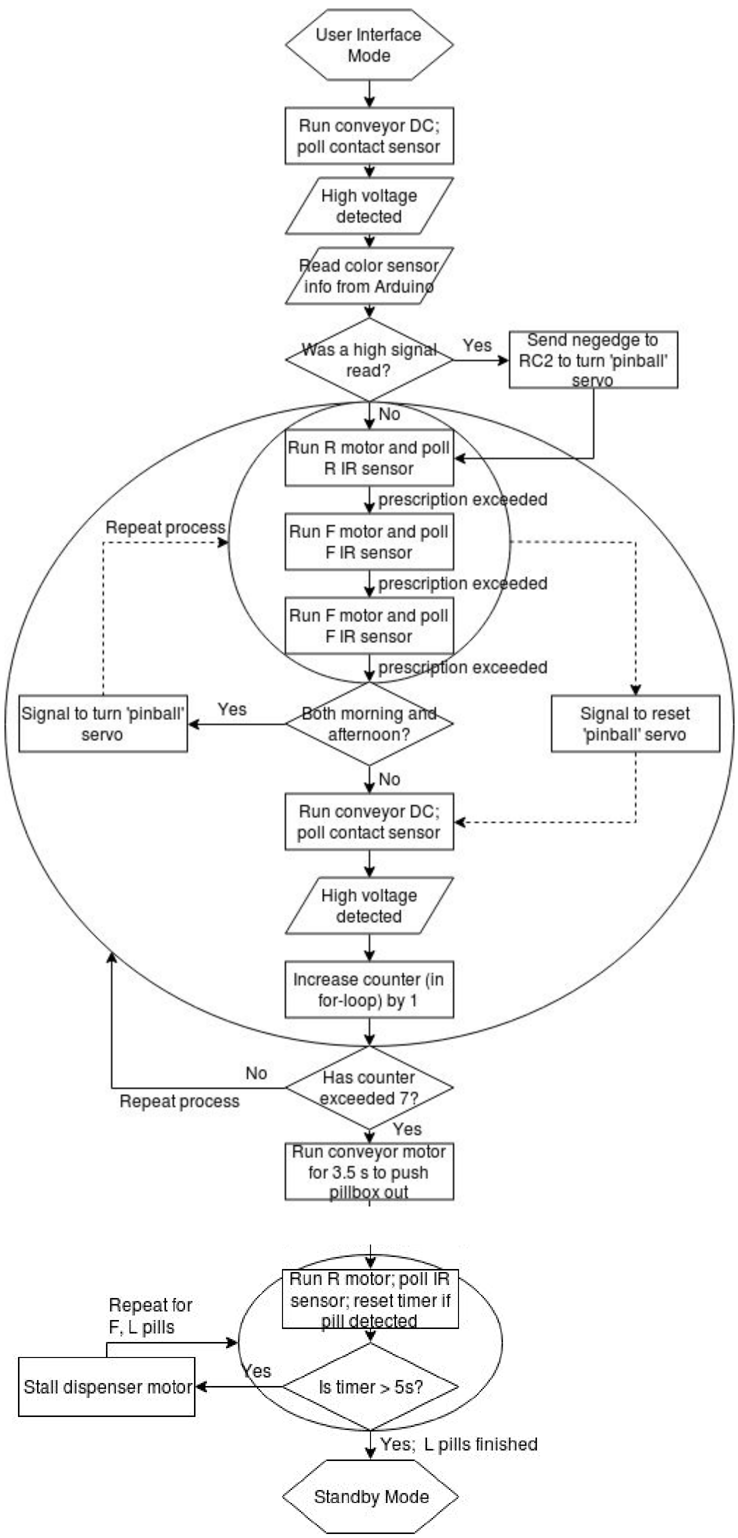
The FS90 servos are controlled using Arduino library functions - particularly servo.write(θ), which sets the angle of rotation to θ. The maximum and minimum allowed angles were determined after the servos were attached. To limit their rotation speeds, angles were changed 1 degree at a time, and a 10 ms delay was incorporated between consecutive changes (this delay was tested to be optimal). To allow the PIC to control the servos, two wires were used to connect the D3 and D5 pins (Arduino) to the RC0 and RC2 pins, respectively. The Arduino polls the D3 and D5 pins to determine when to run the 'base' (5.3.5) and 'pinball' (5.3.4) servos, respectively. The servos are turned counterclockwise on positive edges and clockwise on negative edges.

#### 6.3.3.4. Interfacing with IR sensors

The IR sensors, as mentioned earlier, sends analog voltages to the PIC. The Analog-to-Digital Conversion Module (ADC) was used to convert these voltage readings to 11-bit hexadecimal values. The ADCON1 register on the pic was set to 0b0111, which turned the RE2-0 pins into analog pins (refer to Appendix E for datasheet extract). As with the color sensor, for each IR sensor a threshold must be determined such that if the reading is below this threshold, one can be certain that an object is between the emitter and receiver. The thresholds for the R, F, and L cylinders were set to be 0x3F0, 0x092, and 0x2B9, respectively, but these must be changed if the cylinders were tampered with. However, a major disadvantage is that since none of the RB0, 1, and 2 pins, which are used for external interrupt, can be used to read analog signals, the cylinders can only be spun one at a time. This is because if they were run simultaneously and two pills fell at the same time, the polling technique of thresholding the converted analog signals may not register both events - interrupts are needed to do so. Thus, as mentioned in 4.6.3, extra holes were cut into the rotating disks in order to speed up dispensing so the process does not go over the time limit.

### 6.3.3.5. Functional Flowchart

The flowchart below is an in-detail representation of functional process of the Packaging mode.



### 6.3.4. Updated Pin Assignments

**PIC:**

| Pin | Assignment | Component | Pin | Assignment | Component |
|---|---|---|---|---|---|
| GND | Ground | Ground rail | GND | Ground | Ground rail |
| KPD | Digital Input | Enable/Disable Keypad | RC7 | Digital Output | Conveyor DC Motor FWD |
| RC6 | N/A | N/A | RC5 | Digital Output | Conveyor DC Motor BKWD |
| RC4 | N/A | N/A | RC3 | N/A | N/A |
| RC2 | Digital Output | "Pinball" servo control | RC1 | N/A | N/A |
| RC0 | Digital Output | "Base" servo control | VDD | N/A | N/A |
| RE2 | Analog Input | L. pill IR sensor | VCC | VCC | VCC Rail |
| RE1 | Analog Input | F. pill IR sensor | RE0 | Analog Input | R. pill IR sensor |
| RA7 | N/A | N/A | RA6 | N/A | N/A |
| RA5 | Digital Output | R. Disp. DC Motor FWD | RA4 | Digital Output | R. Disp. DC Motor BKWD |
| RA3 | Digital Output | F. Disp. DC Motor FWD | RA2 | Digital Output | F. Disp. DC Motor BKWD |
| RA1 | Digital Output | L. Disp. DC Motor FWD | RA0 | Digital Output | L. Disp. DC Motor BKWD |
| RD7 | Digital Output | LCD Data Bit | RD6 | Digital Output | LCD Data Bit |
| RD5 | Digital Output | LCD Data Bit | RD4 | Digital Output | LCD Data Bit |
| RD3 | Digital Output | LCD Enable | RD2 | Digital Output | LCD RS |
| RD1 | N/A | N/A | RD0 | N/A | N/A |
| RB7 | Digital Input | Conveyor contact sensor | RB6 | Digital Input | Color Sensor Arduino Input |
| RB5 | N/A | N/A | RB4 | N/A | N/A |
| RB3 | N/A | N/A | RB2 | N/A | N/A |
| RB1 | N/A | N/A | RB0 | N/A | N/A |

**Arduino Nano:**

| Pin | Assignment | Component | Pin | Assignment | Component |
|---|---|---|---|---|---|
| D1 | N/A | N/A | VIN | VIN | Input Voltage |
| D0 | N/A | N/A | GND | GND | TCS34725 GND |
| RESET | N/A | N/A | RESET | N/A | N/A |
| GND | GND | Ground Rail | +5V | Output | TCS34725 VIN |
| D2 | N/A | N/A | A7 | N/A | N/A |
| D3 | Input | ‘Base’ Servo Control (PIC) | A6 | N/A | N/A |
| D4 | N/A | N/A | A5 | SCL | TCS34725 SCL |
| D5 | Input | ‘Pinball’ Servo Control (PIC) | A4 | SDA | TCS34725 SDA |
| D6 | N/A | N/A | A3 | N/A | N/A |
| D7 | Output | ‘Pinball’ Servo Signal | A2 | N/A | N/A |
| D8 | N/A | N/A | A1 | N/A | N/A |
| D9 | Output | H-bridges Enable | A0 | N/A | N/A |
| D10 | N/A | N/A | AREF | N/A | N/A |
| D11 | N/A | N/A | 3V3 | N/A | N/A |
| D12 | Output | Color Sensor Info (to PIC) | D13 | Output | ‘Base’ Servo Signal |

## 6.4. Suggestions for Improvement of Subsystems

### 6.4.1. Electromechanical

It is proposed that the cylinders be mounted onto a flat surface, with cutouts for motor attachments and pill dispensing in order to improve structural stability. Currently, the cylinders have no bottom support, connected by one or two screws into a supporting block on the frame. A flat surface would prevent movements and vibrations of the cylinders, making them more stable; stability is critical for the IR sensors (emitter and receiver pair) attached below each cylinder for counting, as moving them even slightly would affect the calibration results. This redesign also prevents the cylinders from ripping off the screws caused by too much weight.

Currently, several parts of the machine are attached to the frame exclusively using epoxy glue, including 5 out of 6 of the motors. For making easier repairs (modularity) and structural stability, they should be secured by screws and brackets instead. Also, for some physical interfaces where the shapes were awkward and the contact surface is theoretically zero (such as the interface between the inverted plate for lowering pillbox lids and the back side of the pill distribution channel, where the contact surface is only a straight line and they were only held together with epoxy glue), the shapes could be modified, or additional supporting blocks could be added to increase contact surface area and improve the structural integrity.

Also, it would be better if the conveyor belt structure was temporarily attached to the frame using a different method, such as metal clips. Since the two structures are currently held together with only the conveyor belt itself (which is essentially a piece of cloth), when the machine is carried around, the two structures can fall possibly fall apart if the conveyor belt breaks.

### 6.4.2. Circuits

All circuitry components on this device uses solid core wire due to the ease in prototyping and PCB soldering; however, this type of wire is prone to excessive bending, often shearing off due to excessive wear. It is recommended that future designs incorporate braided wire for larger flexibility and larger contact area after soldering.

The mechanical design on the contact sensors need improvement. The current design is prone to deformation, especially with the back-and-forth motion of the conveyor. There are several alternatives for future design; the contact sensors can be built with metals that can handle the deformation elastically. They can also potentially be replaced with break-beam sensors. The motor can be replaced with a stepper motor, eliminating the need for contact sensors altogether at the expense of increasing cost.

The IR sensors need recalibration after each run, with their threshold values adjusted. This is likely due to a combination of changing light conditions and fluctuations in voltage after the machine begins operation; it is recommended that future IR sensors be built in an enclosure in order to isolate for environmental light factors. In addition, the output could be driven with an op amp as opposed to simply being connected to an analog pin on the PIC, in order to offer manual thresholding and ensure robustness against fluctuating voltage. The detection on the PIC can also be replaced with a negative edge detection instead of a threshold, allowing for a wider range of voltages for the IR sensor.

Several ICs on the device are currently soldered directly onto the PCB instead of being placed into an IC holder; this makes debugging and repairs more difficult, as a burnt IC means that the entire board needs to be resoldered. In the future, all IC chips will be placed into appropriate holders in order to ensure ease of debugging.

The current circuit utilizes a dedicated converter in order to step down the voltage from 12V to 5V; this, however, is costly, with the converter costing upwards of $10.00. In the future, it is recommended that this be replaced with one or two 9802 5V voltage regulators, with heat sinks attached in order to prevent overheating.

In terms of wire organization, it is recommended that wire layouts be planned ahead of time in order to maintain organization. In addition, it is recommended that a bus be used in order to connect the various inputs to the PIC board; this further reduces clutter and increases ease of debugging.

#### 6.4.3. Microcontroller

The microcontroller subsystem can be improved in 3 areas. Firstly, as mentioned in section 6.3.3.4, the cylinders could not be run simultaneously as interrupts were not found to be available for registering the converted analog signals. Improving this would either involve finding a way to enable interrupts on the analog pins or read analog signals on the RB2-0 pins, or pursue another solution that allows the cylinders to be run simultaneously without interrupts. One suggestion is to have a short delay of ~200ms between the start of one cylinder and another instead of starting all of them simultaneously, which would drastically reduce the chance of two pills falling at the same time. Secondly, EEPROM on the PIC could be used to store logs of previous runs, which would assist in evaluating the performance of the machine. Finally, at its current state, debugging various parts of the algorithm is a tedious process. For instance, to test the threshold of the IR sensor for the F cylinder, one would have to start the machine and enter a full prescription first. Thus an improvement would be to create testing algorithms for various sensors and mechanisms so that they can be calibrated independently of all other components.

# 7. Solution: Integration of Subsystems

The process of integration revealed a large amount of conflicts, although steps were taken to mitigate the effects. The development of the machine has the potential to be a fairly pipelined process; many components cannot be tested effectively without the prerequisite components developed and in place. Consequently, a large amount of testing can be done only after the prerequisite components have been installed.

## 7.1. Preparation Process

The construction of the machine began when the two main structures (main frame and conveyor belt structure) were constructed first. All the individual components or subsystems were then attached to the frame or the conveyor belt structure, generally from the bottom up. The size and orientation of each subsequent part was dependant on the adjacent component just below it, and if needed, changes were made before the part was made and installed. For example, the conveyor belt was constructed first, followed by the pill distribution channels, and then the dispenser cylinders. This was an attempt to prevent misplacements or parts not matching up with another. All the electromechanical parts were attached for once, in some cases using tape or hot glue instead of permanent adhesives so they can be removed or adjusted later during integration, before integration to make sure that all parts fit together.

For circuits, the key parts (such as H-bridges) were first built separately outside on the breadboard and tested, before being transferred and soldered onto a protoboard, leaving out the interface wires with the electromechanical or microcontroller subsystems. The partial circuit was then tested again while temporarily connected to an actuator and a microcontroller board.

For the microcontroller and programming, most of the functionalities (except for user interface) were pre-programmed, or at least had a skeleton code completed before integration.

## 7.2. Integration Process

When integrating, in general, a component was first taken down from the frame, then attached to loose wires connected to a breadboard (or the partially built circuit on the breadboard), which was temporarily connected to the microcontroller. The component was then tested outside of the machine frame first, which gave the opportunity to adjust the mechanical parts, circuits, and code. The component was then reattached back onto the frame, and the wires are soldered on. On the larger scale, integration were completed separately first on the frame and the conveyor belt, and later on, the conveyor belt structure was integrated with the frame, which only involved connecting wires to the contact sensors.

Integration was first done on the conveyor belt structure, where the side walls were scoured out to a depth of 4mm, making space for the triangular shaped contact sensors. The pusher block with wires were then attached, followed by wires that were temporarily attached to the contact sensors. These were used to test the reliability and connectivity of the system. Once that was completed, other physical parts such as the lid closing mechanism and the wedges for lid opening were attached to the structure. This order ensured that no parts were obstructing the construction and testing of the contact sensors.

On the main frame, the channel switching box ("pinball mechanism") was attached first, which was lined up with the location of the conveyor belt and the wedges. This was followed by the driver roller, where a DC motor was attached to a section of closed PVC pipe using a metal shaft attachment. The motor and roller system was lined up with the height of the conveyor belt top surface, and custom built wooden blocks were used to hold the system in place. An aluminum sheet bracket was made to secure the DC motor onto the wooden blocks.

The cylindrical dispensers were then attached. Wires were first soldered on before they were screwed onto the frame. The cylinders were first oriented such that the holes where pills fall out of the cylinders line up with the rotating base of the channel switching mechanism. This would ensure that extra pills can fall down into the collection ramps below, which were installed onto the frame right afterwards to line up with the holes above. The cylinders, "pinball mechanism", and ramps were then tested to make sure that the pills drop down properly.

The two servos controlling the channel switching barrier and the rotating base were then calibrated to the exact angles by connecting to the Arduino Nano. The angles were tweaked on a temporary program connected to the Arduino Nano, before being used in the main program. Later on, due to design changes in the cylindrical dispensers (explained above in Section 4.6.3), they had to removed, renovated, and rearranged. When two of the cylinders (for long and flat pills) did not work well, they were completely rebuilt. In addition, since the collection channels for catching remaining pills were poorly designed, they were taken down, and either remodelled, or rebuilt (explained above in Section 4.6.4).

Also, earlier in the process, the power supply issued in the design kit was used in substitution of the final power supply as it was not ordered back then. It was attached to the frame using hot glue. When the first ordered power supply arrived, it was found that it supplied too much current and burned some of the components, so a second, smaller power supply was bought, and installed. Another voltage converter had to be bought and installed as well since the machine needed a 5V source. These components were attached to the inside of the left wall (viewed from front), and was connected to the main power switch located on the outside of the left wall. The appropriate power cables were also attached to the power supply and the PIC board.

The PIC board was first left out of the frame, such that it was easier to be moved around to test. Later, it was moved onto the slanted surface on front of the frame, but not attached permanently so it could still be slided around. It was only permanently attached once the rest of the machine was complete. This was the same for the Arduino Nano.

In addition, when wires were first attached to the components, they were colour coded by their purpose (shown in chart below). The key wires were also labelled using tape to avoid confusion, and to identify which pin it would go to on the PIC board. Once the wiring is complete, they were organized by gluing or taping onto the frame, such that they would not clutter the space inside the machine. Wires that travel to the same destination were also bundled (either by twisting them together, or bundled using electrical tape).

Wire colour code:

| **Colour** | Yellow | White | Red/Pink | Black/ Dark Blue |
|---|---|---|---|---|
| **Function** | Signals to actuators | Signals between microcontrollers | Power | Ground |

## 7.3. Testing and Debugging

Testing was a constant process during integration and it occurs in parallel while the parts were being integrated. As mentioned above, the component was tested outside of the machine frame first, before it was installed onto the frame. Individual components and subsystems were then tested separately using temporary programs. Once these parts all worked fine, the code for a full operation was loaded into the PIC and Arduino Nano, and the entire operation was checked.

Below summarizes the results of some functionality tests that were performed on subsystems and the full operation. For full details of the testing procedure and results, please see Appendix D.

- **Cylinders: consistency of pill dispensing (with original contact sensors)**
  -Spherical: 97% (occasionally skips pills)
  -Long: 53% (often stand up vertically and jams when it hits the barrier)
  -Flat: 70% (sometimes stand up vertically and jams when it hits the barrier)

- **Cylinders: percentage of rotations registered by the contact sensors**
  -Spherical: 94.7%
  -Long: 70.7%
  -Flat: 64%
  It was found that if the positioning of the contact sensors was not adjusted, deformation of the wires caused them to lose consistency over time.

- **Testing consistency of pill drop detection with IR sensors**
  -Spherical: 100%
  -Long: 98%
  -Flat: 98%
  These results show that if the IR sensors are properly calibrated and not moved around, then they can perform very consistently.

- **Pill distribution: percentage of pills that end up successfully in the collection reservoir**
  Original channels:
  -Spherical: 71.9%
  -Long: 21.5%
  -Flat: 35.6%
  New channels:
  -Spherical: 92.2%
  -Long: 100.0%
  -Flat: 96.7%

- **Conveyor contact sensor reliability testing**

| Test # (design 1) | Number of stops | Test # (design 2) | Number of stops |
|---|---|---|---|
| 1 | 7 | 1 | 5 |
| 2 | 4 | 2 | 4 |
| 3 | 5 | 3 | 5 |
| 4 | 0 | 4 | 3 |
| 5 | 1 | 5 | 0 |

As with the dispenser contact sensors, it was found that if the conveyor contact sensor was not re-adjusted every time, it slowly loses reliability due to deformation.

From performing these tests, many problems were identified. These problems and how they were addressed are discussed in Section 4.6. First, the contact sensors seemed fairly unreliable, as without adjusting the contact surface after each operation, the success rates are pretty low. This means that the contact sensor designs are poor, and that they should either be replaced or modified. As a result, the contact sensors on the cylindrical dispensers were completely scrapped. The design for the conveyor belt contact sensors were significantly modified, and became much more reliable (although still not good enough to be totally reliable).

Second, the test on the cylinders revealed that barriers cannot be used at all for long pills and flat pills, due to the risk of jamming when the pill stands up vertically. Also, it is very difficult to make sure that exactly one pill drops on each rotation. Therefore, the barriers were removed, and the disks (in some cases the entire cylinder) were rebuilt and reoriented differently.

## 7.4. Current Functionality Status

As of the showcase on Monday, April 9th, 2018, at its best, the machine was able to perform the following functions:

- Switching on and off using the main power switch
- Interfacing with the user on the PIC board and starting the operation
- Dispensing the correct number of spherical and long pills
- Switching channels between two sides of the channel switching mechanism
- Stopping the conveyor belt and pillbox at the right location
- Loading the correct number of spherical pills into the pillbox compartments
- Counting the number of spherical and long pills
- Able to handle the maximum number of pills (45 each)

- Opening and closing the rotating base plate of the channel switching mechanism
- Collecting most of the extra pills into the removable collection reservoir

However, these functions either failed or have not been implemented:

- Detecting the orientation of the pillbox with the colour sensor
- Counting the number of flat pills
- Displaying current time on LCD display
- Resetting the pusher block back to start position after an operation or shutting down
- Loading the correct number of long and flat pills into the pillbox compartments
- Repeatability of machine operation (ability to handle multiple pillboxes back to back)
- Stopping the pusher block at the last stop as operation ends

During the demonstration, most of the core functionalities failed due to a series of smaller issues preventing the operation from progressing, especially the IR sensors on the dispenser cylinders. These issues were partially resolved during the machine evaluation, but were still impeding the overall system functionality.

## 7.5. System Improvement Suggestions

In addition to making all the functionalities work, the following suggestions for improvement can be made to the overall system:

First, more precise manufacturing of aluminum parts, especially the cylindrical dispensers and the ramps guiding the pills down onto the collection reservoir.  Currently, some parts were quickly put together and calibrated to the best position using trial and error, which offered a good short-term solution. However, since there are large spaces between the two ramp sections, there is always a room for uncertainty, and in the long-term, such a design is not acceptable. To make sure that this system would work totally reliably, the dispenser cylinders and their adjacent parts need to be a lot more stable and robust. The ramps need to move a lot less, and connect more directly than having a large space in between sections.

A second improvement would be to better organize the wiring. Currently, the colour code only has 4 colours, which is insufficient to for an outsider to distinguish between the functions of each wire, even if the wires are labelled with tape. Also, a lot of wires had loose connections to other parts (especially to the PIC), which resulted in extensive time wasted debugging when a simple disconnection error occurred. Permanently fixing all the wires would reduce the opportunity of sudden “breakdowns” caused by disconnecting wires.

Another improvement would be to use IR sensors on the conveyor belt instead of contact sensors. Currently, the contact sensors, especially contact surface on the pusher block, requires manual adjustment before every operation, or it bend too much and fail. To stop the need for manual adjustment, IR sensors could be used instead, where in a similar configuration to the cylindrical dispensers, would have strips of material marking the stopping locations of the conveyor belt, which would pass through an IR sensor pair. In general, IR sensors are far more reliable than physical contact sensors that completes a circuit, if they are properly calibrated.

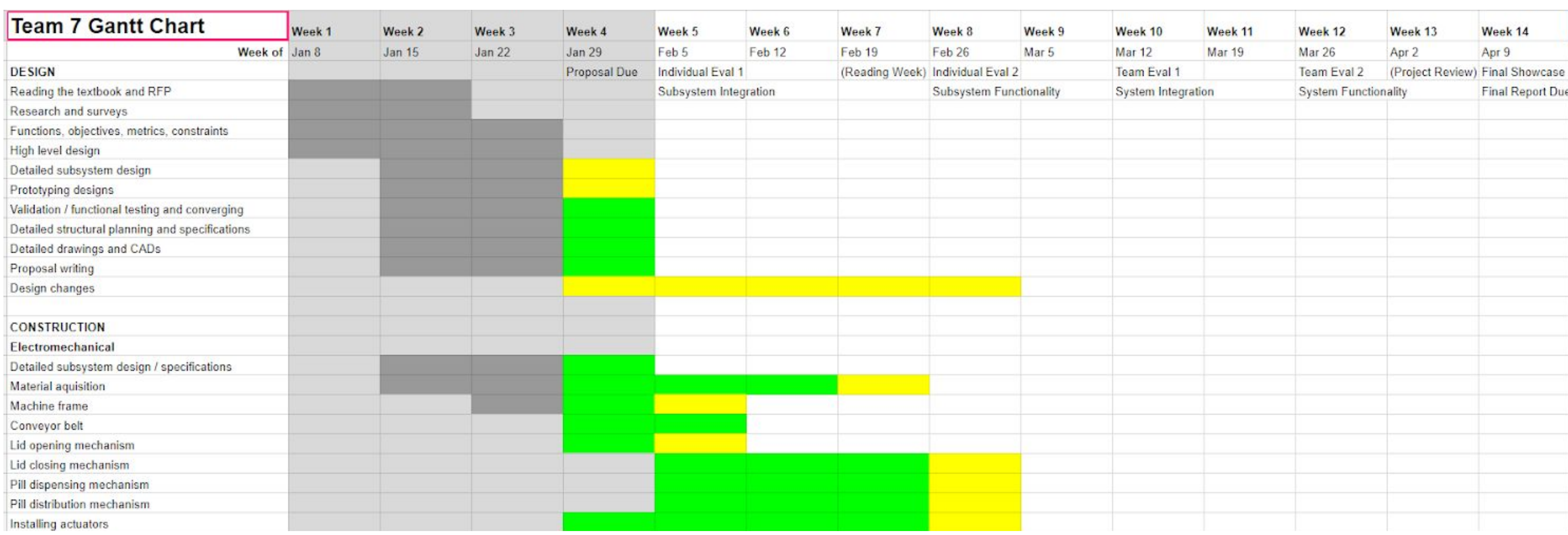
Furthermore, a necessary improvement would be to make turning off the machine as an interrupt, so it would not literally shut down the machine. Also, this would make it possible to perform tasks such as resetting the conveyor belt and the dispenser cylinders after each time the machine is shut down during an operation.

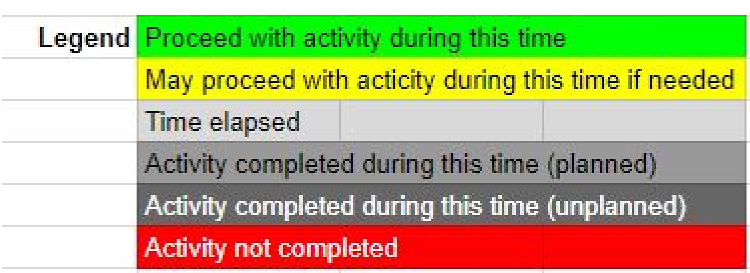
Finally, the design and fabrication of this machine needs to be more organized in general. Instead of changing designs of subsystems several times throughout the term, more planning should have been done at the beginning, so the designs are more robust from the start. This way, more time can be spent working on one design instead of having to work on multiple designs. This would have been a lot more efficient. The details in the planning are discussed below.

# 8. Administration

## 8.1. Initial and Accomplished Schedule

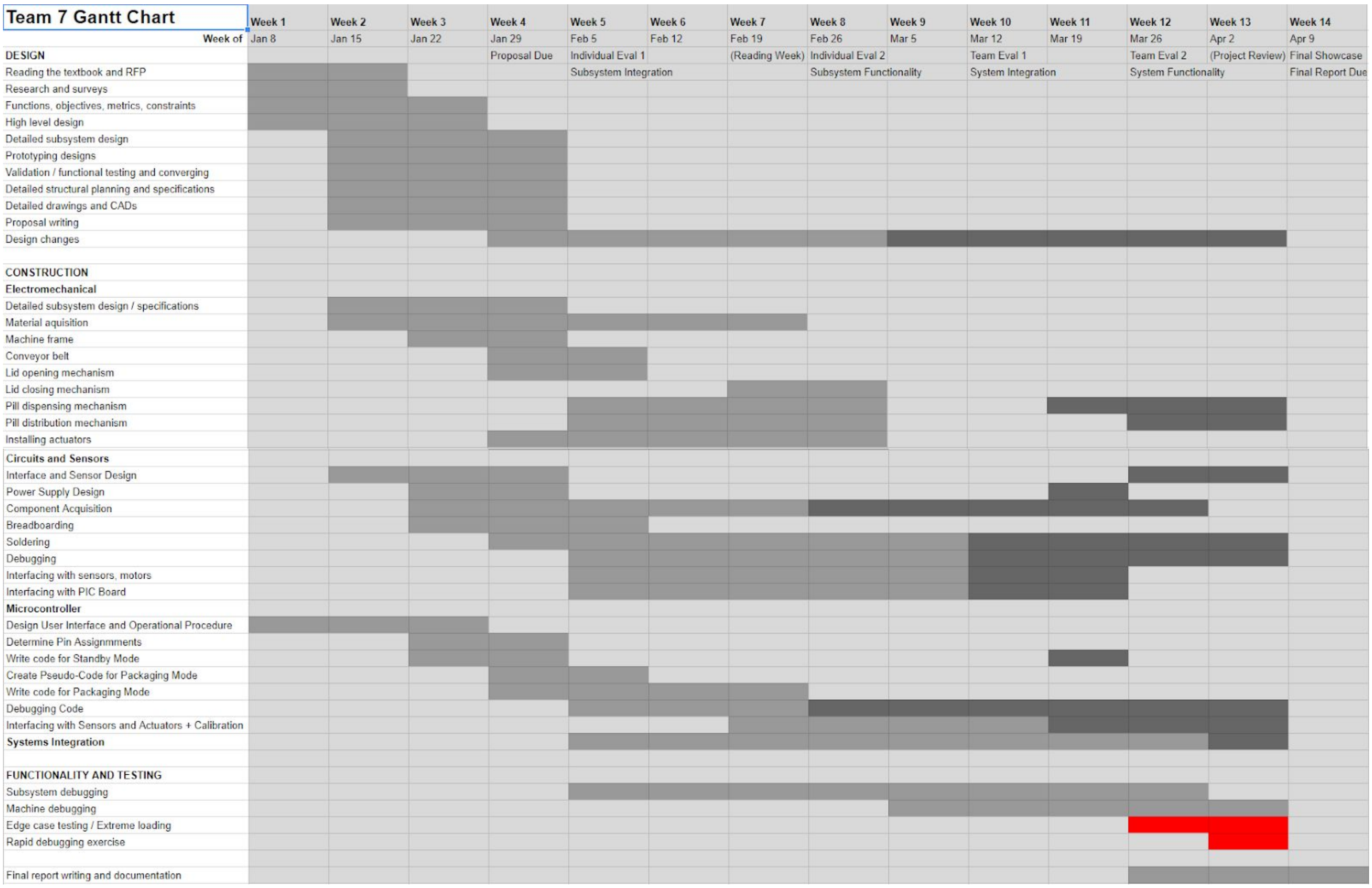
**Initial:**



It was planned that individual subsystem construction should be complete by the end of reading week, and that integration should begin right after. Integration and debugging should be a parallel process, where integrated parts are immediately tested, and then iterated if needed. It also allowed for some flexibility in changes, as highlighted by the yellow colours.

**Final:**




It was noticed that design changes took place a lot longer than originally planned, mainly due to some last minute changes that need to improve the functionality and reliability. This translated to a lot of initially unplanned additional work in all three main areas. It also gave us much less time to perform functional testing. In all, the team significantly underestimated the times taken to complete certain tasks, which led to more pressure, disorganization, and issues towards the end of the term. The takeaways and lessons learned are discussed below.

## 8.2. Final Spendings

The cost of the machine prototype is $210.34, which is well under the constraint of $230.00 (Cost breakdown shown in Table C below). Tallying all the available receipts of team spendings, however, the total money spent on this project as a team was $1076.48 (including the project kit price of $300). However, with many lost records of transactions, this number could possibly be a lot higher. This is over our initially planned maximum of $1000 total. Some of the reasons for this mismanagement and lessons learned are discussed below.

## 8.3. Learning Experiences and Takeaways

The main takeaway from project planning is that we could have planned a lot better and did much more research (or simply comparing designs with other teams) at the start to avoid making so many design changes later in the process. The design team stuck with some relatively challenging designs at the start, which took a lot longer to build and perfect. Some of these designs were scrapped and replaced with simplified designs after comparing with other design teams or reference designs online.

Furthermore, since the design team frequently changed designs, it was difficult to perfect all the designs due to limited time in this course. Therefore, it would have been more effective to just try to perfect a single design for each subsystem over the course of this project, and doing enough research to find an efficient design at the start.

The design team also mishandled some of the problems leading up to and during the demonstration, where a majority of the machine failed as small problems escalated. This was likely caused by team members panicking in the hours leading up to the demonstration. As a result, instead of putting in productive work, the "panic fixing" only caused more problems to arise, eventually disabling most of the machine. The correct approach would have been to calm down and think about the right strategy in fixing these issues.

In terms of budget management, even though the machine itself is well under the limit, the team could have been better in managing expenses. A lot of parts were either lost, broken repeatedly, or unused. This is probably the result of poor organization and planning, as well as carelessness. In addition, the team frequently seeked convenient but expensive sources to purchase components, especially Home Hardware and Creatron Inc. A lot of money could have been saved if orders were made online via cheaper sources like Amazon or Digikey. This would have required more planning as well (such as ordering the right parts instead of going to a store and buy many parts, many end up being useless).

# 9. Conclusion

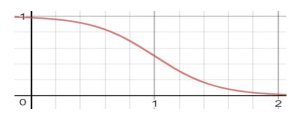
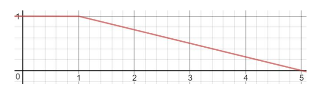
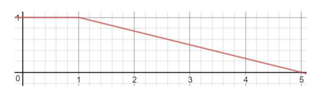
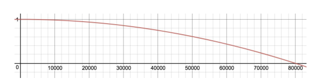
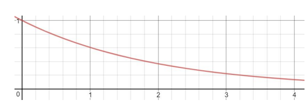
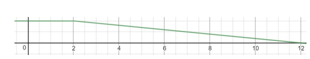
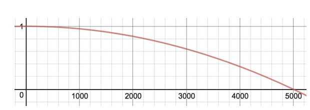
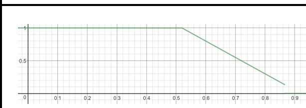
In conclusion, this project was moderately successful. By the time of demonstration the working functionalities included running the conveyor belt, making the box stop at appropriate locations, dispensing the correct number of pills from each dispenser, and controlling servos to direct pills. In terms of objectives, the machine meets the criterion of safety as there exists no identifiable hazards for physical injury nor risk of electrical shock. It is affordable as the total cost was well under the limit of $230 CAD. It is user-friendly as the processes of loading and retrieving the pill boxes and pills were simple and the user interface was intuitive. Finally, it is an elegant design with high manufacturability as all components have been designed to make most use of mechanical parts and have minimal degrees of motion. However, due to the previously mentioned system and subsystem flaws, the results were not 100% reliable, and there exists room for improvement. Furthermore, through this project the team gained insight into the interconnectedness of various subsystems of a large system, and thus the need for careful planning and thorough debugging to ensure compatibilities. Nonetheless, the project successfully demonstrated that it is possible to build a small scale, autonomous, and affordable machine suitable for use in a nursing home or hospital setting.

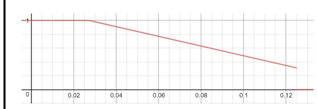
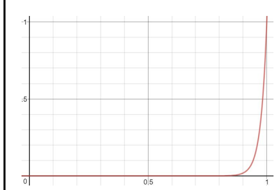
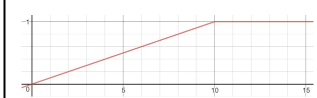
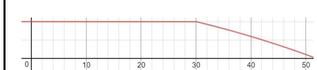
# 10. References

[1] University of Toronto, "RFP #1: The Pill Boxing Machine", Toronto, 2018.
[2] "Tablet Counter | Tablet Counting Equipment", Busch-machinery.com, 2018. [Online]. Available: https://busch-machinery.com/Tablet-Capsule-Counters.htm. [Accessed: 12- Jan- 2018].
[3] Insights: Advancing the Science of Pharmacy. 2018, p. 2.
[4] N. Collins, "Four in ten drugs wrongly administered in hospitals", Telegraph.co.uk, 2018. [Online]. Available: http://www.telegraph.co.uk/news/health/news/8951940/Four-in-ten-drugs-wrongly-administered-in-hospitals.html. [Accessed: 31- Jan- 2018].
[5] S. Ubelacker, "Medication errors caused death or severe harm to 36 Ontario patients: report", CTVNews, 2018. [Online]. Available: https://www.ctvnews.ca/health/health-headlines/medication-errors-caused-death-or-severe-harm-to-36-ontario-patients-report-1.1419539. [Accessed: 31- Jan- 2018].
[6] "Pillo Health | Your Personal Home Health Family Robot", Pillo, 2018. [Online]. Available: http://pillohealth.com/. [Accessed: 31- Jan- 2018].
[7] N. Varvarelis and J. Samuelson, "Electronic Pill Dispenser", US7359765 B2, 2018.
[8] "Google Patents", Patents.google.com, 2008. [Online]. Available: https://patents.google.com/patent/US7359765B2/en. [Accessed: 12- Jan- 2018].
[9] How to Invent a Pill Dispenser. 2018.
https://www.youtube.com/watch?v=D1CmDMKBFCg
[10] J. Biggs, "This candy sorting machine isn’t the hero the world wants but it’s the hero the world needs", TechCrunch, 2017. [Online].
[11] Pill Loading and Dispensing. 2016.
https://www.youtube.com/watch?v=D1CmDMKBFCg
[12] Proposal document
[13] [9]"Arduino - PWM", Arduino.cc, 2018. [Online]. Available: https://www.arduino.cc/en/Tutorial/PWM. [Accessed: 16- Feb- 2018].

# Tables

## A. Objective Model

| Objective | Metric/Parameter | Scale | Unit | Constraints | Utility Function |
|---|---|---|---|---|---|
| **Safety** | | | | | |
| No potentially harmful parts should be exposed (e.g. exposed wires, sharp metals, blades, rotating or fast-moving parts) | Number of exposed parts that may be potentially harmful | Value | 1 | All harmful components must be concealed from easy access | N/A |
| Emergency stop button stops machine immediately | Time taken for operation to stop after emergency button is pressed | Value | Seconds | Machine must stop all motion immediately (<2 seconds) [1] |  |
| **Usability / User-Friendliness** | | | | | |
| User-interface is easy to use | The number of instructions required for the user to operate the interface | Value | 1 | N/A |  |
| Minimize effort required for loading and retrieval of pill box/pills | Number of instructions that the user requires to load/retrieve the box/pills | Value | 1 | N/A |  |
| **Manufacturability and Complexity** | | | | | |
| Minimize the total number of components required | -The number of components in a mechanism / subsystem / the machine divided by its volume<br>-Each electronic part (motor, actuator) is weighed as 5 components | Ratio | $1/m^3$ | N/A |  |
| Minimize motions involved in the packaging process | The number of degrees of motion involved for any one component | Value | 1 | N/A |  |
| Geometry of the design and the parts are simple | Total number of surfaces of all components divided by the number of components | Ratio | 1 | N/A |  |
| **Efficiency** | | | | | |
| Minimize the amount of time required to set up the machine (which includes inserting pill box, loading pills, and entering instructions | Total amount of time required to perform the listed tasks | Value | Seconds | Setup time should be less than 1 minute [1] |  |
| Minimize the amount of time required to complete the loading process | Total time required to organize the pills into the pillbox and return the remaining pills | Value | Seconds | Total time for one operation must not exceed 3 minutes [1] |  |

| | | | | | |
|---|---|---|---|---|---|
| Compactness and Portability | | | | | |
| Minimize the size of the machine | The length of the main diagonal of the bounding box of the subsystem /machine | Value | Meters | Total dimensions of the machine must not exceed 0.50 x 0.50 x 0.50 m [1] |  |
| Minimize the weight of the machine | Total weight of the machine | Value | kg | Total weight of the machine must not exceed 6 kg [1] | |
| Affordability | | | | | |
| Minimize the total cost of all components comprising the machine. | Total cost of all components (manufacturing costs are not included) | Value | $ CAD | Total cost of the machine must not exceed $230 CAD [1] | |
| Reliability | | | | | |
| Minimize the number of errors made | Total number of successful operations over the total number of operations | Ratio | 1 | Must not make any errors |  |
| Maximize robustness of all subsystems by maximizing tolerance for variability in the dimensions of the box and pills | Maximum possible deviation of the object dealt with by the subsystem from the desired position such that the subsystem will still function as intended | Value | mm | N/A |  |
| Elegance | | | | | |
| Minimize noise pollution from the machine | Maximum level of noise generated during machine operation | Value | Decibels | Maximum noise level must not exceed 50 dB [REF] |  |

## B. General Machine Specifications

| Category | | Specification |
|---|---|---|
| Dimensions | Length | 49.5 cm |
| | Width | 42.0 cm |
| | Height | 45.5 cm |
| Weight | | 5.14 kg |
| Total Cost (Broken down in detail below): | | $210.43 |
| Power Supply | | 12V 5A 60W; 5V 5A 25W |

| | | |
|---|---|---|
| Microcontrollers | | -PIC18F4620 DevBugger Board<br>-Arduino Nano (1) |
| Actuators | | -120:1 Geared DC Motor (1)<br>-TGP01S-A130 DC Motor (2)<br>-ZGA25RP DC Motor (2)<br>-FEETECH FS90 Servo Motor (2) |
| Sensors | | -TCS34725 RGB Colour Sensor (1)<br>-LTR-301-LTE-302 IR Pair (1)<br>-LTE-4208-LTR-3208E Pair (2) |
| Run Time | Loading + Inputting Prescription Instructions | 25 s |
| | Operation: Preparation | 6 s |
| | Operation: Dispensing | Up to 90 s depending on prescription |
| | Operation: Returning | Up to 65 s depending on the amount of leftover pills |
| | Total Operation Time | Up to 3 minutes and 6 seconds* |

*Time tested using maximum number of pills possible and the largest prescription possible, and assuming worst case scenario in picking up pills from cylinders.

## C. Cost Breakdown / Bill of Materials

| Item | Quantity | Unit Cost | Total Cost | Supplier |
|---|---|---|---|---|
| **Microcontrollers** | | | | |
| PIC18F4620 DevBugger board | 1 | $50.00 each | $50.00 | Project Kit |
| Character LCD + Keypad | 1 | $6.00 together | $6.00 | Project Kit |
| Arduino Nano | 1 | $5.00 each | $5.00 | Amazon |
| **Power Supply & Power Cables** | | | | |
| Yaetek AC110V/220V to DC12V 5A 60W Switch Power Supply | 1 | $10.00 each | $10.00 | Amazon |
| SuperNight DC 12V 24V to 5V 5A Converter Step Down | 1 | $13.00 each | $13.00 | Amazon |

| | | | | |
|---|---|---|---|---|
| Regulator 25W Regulated Power Supplies Transformer | | | | |
| INSTEN for Computers / Printers / Monitors Black 6-feet 3 Prong Power Cable (Main power cable) | ½ of total length | $12.00 | $6.00 | Home Hardware |
| Jack 21-25 (PIC power cable) | 1 | $1.95 | $1.95 | Digikey |
| **Actuators** | | | | |
| 120:1 Geared DC Motor | 1 | $16.00 each | $16.00 | Creatron |
| TGP01S-A130 DC Motor | 3 | $5.00 each | $15.00 | Project Kit |
| FEETECH FS90 Servo Motor | 2 | $5.25 each | $10.50 | Pololu |
| 4 mm D-shaft gear motor attachment | 1 | $5.30 each | $5.30 | Creatron |
| **Structural Materials** | | | | |
| 0.025” Aluminum sheet | 0.38 m^2 | $26.29 / m^2 | $10.00 | Rona |
| 1” V-shaped aluminum channel | 0.22 m | $1.41 / m | $0.31 | Home Hardware |
| ¼” Plywood | 0.42 m^2 | $10.76 / m^2 | $4.52 | Home Depot |
| ½” Plywood | 0.02 m^2 | $18.74 / m^2 | $0.38 | Home Depot |
| 1/2 “ x 1” Wood | 0.40 m | $0.23 / m | $0.09 | Home Depot |
| 1” x 2” Wood | 2.30 m | $0.52 / m | $1.20 | Home Depot |
| 1” x 4” Wood | 0.90 m | $0.91 / m | $0.82 | Home Depot |
| 2” x 4” Wood | 0.50 m | $1.44 / m | $0.72 | Home Depot |
| ½” Wooden dowel | 0.25 m | $2.07 / m | $0.52 | Home Depot |
| Cotton fabric (conveyor belt) | 0.09 m^2 | $11.06 / m^2 | $1.00 | G & S Dye |
| 1 ¼” PVC piping | 0.35 m | $6.07 / m | $2.12 | Home Depot |
| ¼” Metal thread rod | 0.45 m | $8.22 / m | $3.70 | Home Depot |
| Traction tape | 0.029 m^2 | $48.40 / m^2 | $1.40 | Home Hardware |

| | | | | |
|---|---|---|---|---|
| **Electrical Components** | | | | |
| 1µF capacitor | 8 | $0.25 each | $2.00 | Home Hardware |
| Wires (Red, Yellow, Blue) | 50 ft (est.) | $0.20 / feet | $10.00 | Home Hardware |
| Resistors (1kΩ x 4, 1MΩ x 3, 470Ω x 3) | 10 | $0.03 each | $0.30 | Home Hardware |
| L293D (Dual H-bridge IC) | 2 | $4.50 each | $9.00 | Home Hardware |
| **Sensors** | | | | |
| TCS34725 RGB colour sensor | 1 | $9.00 each | $9.00 | Adafruit |
| LTR-301-LTE-302 IR Pair | 1 | $1.50 each pair | $1.50 | Digikey |
| LTE-4208-LTR-3208E Pair | 2 | $1.00 each pair | $2.00 | Creatron |
| **Miscellaneous** | | | | |
| Bolts and nuts | 7 | $0.40 each pair | $2.80 | Home Depot |
| Screws (any size) | 15 | $2.20 / 10 | $3.30 | Home Hardware |
| Glue, solder, electrical tape, etc. | --- | --- | $5.00 (Estimated) | (Various) |
| **Total Prototype Cost** | | | **$210.43** | |

# Appendices

## Appendix A. Design Values and High Level Objectives (From Proposal)

1. Safety:

**Definition:** Protection and prevention from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.  (MIL-STD-882E for systems safety)
**Justification:** In order to prevent employee exposure to electrical shock, safety must be the utmost priority for this machine. Incorrect circuitry connection during the battery sorting process can cause sparks or flames that would release hazardous chemicals, damage the machine, and possibly injure the operator.

2. Usability and User-friendliness:

**Definition:** Little time/effort is needed to set up and calibrate the machine, "the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human" [6]. In addition, “little time and effort is needed to set up and calibrate the machine, and the machine is modular so that parts can be replaced or repaired easily” [1].

**Justification:** In order to allow employees of a variety of skill levels to operate the machine, the machine must be quick and easy to set up, and the interface must be easy to understand and use by the users, and the loading procedure must be simple. In case of a malfunction or repair, the machine is designed in such a way that it is simple to make repairs and replace parts.

3. Manufacturability and Complexity:

**Definition:** Little time and effort is required to manufacture the machine to an excellent overall quality, and the machine is designed in a way such that it uses a minimum number of components, with simplistic geometry shapes, and has a simple operation process.

**Metrics:** The number of actuators/electronic components (motors, servos, solenoids, etc.) required; the total number of components required; the number of modes/dimensions of motion; complexity of the geometry of the parts

**Justification:** Since the design team has limited time to design and construct the machine, a simpler design / lower complexity would allow for less effort and time spent to make the machine of better quality, and thus, meet the requirements before the deadline.

4. Reliability and Efficiency:

**Definition:** The completion of the pill-boxing and sorting task using minimal time possible, and resulting in a minimum amount of errors. [1]

**Justification:** The machine must be able to complete the pill-boxing task within the required 3-minute time frame, as given by the Request for Proposal. In addition, since the user needs to organize a large number of pills into pillboxes for the entire nursing home, the process needs to be efficient in order to save time. Less mistakes in the pill organization process would also save time and effort for the user, since they would not need to reorganize the pills manually, and would be more desirable for the patients, who rely on the machine to organize their prescription correctly. [1]

5. Compactness and Portability:

**Definition:** The ability to easily lift and transport the machine. [1]

**Justification:** In order to allow the client and the engineering team to test the machine is a variety of setting around campus during the design process, the machine must be portable enough to be carried easily. The supply company will also likely have multiple sorting machines, and the machine not occupy excessive space. As specified by the Request for Proposal, the machine must also be less than 6 kg, and fit within a 0.50 x 0.50 x 0.50 m^3 size constraint. [1]

6. Affordability:

**Definition:** The price to make and assemble the machine.

**Justification:** The total cost of the machine must be under the cost constraint of $230 CAD as specified in the Request for Proposal. The cost must be low enough to make sense economically versus hiring human workers for the client. [1]

7. Reliability:

**Definition:** The machine can withstand wear over repeated usage and mechanical stress. It functions consistently in a wide range of operating environments with a low failure rate. [1]

**Justification:** The client requires the machine to work reliably over repeated use over time without a malfunction.

8. Elegance:

**Definition:** "Machine looks elegant, and operates quietly and smoothly with little or no sensible noise or vibration." [1]

**Justification:** Since a nursing home should be a quiet environment, the operation of the machine should not disturb the residents. In addition, the appearance of the machine should blend in with the nursing home environment and not be visually distracting/unappealing for its residents. [1]

## Appendix B. Constraints as per the RFP

From the Request for Proposal, the following constraints must be applied to the design of the pill boxing machine:

a. The entire prototype (including the reservoirs while containing pills and the box when placed in the machine) shall completely fit within a 0.50 x 0.50 x 0.50 m envelope at all operation times (power cable notwithstanding.)

b. The weight of the machine, including the empty reservoirs, power cable, etc., shall not exceed 6 kg.

c. The total prototype costs shall not exceed $230 CAD before shipment and taxes. For parts purchased in foreign funds, the exchange rate reported by the Central Bank of Canada at the end of business day on January 8th, 2018, will be considered. The manufacturing labour is not considered on top of the material costs in the prototype, unless a part is manufactured using a 3D-printer or CNC machine. In such cases, an additional cost of $5 CAD per manufacturing hour will be assumed. The G-code and
exact manufacturing time for such parts shall be reported.

d. Use of materials such as paper (of any type) or corrugated plastic for fabricating the machine, and non-standard fasteners such as duct tape, masking tape, hot glue, etc., is not acceptable. It is imperative to have the client's explicit consent for other cases similar to the above.

e. The machine can be plugged in the AC, 110V-60Hz, 3-pin outlet. Only one connection cable is allowed.

f. The machine must have an easily-accessible emergency STOP switch that stops all the mechanical moving parts immediately.

g. The machine must be fully autonomous, and no interaction with an external PC or remote control is permitted during the operation. The operation must begin by pressing a <start> button on a keypad.

h. No installation or instrumentation is allowed in addition to what is devised within the machine.

i. The locations for supplying pills and the box and also the pickup location of the box must be clearly specified in the machine.

j. Loading pills and the box, delivering the packed box, and retrieving the remaining pills must be convenient to the operator with no need for disassembling any part of the machine.

k. The time required for loading the pieces into the machine, entering the operator's instructions on the keypad, and starting the operation shall not exceed 1 minute. The number of supplied pills must remain undetermined during the loading period, i.e., machine must not pre-count the pills before the operation begins.

l. Each operation is considered "complete" when the correct type/number of pills are dispensed into all compartments, remaining pills are returned to their reservoirs, and the display shows a message indicating the completion of the process.

m. At the end of each operation, the machine display must be on prompt to show the following information per operator's request: operation time, number of remaining pills in each reservoir, and a summary of the instruction parameters.

n. The machine user interface for both operation and information retrieval shall be self-explanatory, and provide easy navigation for users of various skill levels.

o. Each compartment is packaged “correctly” if all required pills are dispensed in correct numbers; otherwise the compartment packaging is considered as “incorrect.”

p. Each compartment is closed “completely” only if its lid is completely closed and snapped.

q. Each pill (in the compartment, reservoir or machine) is considered as “damaged” if there are clear defects as a result of the operation, to the referee’s discretion. The box is considered as “damaged” if there are clear defects as a result of the operation, to the referee’s discretion, e.g., obvious scratches or deformations, a lid is detached or does not close/snap, etc.

r. The box is considered “unavailable” for pickup if it is not completely located in the pickup place designated in the machine, or is jammed and cannot be removed from the machine.

s. The operation time is the duration between when the <start> button on the keypad is pressed and when the machine shows the completion or termination message on its LCD. No actuation or sensing must occur in the machine prior to the start of the operation. The operation time shall not exceed 3 minutes. Further, the time required for loading the pills into the machine and entering the operator’s instructions on the keypad before the operation shall not exceed 1 minute.

t. The recorded and displayed operation time is considered “correct” if it is equal to the time measured by the referee 1 second. Otherwise, it is assumed “incorrect.”

u. Each operation is “qualified” for scoring if, in addition to the lack of other disqualification factors (next constraint), the machine delivers the box with a minimum of 3 compartments with pills containing correct orders and the lids of minimum 5 compartments are closed completely, returns to standby mode so that the box can be unloaded normally, displays the completion or termination message at the end of its operation, and is able to communicate the operation information.

v. An operation is “disqualified” if any of the following happens to the machine or the team declares the termination. If the first or second operation is disqualified, the team will have 2 minutes to fix the system and run for the next time, if they wish.
– structurally collapses, falls over, hangs or jams (for more than 3 minutes) with no termination display, or
– terminates the operation before delivering the box with minimum 3 compartments containing correct orders and the lid of minimum 5 compartments is closed completely, or
– does not display the termination or completion message on the LCD at the end of operation, or
– is not able to communicate with the operator after termination/completion of the operation, or
– runs longer than 3 minutes before terminating the operation, or

– takes more than 1 minute to load pills and the box in the machine and start the operation.

w. Each team will have a period of maximum 1 minute to set up the machine before it is ready to load pills and the box for each operation. (This time is extended to 2 minutes if the previous operation is disqualified.) If the preparation time exceeds 1 minute, the operation is "disqualified."

x. There will be no control over the conditions of the competition environment.

y. The machine must pose no hazard to the operator, and shall not be perceived as hazardous (e.g., excessive vibration, noise, sporadic movement, or electric sparks during the operation is perceived as dangerous.)

## Appendix C. AHP Decision-Making Process for Subsystems (From Proposal)

### 4.2.1. Box Manipulation

#### 4.2.1.1. Box Movement

The machine must be precise enough to dispense pills into each individual container, in addition to being robust to uncertainties involved with the user. This includes robustness to the orientation of the box in addition to the collection area. It therefore becomes necessary to manipulate either the positioning of the box itself or the dispenser. Additionally, the following factors were given strong consideration, which corroborates with the AHP analysis.

**Degrees of motion:**
A heuristic adapted by the design team is to reduce the degrees of motion necessary for dispensing. A binary system (i.e. a motor only needs to switch between two positions) is preferred to a continuous solution to reduce complexity. Physical constraining mechanisms are given precedence over motors and other electronic actuators, as they were considered to be more cost effective and less complex. As each electronic actuator can perform 1 degree of motion (rotational or translational), reducing the number of degrees of motion that the mechanism requires reduces the number of such actuators needed, which cuts down on cost. In addition, more degrees of motion means greater likelihood of compatibility errors between two or more degrees of motion.

**Size:**
Mechanisms for box manipulation opens up the possibility of moving the box to various areas of the machine for processing. As the box itself has a length of 19 cm, or 38% of the maximum

possible length of the machine, a naive design approach may quickly result in a mechanism that exceeds design constraints.

**Reliability:**

Box manipulation involves the action of opening the lids of each individual container so pills can be dispensed the structural integrity of the container itself is maintained, in addition to closing each lid after the pills have been dispensed. As this process involves working with containers and pills with dimensions on the order of ~1cm, even a small error of a few millimeters may result in misplacement of the pills or damage to the box.

The following is an AHP analysis to determine the relative importance of criteria relevant to the Box Manipulation subsystem:

| | Criterion | Comment | Weights | Rk |
|---|---|---|---|---|
| 1 | Criterion 1 | Geometry | 3.2% | 7 |
| 2 | Criterion 2 | Degrees of motion | 20.1% | 2 |
| 3 | Criterion 3 | # of components | 18.1% | 3 |
| 4 | Criterion 4 | Time | 10.9% | 4 |
| 5 | Criterion 5 | Size | 5.9% | 6 |
| 6 | Criterion 6 | Minimize effort | 8.6% | 5 |
| 7 | Criterion 7 | Robustness | 33.3% | 1 |

| | Criterion 1<br>1 | Criterion 2<br>2 | Criterion 3<br>3 | Criterion 4<br>4 | Criterion 5<br>5 | Criterion 6<br>6 | Criterion 7<br>7 | Criterion 8<br>8 | 0<br>9 | 0<br>10 | normalized principal Eigenvector |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1/6 | 1/4 | 1/4 | 1/2 | 1/4 | 1/5 | - | - | - | 3.17% |
| 2 | 6 | 1 | 3 | 1 | 2 | 4 | 1/3 | - | - | - | 20.08% |
| 3 | 4 | 1/3 | 1 | 3 | 5 | 4 | 1/3 | - | - | - | 18.05% |
| 4 | 4 | 1 | 1/3 | 1 | 1 | 3 | 1/5 | - | - | - | 10.95% |
| 5 | 2 | 1/2 | 1/5 | 1 | 1 | 1/4 | 1/4 | - | - | - | 5.88% |
| 6 | 4 | 1/4 | 1/4 | 1/3 | 4 | 1 | 1/3 | - | - | - | 8.59% |
| 7 | 5 | 3 | 3 | 5 | 4 | 3 | 1 | - | - | - | 33.28% |

Consistency index: 13%

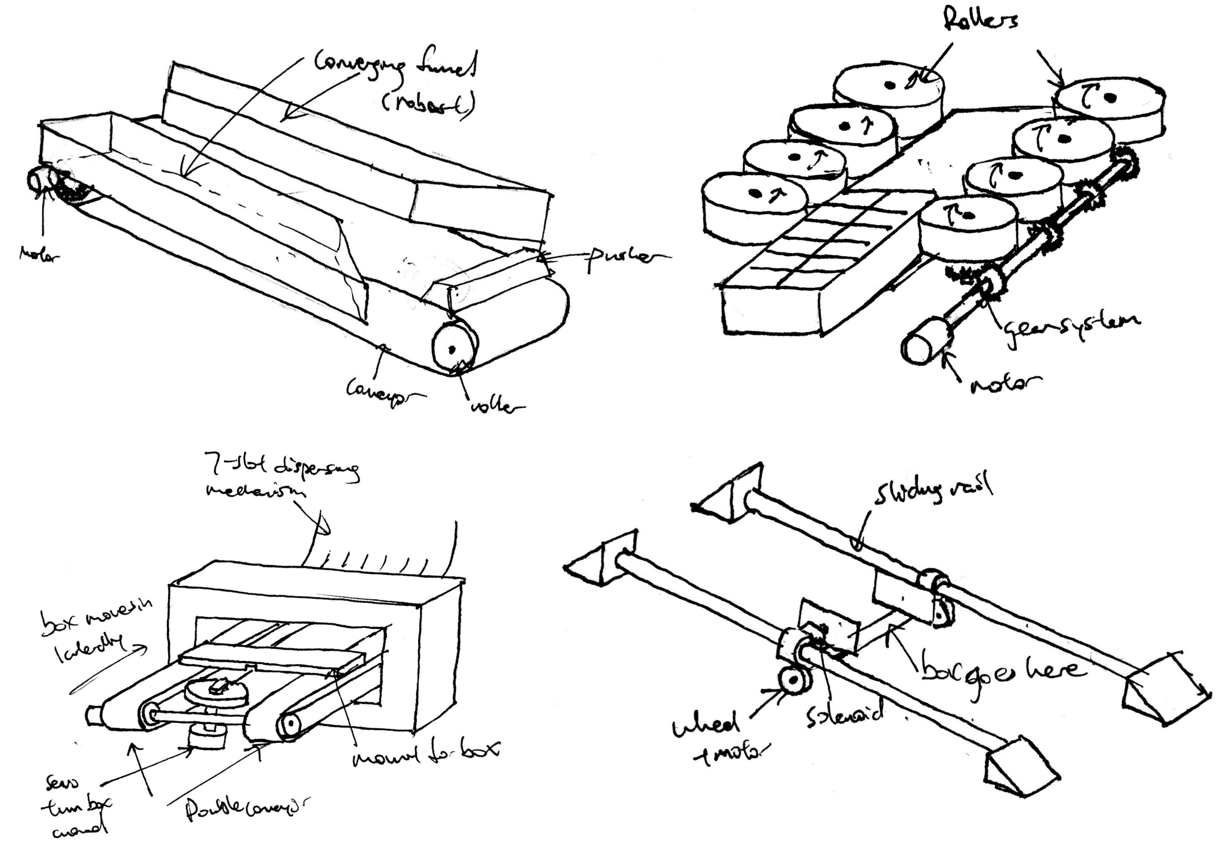
| | | Criteria | | more important ? | Scale |
|---|---|---|---|---|---|
| i | j | A | B | A or B | (1-9) |
| 1 | 2 | Criterion 1 | Criterion 2 | B | 6 |
| 1 | 3 | | Criterion 3 | B | 4 |
| 1 | 4 | | Criterion 4 | B | 4 |
| 1 | 5 | | Criterion 5 | B | 2 |
| 1 | 6 | | Criterion 6 | B | 4 |
| 1 | 7 | | Criterion 7 | B | 5 |
| 1 | 8 | | Criterion 8 | | |
| 2 | 3 | Criterion 2 | Criterion 3 | A | 3 |
| 2 | 4 | | Criterion 4 | B | 1 |
| 2 | 5 | | Criterion 5 | A | 2 |
| 2 | 6 | | Criterion 6 | A | 4 |
| 2 | 7 | | Criterion 7 | B | 3 |
| 2 | 8 | | Criterion 8 | | |
| 3 | 4 | Criterion 3 | Criterion 4 | A | 3 |
| 3 | 5 | | Criterion 5 | A | 5 |
| 3 | 6 | | Criterion 6 | A | 4 |
| 3 | 7 | | Criterion 7 | B | 3 |
| 3 | 8 | | Criterion 8 | | |
| 4 | 5 | Criterion 4 | Criterion 5 | B | 1 |
| 4 | 6 | | Criterion 6 | A | 3 |
| 4 | 7 | | Criterion 7 | B | 5 |
| 4 | 8 | | Criterion 8 | | |
| 5 | 6 | Criterion 5 | Criterion 6 | B | 4 |
| 5 | 7 | | Criterion 7 | B | 4 |
| 5 | 8 | | Criterion 8 | | |
| 6 | 7 | Criterion 6 | Criterion 7 | B | 3 |
| 6 | 8 | | Criterion 8 | | |

Note that, due to the high cost and complexity associated with actuators, one actuator counts as five components in our analysis as a normalization factor.

In addition, other normalization factors were used for time, size and volume. As each subcomponent is not expected to be the size or expected to account for the entire operational time budget of the machine, an estimate for the percentage of the total size and time required for each subcomponent as a function of the entire machine is found. The raw utility values for each individual subcomponent is then divided by this scaling factor in order to calculate for the effective utility value.

| Placement | Design #1 | | Design #2 | | Design #3 | | Design #4 | | | Weights | | Scaling Factors | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normalizec | Utility | Normalizec | Utility | Normalizec | Utility | Normalizec | Utility | | | | | |
| Design | Single Conveyor Belt | | Rollers | | Track | | Lateral | | | | | Time | 0.2 |
| Metrics | | | | | | | | | | | | Size | 0.85 |
| Geometry | 2.88 | 0.91 | 3.10 | 0.89 | 2.67 | 0.93 | 3.07 | 0.89 | | 0.03 | | Volume | 1 |
| Degrees of motior | 1.00 | 0.61 | 1.00 | 0.61 | 1.00 | 0.61 | 2.00 | 0.37 | | 0.20 | | | |
| # of Components | 1600.00 | 0.90 | 2500.00 | 0.75 | 3833.33 | 0.41 | 5416.67 | 0.00 | | 0.18 | | | |
| Time | 1.88 | 0.69 | 1.88 | 0.69 | 1.88 | 0.69 | 2.71 | 0.32 | | 0.11 | | | |
| Size | 0.59 | 0.82 | 0.59 | 0.82 | 0.56 | 0.89 | 0.44 | 1.00 | | 0.06 | | | |
| Minimize effort | 1.00 | 1.00 | 2.00 | 0.75 | 2.00 | 0.75 | 2.00 | 0.75 | | 0.09 | | | |
| Robustness | 10.00 | 1.00 | 5.00 | 0.50 | 2.00 | 0.20 | 10.00 | 1.00 | | 0.33 | | | |
| | | | | | | | | | | | | | |
| Weighted sum | | 0.86 | | 0.64 | | 0.48 | | 0.59 | | | | | |

Sketches of Designs (1 to 4 in clockwise order):



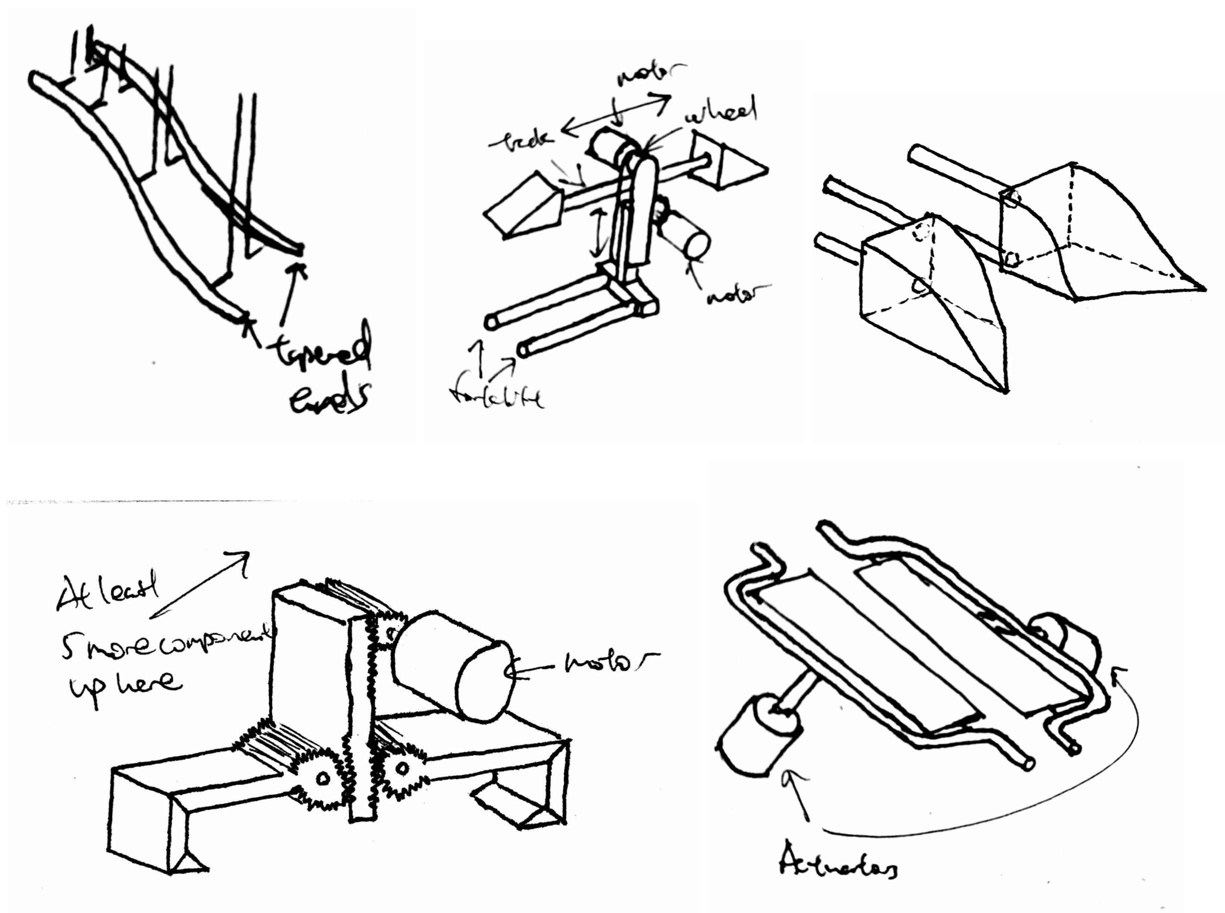

From the AHP and utility-based analysis, it was determined that the single conveyor belt design with a pusher block would be strongly preferred for the box movement system. Its relative ease of construction and good reliability potential is desired by the design team.

### 4.2.1.2. Opening the Lids

As per the RFP, the user would clip open all the lids of the pillbox and leave it in a rest position before placing it inside the machine for processing. At rest position, there would be a small gap ,~5 mm at the widest, between the lid and the box structure where a mechanism can reach inside and open the lid. Using the same set of criteria from above ,excluding “minimize [user] effort” as the process is autonomous.

| Opening the Lids | Design #1 | | Design #2 | | Design #3 | | Design #4 | | Design #5 | | Weights | | Scaling Factors | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normalizec | Utility | Normalizec | Utility | Normalizec | Utility | Normalizec | Utility | Normalizec | Utility | | | | |
| Design | Double Rod | | Forklift | | Double Wedge | | Double Actuator | | Claw | | | | Time | 0.05 |
| Metrics | | | | | | | | | | | | | Size | 0.25 |
| Geometry | 1.80 | 1.00 | Complex | 0.00 | 3.00 | 0.90 | 3.67 | 0.83 | Complex | 0.00 | 0.03 | | Volume | 3.00 |
| Degrees of motior | 0.00 | 1.00 | 2.00 | 0.37 | 0.00 | 1.00 | 2.00 | 0.37 | 2.00 | 0.37 | 0.20 | | | |
| # of Components | 4629.63 | 0.14 | 8333.33 | 0.00 | 4166.67 | 0.31 | 2962.96 | 0.65 | 4861.11 | 0.05 | 0.18 | | | |
| Time | 0.00 | 1.00 | 1.67 | 0.77 | 0.00 | 1.00 | 0.50 | 1.00 | 7.00 | 0.00 | 0.11 | | | |
| Size | 0.84 | 0.20 | 0.88 | 0.00 | 0.64 | 0.69 | 1.09 | 0.00 | 0.70 | 0.54 | 0.06 | | | |
| Robustness | 3.00 | 0.30 | 3.00 | 0.30 | 3.00 | 0.30 | 2.00 | 0.20 | 10.00 | 1.00 | 0.33 | | | |
| | | | | | | | | | | | | | | |
| Weighted sum | | 0.52 | | 0.28 | | 0.58 | | 0.43 | | 0.49 | | | | |

Sketches of Designs (1 to 5 in clockwise order):




From the utility-based analysis, it was found that the double wedge design was somewhat preferred for opening the lids of the pillbox. It was quite simple to construct. From the testing of a mostly functional prototype of the double wedge design, it was found that the box can slide underneath the wedge with relatively ease while the lids are opened. This design also integrates well with the component that drops the pills into the pillbox (discussed in Section 4.2.2).

#### 4.2.1.3. Closing the Lids

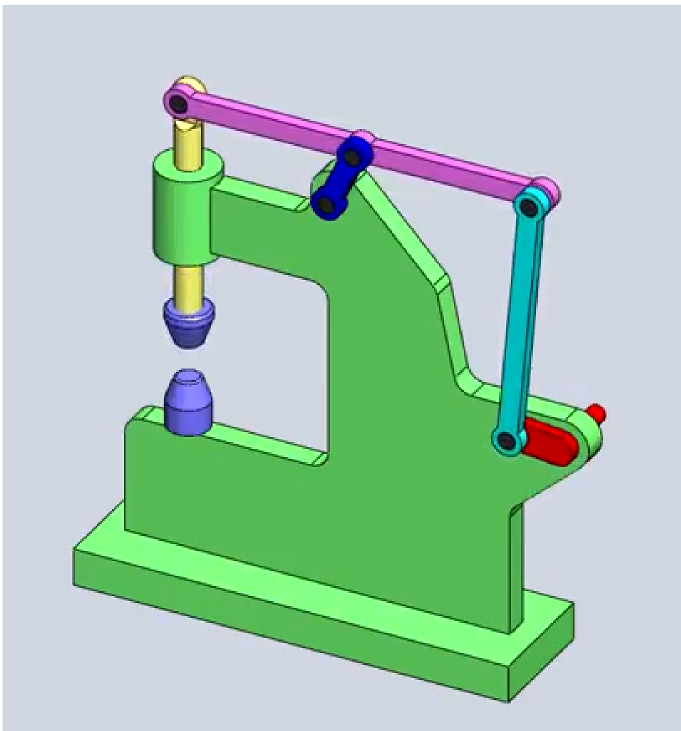
As per the RFP, the lids of the pillbox needs to be closed and locked in place when it is returned to the user. Intuitively, this action would require a force compressing down onto the lids as the pillbox moves forward on the conveyor belt.

From testing out several different designs (including a roller, a flat plate, a block of wood, and pressers), however, it was found that the lid closing mechanism requires the application of not only force, but force in a specific location and direction. A pressure force that possesses a purely vertical component (press) was found to be unfeasibly large due to the snapping mechanism of the lid, while forces that act purely in the horizontal direction (rollers) were found to press on and damage the lid (figure on right). Through the use of a Newton meter for force analysis, it was found that directional or impulsive force application is more effective. Methods for increasing the effective applied force include:


- Dropping a mass onto the lid applies a sharp impulse, which snaps the lid closed even with a relatively small mass and drop height
- Due to the lids opening from the centroid of the pillbox, it was found that applying force towards the outer edge of the lid increases the moment arm, hence increasing torque and reducing the amount of force required
- Decreasing the contact surface area increases local pressure, which is more effective in closing the lid than a larger contact surface area, and
- A block with a larger second moment of area is more effective than a thin plate, as less energy goes into elastic deformation and buckling.

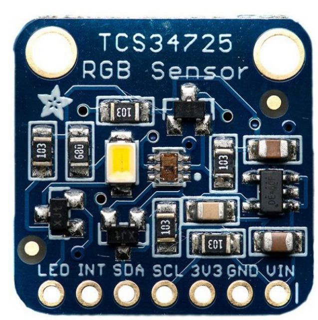
Therefore, designs such as a single roller, a flat plate, and a plain wooden block that apply only downwards force are ruled out. A block with a more precisely manufactured shape is needed to provide both downwards compression, as well as a horizontal force, and limit the contact surface area. An actuator is also required to provide linear downwards motion and impulse.

Upon research of possible designs online, one particular design was considered (figure below). It involved a set of bars that move to transfer motion from a motor, and a piece that presses straight down. However, instead of using a round presser piece, the piece would be precisely manufactured to clamp down the lids from both sides, providing both vertical and horizontal force. Although this structure is relatively complex, it was deemed to be more reliable.



#### 4.2.1.4. Checking Box Orientation

There exists the need to differentiate between the two orientations of the box, assuming that the user places the box on the conveyor and does not check which side it is on. The only physical differentiating features between the two sides are the lid colors in addition to the text on the board. Reading text requires OCR, which involves a camera, and this was deemed to be beyond our budget.



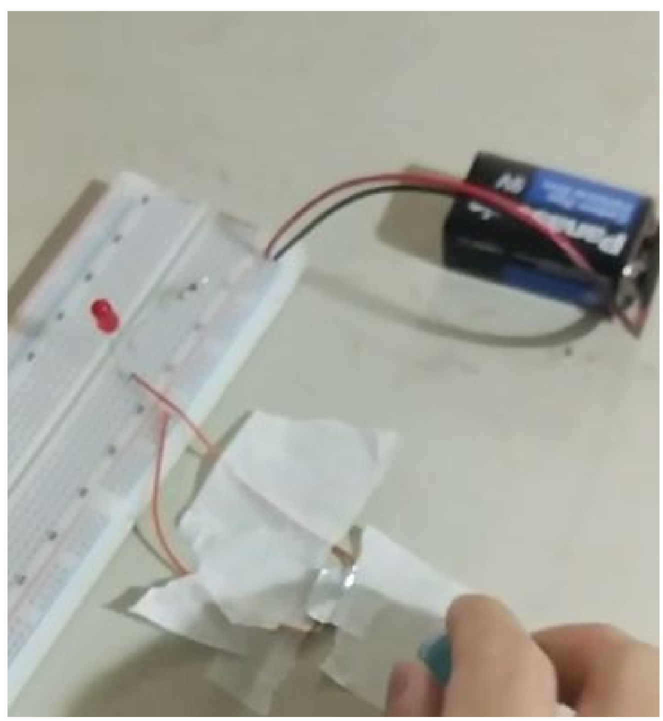

An RGB color sensor (shown on right figure) could then used, as this provides the necessary degree of sensitivity at an acceptable price range to detect the color of the lid and hence the orientation of the box.

### 4.2.2. Pill Counting

As per the RFP, the machine needs to count the remaining number of pills for each of the three types of pills. At first, the following options were considered.

| Design | Description |
|---|---|
| Pressure sensor | A small spring-loaded plate, with two wires on each side. Upon a detection event, the plate elastically deform due to the contact with the pill and completes the circuit, sending a signal to the microcontroller. This design is inexpensive and easy to manufacture; however, it needs a larger mass in order to trigger. |
| Light sensor | A LED with a photodiode. A passing pill lowers transmittance to the photodiode, creating a detectable dip in light intensity which can then be used as a detection event. This sensor has low latency, but has a higher cost and is not robust against external influences (for example, flipping the light switch could potentially trigger the sensor). |
| Force sensor | The force sensor interfaces with the PIC board by sending force values, in newtons. By thresholding this value it becomes possible to detect an event. It is more sensitive than the pressure sensor, but is more expensive to implement and solutions need to be implemented, be it hardware or software, to debounce the signal. |

As per constraint o [Appendix A], there cannot be any errors (i.e. counting the wrong number of pills).



A pressure sensor was then prototyped with aluminum acting as the plate, with two wires on each side. This sensor achieved satisfactory performance, detecting each individual event despite the low fidelity. This sensor was found to be unable to detect the ellipsoid pills rolling on a flat surface; this was remedied by dropping the pills from a height. The sensor was ultimately chosen due to its accuracy. It is also vastly more cost-effective and simple compared to the other designs; the cost for 2 wires and a strip of aluminum is negligible.

### 4.2.3. Pill Dispensing

As per the RFP, there can be three separate compartments for the user to dispense three different types of pills. The pills then drop down one at a time from the compartments to the pill distribution system to be loaded into the pillbox (one pill at a time ensures that the right number of pills are loaded).

Although the three compartments can have a similar outline and structure, since the three types of pills greatly differ in shape and size, the specifications of each compartment may be different.

From testing out several prototypes of systems that control pill dropping (including basic funnels, linear actuator pushing the pills into a slot, cylinder with rotating base with slot, rotating catchers), it was found that:

- Basic funnel-shaped dispensers without further modification are prone to jamming as the pills pile together.
- Simply pushing and dropping the pills through a slit is unreliable, because if the slit is too large, several pills may fit through at once; and on the other hand, if the slit is too small, jamming could be an issue.
- Using a rotating catcher (the catcher has slots that can only fit one pill at a time, and is powered by a motor) would be difficult for the long pill, since it does not always enter the catcher at the same orientation. Building additional slider ramps would increase complexity and consume too much space inside the machine.
- Attaching viberators to the dispenser to prevent jamming would create too much noise; and the vibration may destabilize the parts.

For these reasons, designs such as the basic funnel, boxes with a pusher and slit, and the rotating catcher were disregarded.

The remaining designs were evaluated using an AHP-ranked metrics system, and a utility-based analysis. Some of these designs drew inspiration from personal design projects on Youtube. [9][10][11][12]

| | Criterion | Comment | Weights | Rk |
|---|---|---|---|---|
| 1 | Criterion 1 | # of Components | 18.9% | 2 |
| 2 | Criterion 2 | Geometry | 13.5% | 3 |
| 3 | Criterion 3 | Error Rate | 62.3% | 1 |
| 4 | Criterion 4 | Size | 5.3% | 4 |

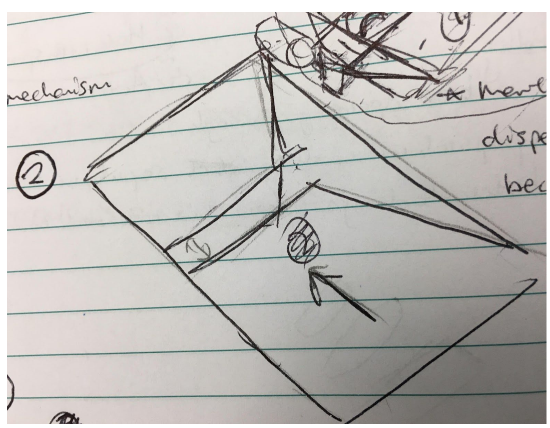
| Matrix | | Criterion 1 | Criterion 2 | Criterion 3 | Criterion 4 | Criterion 5 | Criterion 6 | Criterion 7 | Criterion 8 | 0 | 0 | normalized principal Eigenvector |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Criterion 1 | 1 | - | 2 | 1/5 | 4 | - | - | - | - | - | - | 18.95% |
| Criterion 2 | 2 | 1/2 | - | 1/5 | 4 | - | - | - | - | - | - | 13.48% |
| Criterion 3 | 3 | 5 | 5 | - | 7 | - | - | - | - | - | - | 62.26% |
| Criterion 4 | 4 | 1/4 | 1/4 | 1/7 | - | - | - | - | - | - | - | 5.31% |

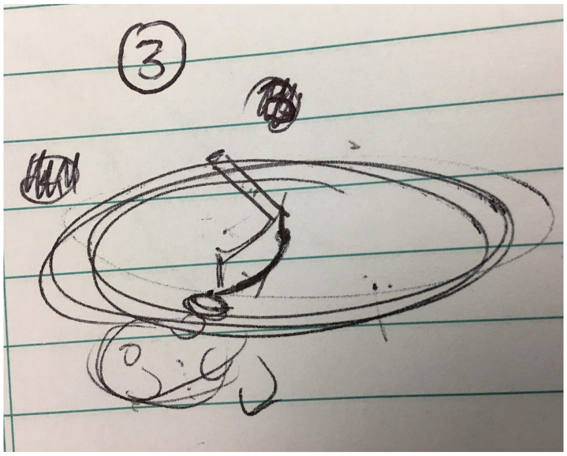
-Consistency ratio: 8%

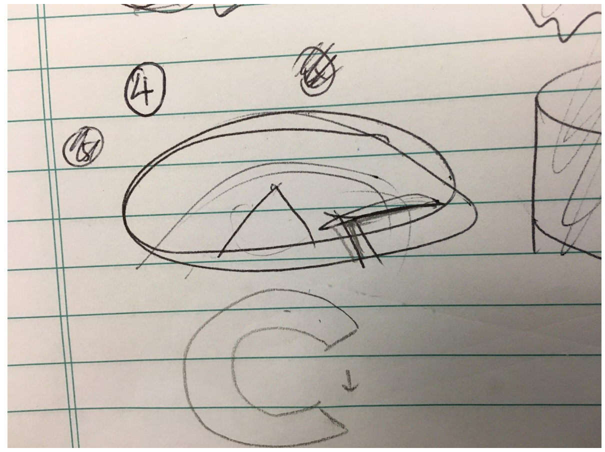
| | Design #1 | | Design #2 | | Design #3 | | Design #4 | | Weights | Scaling Factors | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normalized | Utility | Normalized | Utility | Normalized | Utility | Normalized | Utility | | | |
| Design | | | | | | | | | | Size | 0.3 |
| Metrics | | | | | | | | | | | |
| Geometry | 3.45 | 0.85 | High | 0.00 | 2.00 | 1.00 | 3.00 | 0.90 | 0.19 | | |
| # of Components | 2500.00 | 0.75 | 5833.33 | 0.00 | 10000.00 | 0.00 | 4000.00 | 0.36 | 0.13 | | |
| Success Rate | 0.95 | 0.14 | 0.87 | 0.01 | 0.98 | 0.45 | 1.00 | 1.00 | 0.62 | | |
| Size | 0.62 | 0.75 | 0.67 | 0.63 | 0.53 | 0.97 | 0.53 | 0.97 | 0.05 | | |
| | | | | | | | | | | | |
| Weighted sum | | 0.39 | | 0.04 | | 0.52 | | 0.89 | | | |

*One motor was discounted since all designs have at least one.

Design 1: Slot with rotating collector
Design 2: Simple slot
Design 3: Disk with rotating strip
Design 4: Cylinder with rotating disk with a slot and an opening






In the end, it was determined that the cylindrical dispenser with a motor-driven rotating base, which has a slit cut in the base to only allow for one pill to be dropped each time, would be used for the pill dispensing mechanism (Design 4). It has a significant edge over the other designs on its reliability.

### 4.2.4. Pill Distribution

Based on the definition of this subsystem, the pill distribution subsystem has two components: the component which guides the pills into the correct pillbox compartment, and the component that guides the remaining pills into the collection reservoir (although the two parts can be combined).

Since pill distribution is the part that determines the overall functionality and reliability of the machine, both the error rate and the running time are important parameters.

The following designs were mostly reference designs from existing projects, with some modifications (such as the removable plate in Design 4).

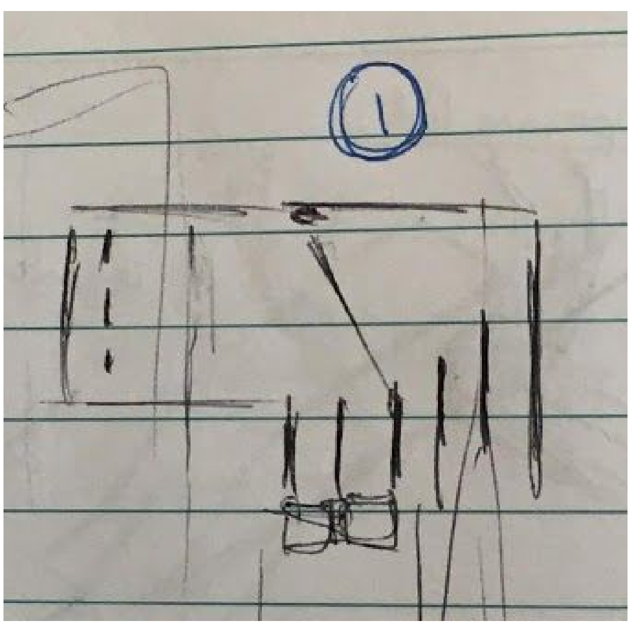
| | Criterion | Comment | Weights | Rk |
|---|---|---|---|---|
| 1 | Criterion 1 | # of Components | 12.3% | 3 |
| 2 | Criterion 2 | Geometry | 11.3% | 4 |
| 3 | Criterion 3 | Error Rate | 36.5% | 1 |
| 4 | Criterion 4 | Size | 3.9% | 6 |
| 5 | Criterion 5 | Degrees of Motions | 9.7% | 5 |
| 6 | Criterion 6 | Time | 26.3% | 2 |

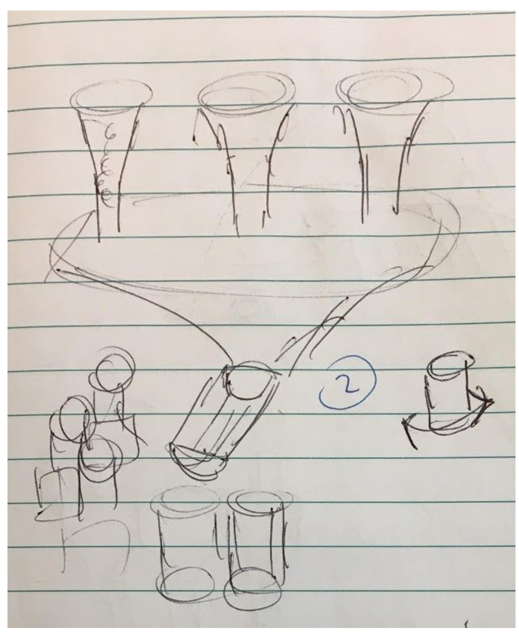
| Matrix | | Criterion 1 | Criterion 2 | Criterion 3 | Criterion 4 | Criterion 5 | Criterion 6 | Criterion 7 | Criterion 8 | 0 | 0 | normalized principal Eigenvector |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Criterion 1 | 1 | - | 2 | 1/4 | 4 | 1 | 1/3 | - | - | - | - | 12.27% |
| Criterion 2 | 2 | 1/2 | - | 1/4 | 5 | 2 | 1/4 | - | - | - | - | 11.32% |
| Criterion 3 | 3 | 4 | 4 | - | 6 | 4 | 1 1/2 | - | - | - | - | 36.48% |
| Criterion 4 | 4 | 1/4 | 1/5 | 1/6 | - | 1/3 | 1/4 | - | - | - | - | 3.94% |
| Criterion 5 | 5 | 1 | 1/2 | 1/4 | 3 | - | 1/2 | - | - | - | - | 9.67% |
| Criterion 6 | 6 | 3 | 4 | 2/3 | 4 | 2 | - | - | - | - | - | 26.31% |

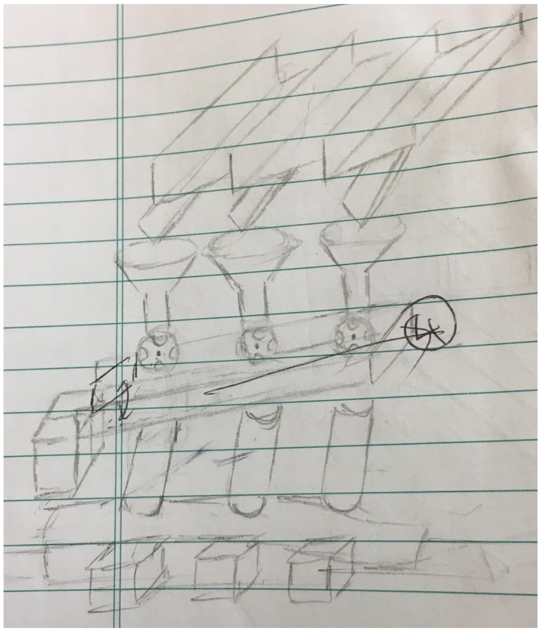
**Consistency ratio:** 6%

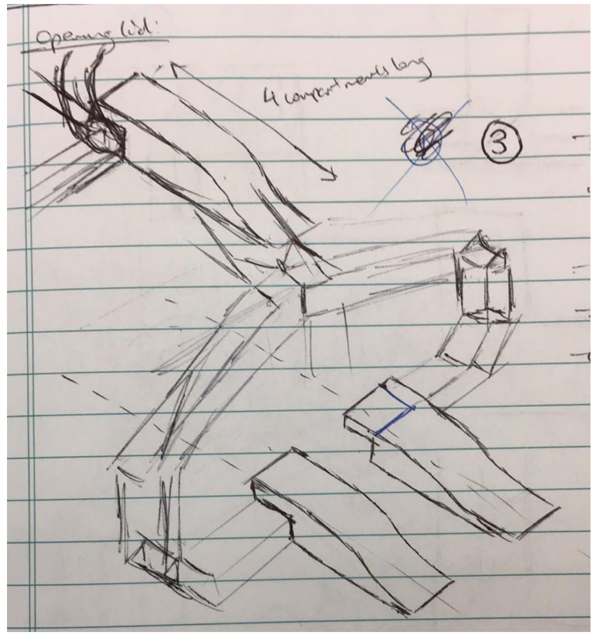
| | Design #1 | | Design #2 | | Design #3 | | Design #4 | | Weights | Scaling Factors | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normalized | Utility | Normalized | Utility | Normalized | Utility | Normalized | Utility | | | |
| **Design** | | | | | | | | | | Size | 0.5 |
| **Metrics** | | | | | | | | | | Time | 0.65 |
| # of Components | 1111.11 | 0.95 | 2777.78 | 0.69 | 2720.00 | 0.70 | 2166.67 | 0.81 | 0.12 | | |
| Geometry | 4.00 | 0.80 | 3.80 | 0.82 | 8.55 | 0.35 | high | 0.00 | 0.11 | | |
| Success Rate | 0.92 | 0.04 | 0.90 | 0.02 | 0.99 | 0.67 | 0.99 | 0.67 | 0.36 | | |
| Size | 0.81 | 0.28 | 0.73 | 0.47 | 0.81 | 0.27 | 0.75 | 0.43 | 0.04 | | |
| Degree of Motions | 1.00 | 0.61 | 1.00 | 0.61 | 4.00 | 0.14 | 2.00 | 0.37 | 0.10 | | |
| Time | 2.56 | 0.39 | 2.05 | 0.61 | 1.79 | 0.72 | 1.79 | 0.72 | 0.26 | | |
| | | | | | | | | | | | |
| **Weighted sum** | | **0.39** | | **0.42** | | **0.58** | | **0.59** | | | |

Design 1: 5-channels with rotating barrier
Design 2: Rotating tube
Design 3: Side loading into the pillbox
Design 4: “Pinball mechanism” and movable base plate





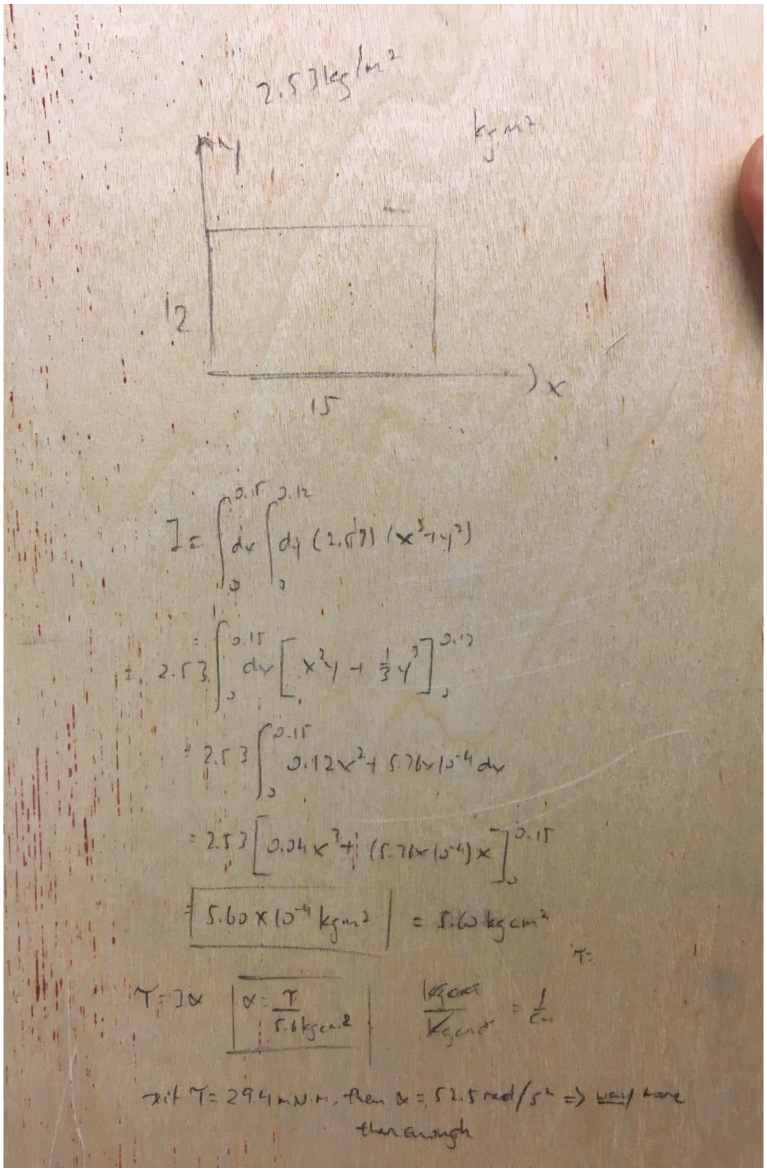



From the AHP and utility-based analysis, the “pinball mechanism” design was chosen since it has a slightly higher edge on reliability over the other designs.

## Appendix D. Testing Results and Calculations

### 1. Torque calculation for base plate of pill distribution funnel

The following image is the torque calculation needed to determine whether the FEETECH FS90 servo is able to turn the base plate at a high enough speed. Torque and moment of inertia was taken about the corner of the plate, and the equation (T = Ia) was applied. Using a torque rating of 29.4 mNm for the motor [], the highest angular speed that can be achieved is 52.5 rad/s, which is far more than necessary.




**2. Cylinders: consistency of pill dispensing (with original contact sensors (Fig. _))**

Each cylindrical dispenser was loaded with 45 pills (the maximum capacity). For 3 cycles, the cylinders were unloaded continuously in one run until the pills run out. The number of rotations that dropped exactly one pill was recorded. If the disk jams due to a misplaced pill, it counts as one error, and the operation continues after the pill is cleared. This number was then divided by the total expected rotations (45) to get a percentage success rate. The results are as follows:

| **Dispenser** | Test 1: # of successes | Success Rate (%) | Test 2: # of successes | Success Rate (%) | Test 3: # of successes | Success Rate (%) | Avg. Success Rate (%) |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Spherical** | 45 | 100 | 43 | 95.6 | 43 | 95.6 | 97.0 |
| **Long** | 23 | 51.1 | 27 | 60.0 | 22 | 48.9 | 53.3 |
| **Flat** | 34 | 75.6 | 32 | 71.1 | 29 | 64.4 | 70.3 |

-Spherical: 97% (occasionally skips pills)
-Long: 53% (often stand up vertically and jams when it hits the barrier)
-Flat: 70% (sometimes stand up vertically and jams when it hits the barrier)

**3. Cylinders: percentage of rotations registered by the contact sensors:**

The consistency of the contact sensors was determined by running the cylinders 25 times, and determining the number of turns for which the contact sensor registers a signal. A program was created on the PIC such that every time an interrupt is received from the contact sensor, a counter is incremented. The results are as follows:

| **Dispenser** | Test 1: # of successes | Success Rate (%) | Test 2: # of successes | Success Rate (%) | Test 3: # of successes | Success Rate (%) | Avg. Success Rate (%) |
|---|---|---|---|---|---|---|---|
| **Spherical** | 25 | 100 | 24 | 96 | 22 | 88 | 94.7 |
| **Long** | 21 | 84 | 18 | 72 | 14 | 56 | 70.7 |
| **Flat** | 18 | 72 | 17 | 68 | 13 | 52 | 64 |

It was found that if the positioning of the contact sensors was not adjusted, deformation of the wires caused them to lose consistency over time.

**4. Testing consistency of pill drop detection with IR sensors**

To test the effectiveness of IR sensors, the cylinders were tested one at a time. For each pill type, the cylinder was loaded with 25 pills, and a program was used to count the number of times the converted analog voltage dropped below the threshold calibrated for that pill type. The results are as follows:

| **Dispenser** | Test 1: # of successes | Success Rate (%) | Test 2: # of successes | Success Rate (%) | Avg. Success Rate (%) |
|---|---|---|---|---|---|
| **Spherical** | 25 | 100 | 25 | 100 | 100 |

| | | | | | |
|---|---|---|---|---|---|
| **Long** | 24 | 96 | 25 | 100 | 98 |
| **Flat** | 25 | 100 | 24 | 96 | 98 |

These results show that if the IR sensors are properly calibrated and not moved around, then they can perform very consistently.

**5. Pill distribution: percentage of pills that end up successfully in the collection reservoir:**
This test was performed to determine the percentage of time that collection channels below the distribution funnel successfully caught pills falling from the cylinders. 45 pills were loaded into each cylinder, and the number of each pill that ended up in the collection reservoirs were counted. The results are as follows:

Original channels:

| **Dispenser** | Test 1: # of successes | Success Rate (%) | Test 2: # of successes | Success Rate (%) | Test 3: # of successes | Success Rate (%) | Avg. Success Rate (%) |
|---|---|---|---|---|---|---|---|
| **Spherical** | 34 | 75.6 | 25 | 55.6 | 38 | 84.4 | 71.9 |
| **Long** | 6 | 13.3 | 13 | 28.9 | 10 | 22.2 | 21.5 |
| **Flat** | 17 | 37.8 | 23 | 51.1 | 8 | 17.8 | 35.6 |

New channels:

| **Dispenser** | Test 1: # of successes | Success Rate (%) | Test 2: # of successes | Success Rate (%) | Avg. Success Rate (%) |
|---|---|---|---|---|---|
| **Spherical** | 40 | 88.9 | 43 | 95.6 | 92.2 |
| **Long** | 45 | 100 | 45 | 100 | 100 |
| **Flat** | 45 | 100 | 42 | 93.3 | 96.7 |

**6. Conveyor contact sensor reliability testing**
The conveyor belt contact sensor was tested by performing multiple runs and determining the number of times that the pill box stopped, out of 7. This was done for two pusher block designs: one with a simple stretch of wire across it and one with an aluminum sheet bent around it. The results are as follows:

| Test # (design 1) | Number of stops | Test # (design 2) | Number of stops |
|---|---|---|---|
| 1 | 7 | 1 | 5 |
| 2 | 4 | 2 | 4 |
| 3 | 5 | 3 | 5 |
| 4 | 0 | 4 | 3 |
| 5 | 1 | 5 | 0 |

As with the dispenser contact sensors, it was found that if the conveyor contact sensor was not re-adjusted every time, it slowly loses reliability due to deformation.

**7. Force required to pull box through the machine**

The force required to pull the box through the machine was tested by fastening the box to a piece of fabric, attaching a string between the fabric and and a luggage scale, then pulling it through. The results from several trials are shown below.

| Trial # | Max. Force (lb) |
|---|---|
| 1 | 2.6 |
| 2 | 1.5 |
| 3 | 1.3 |
| 4 | 1.9 |

The max recorded force amongst all trials was 2.6 lb, or 1.18 kg. Using ¾ in radius of the PVC rollers, this force generates a torque of 2.25 kg cm on the conveyor motor. This is less than the torque rating of 7.9 kg cm for the motor selected, which affirms that this motor is sufficient for driving the conveyor.

**8. Deformation testing of conveyor**

This test was done to determine the deformation of the fabric used for the conveyor belt to see if it is an appropriate material. A weight of 11.21 lb, which is larger than the maximum force required to pull the box, was attached to the end of the fabric, and the elongation was measured. This was determined to be a mere 3%, meaning that fabric is a suitable material.

**9. Time calculation for dispensing process**

This calculation was done to verify that the runtime of the packaging process is less than the 3 minute limit []. The conveyor DC motor's rotation rate is 50 rpm, or 5.236 rad/s. Since the radius of the rollers is ¾ in, this corresponds to a movement speed of 9.97 cm/s of the conveyor belt. Dividing 49 cm, the length of the conveyor belt, by this number gives 4.91 s, which is the amount of time it takes the box to move through the machine. The rotation rate of the dispenser cylinders using a 12% duty cycle is approximately 3.14 rad/s, meaning that it takes 0.67 seconds to dispense 1 pill. 4 pills would thus take 2.68 seconds to dispense. Assuming that both afternoon and morning compartments have to be filled, a time of 6.5 seconds to turn the 'pinball' servo back and forth (counting all delays is required). (2(2.68) + 6.5) x 7 = 83.0 seconds, which is the total amount of time needed to for dispensing into the box. Finally, assuming that 45 pills were loaded, dispensing 14 gives 31 remaining, which would take 31 x 0.67 = 20.8 seconds to empty into the collection reservoir. Multiplying this by 3 gives 62.3 seconds. Adding everything together, the machine operation process takes a total of 150.2 seconds or 2 min. 30s, less than the 3 minute time limit.

**-Cotton**

| | |
|---|---|
| Specific gravity | 1.54 |
| Strength (Tenacity) | 3.0 - 4.9 g/d<br>(cotton is 20% stronger when wet)<br>fiber elongation is almost linear to<br>the stress imposed |
| Elasticity | Relatively low |
| Absorbency and Moisture Regain | 7-8% at standard conditions |
| Birefringence | 0.046 |
| Dielectric constant | 3.9-7.5 |
| Resistivity | Order of $10^9$ ohm/cm$^3$ |
| Micronaire | 2.0 - 6.5 (upland cotton) |
| Denier | 0.7 - 2.3 (upland cotton) |
| Length | 0.9 - 1.2 in (upland cotton) |
| Diameter | 9.77 - 27.26 |
| Coefficient of friction | 0.25 (for raw dry cotton, otherwise strongly changes for treated and/or wet fiber) |
| Thermal Properties | Decomposes when exposed at the temperatures about 300$^o$F |

**-Aluminum** (Most likely 6061-T6)
https://www.azom.com/article.aspx?ArticleID=2863

| Property | Value |
|---|---|
| Atomic Number | 13 |
| Atomic Weight (g/mol) | 26.98 |
| Valency | 3 |
| Crystal Structure | FCC |
| Melting Point (°C) | 660.2 |
| Boiling Point (°C) | 2480 |
| Mean Specific Heat (0-100°C) (cal/g.°C) | 0.219 |
| Thermal Conductivity (0-100°C) (cal/cms. °C) | 0.57 |
| Co-Efficient of Linear Expansion (0-100°C) ($x10^{-6}$/°C) | 23.5 |
| Electrical Resistivity at 20°C (Ω.cm) | 2.69 |
| Density ($g/cm^3$) | 2.6898 |
| Modulus of Elasticity (GPa) | 68.3 |
| Poissons Ratio | 0.34 |

| Alloy | Temper | Proof Stress 0.20% (MPa) | Tensile Strength (MPa) | Shear Strength (MPa) | Elongation A5 (%) | Elongation A50 (%) | Hardness Brinell HB | Hardness Vickers HV | Fatigue Endur. Limit (MPa) |
|---|---|---|---|---|---|---|---|---|---|
| AA1050A | H2 | 85 | 100 | 60 | 12 | | 30 | 30 | |
| | H4 | 105 | 115 | 70 | 10 | 9 | 35 | 36 | 70 |
| | H6 | 120 | 130 | 80 | 7 | | 39 | | |
| | H8 | 140 | 150 | 85 | 6 | 5 | 43 | 44 | 100 |
| | H9 | 170 | 180 | | | 3 | 48 | 51 | |
| AA2011 | 0 | 35 | 80 | 50 | 42 | 38 | 21 | 20 | 50 |
| | T3 | 290 | 365 | 220 | 15 | 15 | 95 | 100 | 250 |
| | T4 | 270 | 350 | 210 | 18 | 18 | 90 | 95 | 250 |
| | T6 | 300 | 395 | 235 | 12 | 12 | 110 | 115 | 250 |
| | T8 | 315 | 420 | 250 | 13 | 12 | 115 | 120 | 250 |
| AA3103 | H2 | 115 | 135 | 80 | 11 | 11 | 40 | 40 | |
| | H4 | 140 | 155 | 90 | 9 | 9 | 45 | 46 | 130 |
| | H6 | 160 | 175 | 100 | 8 | 6 | 50 | 50 | |
| | H8 | 180 | 200 | 110 | 6 | 6 | 55 | 55 | 150 |
| | H9 | 210 | 240 | 125 | 4 | 3 | 65 | 70 | |
| AA5083 | 0 | 45 | 105 | 70 | 29 | 25 | 29 | 29 | 100 |
| | H2 | 240 | 330 | 185 | 17 | 16 | 90 | 95 | 280 |
| | H4 | 275 | 360 | 200 | 16 | 14 | 100 | 105 | 280 |
| | H6 | 305 | 380 | 210 | 10 | 9 | 105 | 110 | |
| | H8 | 335 | 400 | 220 | 9 | 8 | 110 | 115 | |
| | H9 | 370 | 420 | 230 | 5 | 5 | 115 | 120 | |
| AA5251 | 0 | 145 | 300 | 175 | 23 | 22 | 70 | 75 | 250 |
| | H2 | 165 | 210 | 125 | 14 | 14 | 60 | 65 | |
| | H4 | 190 | 230 | 135 | 13 | 12 | 65 | 70 | 230 |
| | H6 | 215 | 255 | 145 | 9 | 8 | 70 | 75 | |
| | H8 | 240 | 280 | 155 | 8 | 7 | 80 | 80 | 250 |
| | H9 | 270 | 310 | 165 | 5 | 4 | 90 | 90 | |
| AA5754 | 0 | 80 | 180 | 115 | 26 | 25 | 45 | 46 | 200 |
| | H2 | 185 | 245 | 150 | 15 | 14 | 70 | 75 | |
| | H4 | 215 | 270 | 160 | 14 | 12 | 75 | 80 | 250 |
| | H6 | 245 | 290 | 170 | 10 | 9 | 80 | 85 | |
| | H8 | 270 | 315 | 180 | 9 | 8 | 90 | 90 | 280 |
| | H9 | 300 | 340 | 190 | 5 | 4 | 95 | 100 | |
| | 0 | 100 | 215 | 140 | 25 | 24 | 55 | 55 | 220 |
| AA6063 | 0 | 50 | 100 | 70 | 27 | 26 | 25 | 85 | 110 |
| | T1 | 90 | 150 | 95 | 26 | 24 | 45 | 45 | 150 |
| | T4 | 90 | 160 | 110 | 21 | 21 | 50 | 50 | 150 |
| | T5 | 175 | 215 | 135 | 14 | 13 | 60 | 65 | 150 |
| | T6 | 210 | 245 | 150 | 14 | 12 | 75 | 80 | 150 |
| | T8 | 240 | 260 | 155 | | 9 | 80 | 85 | |
| AA6082 | 0 | 60 | 130 | 85 | 27 | 26 | 35 | 35 | 120 |
| | T1 | 170 | 260 | 155 | 24 | 24 | 70 | 75 | 200 |
| | T4 | 170 | 260 | 170 | 19 | 19 | 70 | 75 | 200 |
| | T5 | 275 | 325 | 195 | 11 | 11 | 90 | 95 | 210 |
| | T6 | 310 | 340 | 210 | 11 | 11 | 95 | 100 | 210 |
| AA6262 | T6 | 240 | 290 | | 8 | | | | |
| | T9 | 330 | 360 | | 3 | | | | |
| AA7075 | 0 | 105 | 225 | 150 | | 17 | 60 | 65 | 230 |
| | T6 | 505 | 570 | 350 | 10 | 10 | 150 | 160 | 300 |
| | T7 | 435 | 505 | 305 | 13 | 12 | 140 | 150 | 300 |

**-PVC** http://www.vinidex.com.au/technical/material-properties/pvc-properties/

| Property | Value | Conditions and Remarks |
|---|---|---|
| Physical properties | | |
| Molecular weight (resin) | 140000 | cf: K57 PVC 70,000 |
| Relative density | 1.42 - 1.48 | cf: PE 0.95 - 0.96, GRP 1.4 - 2.1, CI 7.2, Clay 1.8 - 2.6 |
| Water absorption | 0.0012 | 23°C, 24 hours cf: AC 18 - 20% AS1711 |
| Hardness | 80 | Shore D Durometer, Brinell 15, Rockwell R 114, cf: PE Shore D 60 |
| Impact strength - 20°C | 20 kJ/m$^2$ | Charpy 250 μm notch tip radius |
| Impact strength - 0°C | 8 kJ/m$^2$ | Charpy 250 μm notch tip radius |
| Coefficient of friction | 0.4 | PVC to PVC cf: PE 0.25, PA 0.3 |
| Mechanical properties | | |
| Ultimate tensile strength | 52 MPa | AS 1175 Tensometer at constant strain rate cf: PE 30 |
| Elongation at break | 50 - 80% | AS 1175 Tensometer at constant strain rate cf: PE 600-900 |
| Short term creep rupture | 44 MPa | Constant load 1 hour value cf: PE 14, ABS 25 |
| Long term creep rupture | 28 MPa | Constant load extrapolated 50 year value cf: PE 8-12 |
| Elastic tensile modulus | 3.0 - 3.3 GPa | 1% strain at 100 seconds cf: PE 0.9-1.2 |
| Elastic flexural modulus | 2.7 - 3.0 GPa | 1% strain at 100 seconds cf: PE 0.7-0.9 |
| Long term creep modulus | 0.9 - 1.2 GPa | Constant load extrapolated 50 year secant value cf: PE 0.2 - 0.3 |
| Shear modulus | 1.0 GPa | 1% strain at 100 seconds G=E/2/(1+μ) cf: PE 0.2 |
| Bulk modulus | 4.7 GPa | 1% strain at 100 seconds K=E/3/(1-2μ) cf: PE 2.0 |
| Poisson's ratio | 0.4 | Increases marginally with time under load. cf: PE 0.45 |

**-Softwood** (Douglas Fir) http://www.wood-database.com/douglas-fir/

**Common Name(s):** Douglas-Fir

**Scientific Name:** Pseudotsuga menziesii

**Distribution:** Western North America

**Tree Size:** 200-250 ft (60-75 m) tall, 5-6 ft (1.5-2 m) trunk diameter

**Average Dried Weight:** 32 lbs/ft$^3$ (510 kg/m$^3$)

**Specific Gravity (Basic, 12% MC):** .45, .51

**Janka Hardness:** 620 lb$_f$ (2,760 N)

**Modulus of Rupture:** 12,500 lb$_f$/in$^2$ (86.2 MPa)

**Elastic Modulus:** 1,765,000 lb$_f$/in$^2$ (12.17 GPa)

**Crushing Strength:** 6,950 lb$_f$/in$^2$ (47.9 MPa)

**Shrinkage:** Radial: 4.5%, Tangential: 7.3%, Volumetric: 11.6%, T/R Ratio: 1.6

**-Plywood**

| Mill | Type of hardboard | Moisture content (%) | Specific gravity | Modulus of elasticity GPa | Modulus of elasticity (×10$^6$ lb in$^{-2}$) | Modulus of rupture MPa | Modulus of rupture (lb in$^{-2}$) | Ultimate tensile stress MPa | Ultimate tensile stress (lb in$^{-2}$) | Internal bond MPa | Internal bond (lb in$^{-2}$) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1/8-in. standard | 4.6 | 0.9 | 3.83 | (556) | 31.44 | (4,560) | 23.24 | (3,370) | 1.24 | (180) |
| B | | 6.5 | 1.02 | 4.36 | (633) | 33.92 | (4,920) | 23.17 | (3,360) | 2.76 | (400) |
| C | | 5.2 | 0.94 | 4.20 | (609) | 45.85 | (6,650) | 37.58 | (5,450) | 2.17 | (315) |
| D | | 5.6 | 0.9 | 3.32 | (482) | 38.75 | (5,620) | 28.61 | (4,150) | 1.55 | (225) |
| E | | 6.5 | 0.95 | 3.55 | (515) | 47.50 | (6,890) | 32.96 | (4,780) | 3.52 | (510) |
| F | | 7.7 | 0.91 | 3.23 | (468) | 37.85 | (5,490) | 25.72 | (3,730) | 1.93 | (280) |
| B | 1/4-in. standard | 6.4 | 1.02 | 4.45 | (645) | 33.85 | (4,910) | 22.61 | (3,280) | 1.86 | (270) |
| E | | 6.0 | 0.90 | 3.88 | (563) | 38.96 | (5,650) | 23.65 | (3,430) | 1.65 | (240) |
| A | 1/4-in. tempered | 4.9 | 0.99 | 5.30 | (768) | 53.02 | (7,690) | 31.58 | (4,580) | 1.79 | (260) |
| F | 1/4-in. tempered | 6.9 | 0.98 | 5.14 | (745) | 55.57 | (8,060) | 30.61 | (4,440) | 1.86 | (270) |

[a]From McNatt and Myers (1993).

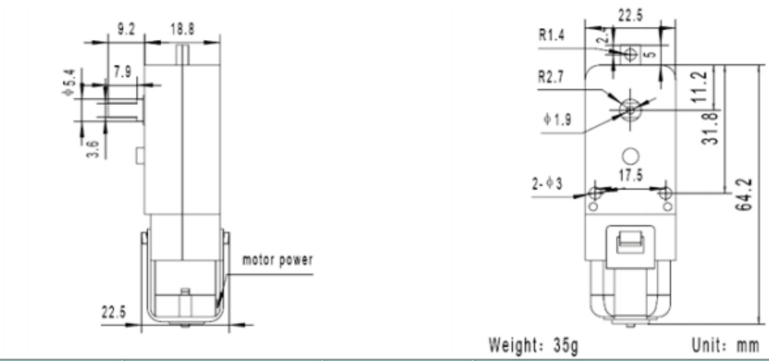
https://www.fpl.fs.fed.us/documnts/fplgtr/fplgtr190/chapter_12.pdf

# Appendix E. Relevant Data Sheet Extracts

## 1. TGP01S-A130 DC Gear Motor

Model：TGP01S-A130
Gear Motor

Applications
Description
Reduction Ratio:
1/48,、1/120、1/180、1/220、1/256、1/288




| MODEL | | VOLTAGE | | NO LOAD | | AT LOAD | | | | | STALL | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | OPERATING RANGE | NOMINAL | SPEED | CURRENT | SPEED | CURRENT | TORQUE | | OUTPUT | TORQUE | | CURRENT |
| | | | V | rpm | A | rpm | A | N.m | Kg.cm | w | N.m | Kg.cm | A |
| TGP01S-A130 | 14150-120 | 3.0-12.0 | 4.5 | 100 | 0.20 | 87 | 0.50 | 0.066 | 0.675 | 0.40 | 0.265 | 2.700 | 1.25 |
| | 18100-220 | 3.0-12.0 | 3.0 | 50 | 0.20 | 40 | 0.60 | 0.120 | 1.233 | 0.6 | 0.393 | 4.003 | 2.15 |

## 2. FS-90 Servo Motor

### FS90--Micro 9g servo for Air plane

**Model No.: FS90**


**Product Description**

**Features**

-Micro analog plastic gears servo
-Operating Voltage:4.8-6Volts
-Interface: (like JR)
-Wire length: 20cm

| | | |
|---|---|---|
| Power | 4.8V | 6V |
| Speed | 0.12sec/60degree | 0.07sec/60degree |
| Torque | 1.3kg.cm/18.9oz.in | |
| Weight | 9g(0.32oz) | |
| Size | 23.2*12.5*22.0mm | |
| Application | for Air plane | |

**Connection description**

Orange = Signal input
Red = +5V
Brown = 0V

**Typical Signals**

**MIN_WIDTH 544 // shortest pulse sent to a servo 0°**
**MAX_WIDTH 2400 // longest pulse sent to a servo 180°**
**NEUTRAL_PULSE_WIDTH 1500 // MID pulse width when servo is at 90°**
**REFRESH_INTERVAL 20000 // min time to refresh in microseconds**

## 3. Creatron 120:1 25D Metal Gear Motor


120:1 25D METAL GEAR MOTOR (6V 50RPM)

MOTOR-100650

This 1.46" x 0.79" x 0.79" gearmotor is a brushed DC motor with 120:1 metal gearbox that has the power to deliver both speed and torque. These units have a 0.276"-long, 4 mm-diameter D-shaped output shaft.

CAD$15.99

1 + - ADD TO CART

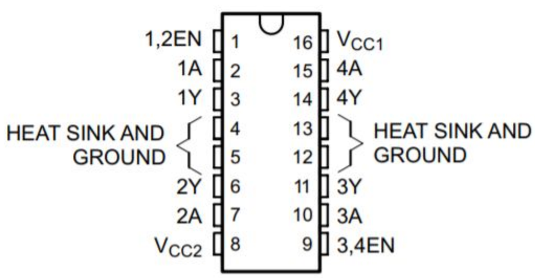
Add to Wishlist

MORE INFO | PRODUCT REVIEWS(0)

**Motor Rating @ 6V:**

- 120:1 gear ratio
- 50RPM
- 100mA free run current
- 3A stall current
- 7.9kg*cm torque
- 4mm shaft diameter

## 4. L293 Dual H-Bridge




**Pin Functions**

| PIN | | TYPE | DESCRIPTION |
|---|---|---|---|
| NAME | NO. | | |
| 1,2EN | 1 | I | Enable driver channels 1 and 2 (active high input) |
| <1:4>A | 2, 7, 10, 15 | I | Driver inputs, noninverting |
| <1:4>Y | 3, 6, 11, 14 | O | Driver outputs |
| 3,4EN | 9 | I | Enable driver channels 3 and 4 (active high input) |
| GROUND | 4, 5, 12, 13 | — | Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias |
| $V_{CC1}$ | 16 | — | 5-V supply for internal logic translation |
| $V_{CC2}$ | 8 | — | Power VCC for drivers 4.5 V to 36 V |

## 5. TCS34725 Color detection chip

TAOS® TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS®

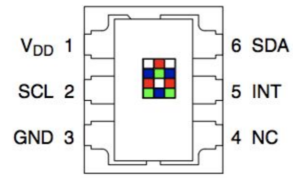
**TCS3472**
**COLOR LIGHT-TO-DIGITAL CONVERTER with IR FILTER**

TAOS135 – AUGUST 2012

### Features

- **Red, Green, Blue (RGB), and Clear Light Sensing with IR Blocking Filter**
  - **Programmable Analog Gain and Integration Time**
  - **3,800,000:1 Dynamic Range**
  - **Very High Sensitivity — Ideally Suited for Operation Behind Dark Glass**
- **Maskable Interrupt**
  - **Programmable Upper and Lower Thresholds with Persistence Filter**
- **Power Management**
  - **Low Power — 2.5-μA Sleep State**
  - **65-μA Wait State with Programmable Wait State Time from 2.4 ms to > 7 Seconds**
- **$I^2C$ Fast Mode Compatible Interface**
  - **Data Rates up to 400 kbit/s**
  - **Input Voltage Levels Compatible with $V_{DD}$ or 1.8 V Bus**
- **Register Set and Pin Compatible with the TCS3x71 Series**
- **Small 2 mm × 2.4 mm Dual Flat No-Lead (FN) Package**

**PACKAGE FN**
**DUAL FLAT NO-LEAD**
**(TOP VIEW)**




Package Drawing Not to Scale

### Applications

- **RGB LED Backlight Control**
- **Light Color Temperature Measurement**
- **Ambient Light Sensing for Display Backlight Control**
- **Fluid and Gas Analysis**
- **Product Color Verification and Sorting**

### End Products and Market Segments

- **TVs, Mobile Handsets, Tablets, Computers, and Monitors**
- **Consumer and Commercial Printing**
- **Medical and Health Fitness**
- **Solid State Lighting (SSL) and Digital Signage**
- **Industrial Automation**

### Description

The TCS3472 device provides a digital return of red, green, blue (RGB), and clear light sensing values. An IR blocking filter, integrated on-chip and localized to the color sensing photodiodes, minimizes the IR spectral component of the incoming light and allows color measurements to be made accurately. The high sensitivity, wide dynamic range, and IR blocking filter make the TCS3472 an ideal color sensor solution for use under varying lighting conditions and through attenuating materials.

The TCS3472 color sensor has a wide range of applications including RGB LED backlight control, solid-state lighting, health/fitness products, industrial process controls and medical diagnostic equipment. In addition, the IR blocking filter enables the TCS3472 to perform ambient light sensing (ALS). Ambient light sensing is widely used in display-based products such as cell phones, notebooks, and TVs to sense the lighting environment and enable automatic display brightness for optimal viewing and power savings. The TCS3472, itself, can enter a lower-power wait state between light sensing measurements to further reduce the average power consumption.

The RGBC engine contains RGBC gain control (AGAIN) and four integrating analog-to-digital converters (ADC) for the RGBC photodiodes. The RGBC integration time (ATIME) impacts both the resolution and the sensitivity of the RGBC reading. Integration of all four channels occurs simultaneously and upon completion of the conversion cycle, the results are transferred to the color data registers. This data is also referred to as channel *count*.

## RGBC Timing Register (0x01)

The RGBC timing register controls the internal integration time of the RGBC clear and IR channel ADCs in 2.4-ms increments. Max RGBC Count = (256 − ATIME) × 1024 up to a maximum of 65535.

**Table 6. RGBC Timing Register**

| FIELD | BITS | DESCRIPTION | | | |
|---|---|---|---|---|---|
| ATIME | 7:0 | **VALUE** | **INTEG_CYCLES** | **TIME** | **MAX COUNT** |
| | | 0xFF | 1 | 2.4 ms | 1024 |
| | | 0xF6 | 10 | 24 ms | 10240 |
| | | 0xD5 | 42 | 101 ms | 43008 |
| | | 0xC0 | 64 | 154 ms | 65535 |
| | | 0x00 | 256 | 700 ms | 65535 |

**Table 11. Control Register**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| **CONTROL** | **Reserved** | | | | | | **AGAIN** | | **Address 0x0F** |

| FIELD | BITS | DESCRIPTION | |
|---|---|---|---|
| Reserved | 7:2 | Reserved. Write bits as 0 | |
| AGAIN | 1:0 | RGBC Gain Control. | |
| | | **FIELD VALUE** | **RGBC GAIN VALUE** |
| | | 00 | 1× gain |
| | | 01 | 4× gain |
| | | 10 | 16× gain |
| | | 11 | 60× gain |

## RGBC Channel Data Registers (0x14 – 0x1B)

Clear, red, green, and blue data is stored as 16-bit values. To ensure the data is read correctly, a two-byte read $I^2C$ transaction should be used with a read word protocol bit set in the command register. With this operation, when the lower byte register is read, the upper eight bits are stored into a shadow register, which is read by a subsequent read to the upper byte. The upper register will read the correct value even if additional ADC integration cycles end between the reading of the lower and upper registers.

**Table 14. ADC Channel Data Registers**

| REGISTER | ADDRESS | BITS | DESCRIPTION |
|---|---|---|---|
| CDATA | 0x14 | 7:0 | Clear data low byte |
| CDATAH | 0x15 | 7:0 | Clear data high byte |
| RDATA | 0x16 | 7:0 | Red data low byte |
| RDATAH | 0x17 | 7:0 | Red data high byte |
| GDATA | 0x18 | 7:0 | Green data low byte |
| GDATAH | 0x19 | 7:0 | Green data high byte |
| BDATA | 0x1A | 7:0 | Blue data low byte |
| BDATAH | 0x1B | 7:0 | Blue data high byte |

## 6. IR Sensors Data Sheets

LTE-302

**ABSOLUTE MAXIMUM RATINGS AT TA=25℃**

| PARAMETER | MAXIMUM RATING | UNIT |
|---|---|---|
| Power Dissipation | 75 | mW |
| Peak Forward Current (300pps, 10 μs pulse) | 1 | A |
| Continuous Forward Current | 50 | mA |
| Reverse Voltage | 5 | V |
| Operating Temperature Range | -40℃ to + 85℃ | |
| Storage Temperature Range | -55℃ to + 100℃ | |
| Lead Soldering Temperature [1.6mm(.063") From Body] | 260℃ for 5 Seconds | |

**ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25℃**

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITION | BIN NO. |
|---|---|---|---|---|---|---|---|
| Aperture Radiant Incidence | Ee | 0.088 | | 0.168 | mW/cm² | $I_F$ = 20mA | BIN B |
| | | 0.112 | | 0.204 | | | BIN C |
| | | 0.136 | | 0.240 | | | BIN D |
| | | 0.160 | | 0.288 | | | BIN E |
| | | 0.192 | | | | | BIN F |
| Radiant Intensity | $I_E$ | 0.662 | | 1.263 | mW/sr | $I_F$ = 20mA | BIN B |
| | | 0.842 | | 1.534 | | | BIN C |
| | | 1.023 | | 1.805 | | | BIN D |
| | | 1.203 | | 2.165 | | | BIN E |
| | | 1.444 | | | | | BIN F |
| Peak Emission Wavelength | $\lambda_{Peak}$ | | 940 | | nm | $I_F$ = 20mA | |
| Spectral Line Half-Width | $\Delta\lambda$ | | 50 | | nm | $I_F$ = 20mA | |
| Forward Voltage | $V_F$ | | 1.2 | 1.6 | V | $I_F$ = 20mA | |
| Reverse Current | $I_R$ | | | 100 | μA | $V_R$ = 5V | |
| Viewing Angle (See FIG.6) | $2\theta_{1/2}$ | | 40 | | deg. | | |

## ABSOLUTE MAXIMUM RATINGS AT TA=25℃

| PARAMETER | MAXIMUM RATING | UNIT |
|---|---|---|
| Power Dissipation | 100 | mW |
| Collector-Emitter Voltage | 30 | V |
| Emitter-Collector Voltage | 5 | V |
| Operating Temperature Range | -40℃ to + 85℃ | |
| Storage Temperature Range | -55℃ to + 100℃ | |
| Lead Soldering Temperature [1.6mm(.063") From Body] | 260℃ for 5 Seconds | |

## ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25℃

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITION | BIN NO. |
|---|---|---|---|---|---|---|---|
| Collector-Emitter Breakdown Voltage | $V_{(BR)CEO}$ | 30 | | | V | $I_C$ = 1mA<br>Ee = 0mW/cm² | |
| Emitter-Collector Breakdown Voltage | $V_{(BR)ECO}$ | 5 | | | V | $I_E$ = 100 $\mu$A<br>Ee = 0mW/cm² | |
| Collector Emitter Saturation Voltage | $V_{CE(SAT)}$ | | | 0.4 | V | $I_C$ = 0.1mA<br>Ee = 1mW/cm² | |
| Rise Time | Tr | | 10 | | $\mu$s | $V_{CC}$ = 5V<br>$I_C$ = 1mA<br>$R_L$ = 1K$\Omega$ | |
| Fall Time | Tf | | 15 | | $\mu$s | | |
| Collector Dark Current | $I_{CEO}$ | | | 100 | nA | $V_{CE}$ = 10V<br>Ee = 0mW/cm² | |
| On State Collector Current | $I_{C(ON)}$ | 0.20 | | 0.60 | mA | $V_{CE}$ = 5V<br>Ee = 1mW/cm²<br>$\lambda$ =940nm | BIN A |
| | | 0.40 | | 1.08 | | | BIN B |
| | | 0.72 | | 1.56 | | | BIN C |
| | | 1.04 | | 1.80 | | | BIN D |
| | | 1.20 | | 2.40 | | | BIN E |
| | | 1.60 | | 3.00 | | | BIN F |
| | | 2.00 | | 3.84 | | | BIN G |
| | | 2.56 | | | | | BIN H |

## ABSOLUTE MAXIMUM RATINGS AT TA=25℃

| PARAMETER | MAXIMUM RATING | UNIT |
|---|---|---|
| Power Dissipation | 100 | mW |
| Peak Forward Current (300pps, 10 μs pulse) | 3 | A |
| Continuous Forward Current | 50 | mA |
| Reverse Voltage | 5 | V |
| Operating Temperature Range | -40℃ to + 85℃ | |
| Storage Temperature Range | -55℃ to + 100℃ | |
| Lead Soldering Temperature [1.6mm(.063") From Body] | 260℃ for 5 Seconds | |

## ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25℃

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITION | BIN NO. |
|---|---|---|---|---|---|---|---|
| Aperture Radiant Incidence | Ee | 0.44 | | 0.96 | mW/cm² | $I_F$ = 20mA | BIN A |
| | | 0.64 | | 1.20 | | | BIN B |
| | | 0.80 | | 1.68 | | | BIN C |
| | | 1.12 | | 1.94 | | | BIN D1 |
| | | 1.30 | | 2.54 | | | BIN D2 |
| | | 1.70 | | 3.14 | | | BIN D3 |
| | | 2.10 | | | | | BIN D4 |
| Radiant Intensity | $I_E$ | 3.31 | | 7.22 | mW/sr | $I_F$ = 20mA | BIN A |
| | | 4.81 | | 9.02 | | | BIN B |
| | | 6.02 | | 12.63 | | | BIN C |
| | | 8.40 | | 14.58 | | | BIN D1 |
| | | 9.72 | | 19.08 | | | BIN D2 |
| | | 12.72 | | 23.58 | | | BIN D3 |
| | | 15.72 | | | | | BIN D4 |
| Peak Emission Wavelength | $\lambda_{Peak}$ | | 940 | | nm | $I_F$ = 20mA | |
| Spectral Line Half-Width | $\Delta\lambda$ | | 50 | | nm | $I_F$ = 20mA | |
| Forward Voltage | $V_F$ | | 1.2 | 1.6 | V | $I_F$ = 20mA | |
| Reverse Current | $I_R$ | | | 100 | μA | $V_R$ = 5V | |
| Viewing Angle (See FIG.6) | $2\theta_{1/2}$ | | 20 | | deg. | | |

| PARAMETER | MAXIMUM RATING | UNIT |
|---|---|---|
| Power Dissipation | 100 | mW |
| Collector-Emitter Voltage | 30 | V |
| Emitter-Collector Voltage | 5 | V |
| Operating Temperature Range | -40℃ to + 85℃ | |
| Storage Temperature Range | -55℃ to + 100℃ | |
| Lead Soldering Temperature [1.6mm(.063") From Body] | 260℃ for 5 Seconds | |

## ELECTRICAL / OPTICAL CHARACTERISTICS AT TA=25℃

| PARAMETER | SYMBOL | MIN. | TYP. | MAX | UNIT | TEST CONDITION | BIN NO. |
|---|---|---|---|---|---|---|---|
| Collector-Emitter Breakdown Voltage | $V_{(BR)CEO}$ | 30 | | | V | $I_C$ = 1mA<br>Ee = 0mW/㎠ | |
| Emitter-Collector Breakdown Voltage | $V_{(BR)ECO}$ | 5 | | | V | $I_E$ = 100 μA<br>Ee = 0mW/㎠ | |
| Collector Emitter Saturation Voltage | $V_{CE(SAT)}$ | | 0.1 | 0.4 | V | $I_C$ = 100 μA<br>Ee = 1mW/㎠ | |
| Rise Time | Tr | | 10 | | μs | $V_{CC}$ = 5V<br>$I_C$ = 1mA<br>$R_L$ = 1KΩ | |
| Fall Time | Tf | | 15 | | μs | | |
| Collector Dark Current | $I_{CEO}$ | | | 100 | nA | $V_{CE}$ = 10V<br>Ee = 0mW/㎠ | |
| On State Collector Current | $I_{C(ON)}$ | 0.64 | | 1.68 | mA | $V_{CE}$ = 5V<br>Ee = 1mW/㎠<br>λ =940nm | BIN A |
| | | 1.12 | | 2.16 | | | BIN B |
| | | 1.44 | | 2.64 | | | BIN C |
| | | 1.76 | | 3.12 | | | BIN D |
| | | 2.08 | | 3.60 | | | BIN E |
| | | 2.40 | | | | | BIN F |

## 7. PIC18F4620 ADC Module

# PIC18F2525/2620/4525/4620

**REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-q(1) | R/W-q(1) | R/W-q(1) |
|---|---|---|---|---|---|---|---|
| — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

**Legend:**
R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0'
-n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

| PCFG3: PCFG0 | AN12 | AN11 | AN10 | AN9 | AN8 | AN7(2) | AN6(2) | AN5(2) | AN4 | AN3 | AN2 | AN1 | AN0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000(1) | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0001 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0010 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0011 | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0100 | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0101 | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0110 | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0111(1) | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 1000 | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 1001 | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 1010 | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 1011 | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 1100 | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 1101 | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 1110 | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 1111 | D | D | D | D | D | D | D | D | D | D | D | D | D |

A = Analog input D = Digital I/O

## REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | — | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 |
| bit 7 | | | | | | | bit 0 |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **Unimplemented:** Read as ‘0’

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits
111 = 20 $T_{AD}$
110 = 16 $T_{AD}$
101 = 12 $T_{AD}$
100 = 8 $T_{AD}$
011 = 6 $T_{AD}$
010 = 4 $T_{AD}$
001 = 2 $T_{AD}$
000 = 0 $T_{AD}$(1)

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits
111 = $F_{RC}$ (clock derived from A/D RC oscillator)(1)
110 = $F_{OSC}/64$
101 = $F_{OSC}/16$
100 = $F_{OSC}/4$
011 = $F_{RC}$ (clock derived from A/D RC oscillator)(1)
010 = $F_{OSC}/32$
001 = $F_{OSC}/8$
000 = $F_{OSC}/2$

**Note 1:** If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

# Appendix F: Full Code with Comments

## PIC:

```c
/*
 * File:   main.c
 * Author: jamesjin
 *
 * Created on February 5, 2018, 12:43 PM
 */

#include <xc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "configBits.h"
#include "lcd.h"

unsigned short readADC(char channel);
void disp_Compartment(void);

const char keys[] = "123A456B789C*0#D";
const int move_keys[4] = {5, 8, 9, 10};

int status = 1, dispStat = 0;
int summary_data_avail = 0;
int nxt_msg = 0; // Counter that indicates which message should be displayed on the screen
int cursor_pos = 0;

int presc[3] = {0}; // No. of each pill per dosage - 0-R; 1-F; 2-L
int dispCounters[3] = {0}; // No. of each pill dispensed into a single compartment
int remainingPills[3] = {0}; // No. of each pill remaining at the end

int dTime = 0; // Dosage time; 1-Morning, 2-Both; 3-Afternoon
int dReps = 0; // Dosage repetition; 1-Alt.S, 2-Everyday, 3-Alt.M
int boxOrientation; // Data from color sensor (default: blue faces the user/front of machine)

int conveyorDCstat = 0, prev_conveyorDCstat = 0; // Used to poll when the conveyor gets to next stop
int dem_status = 0; // Variable for demonstration purposes only. Set to 0 to print to LCD.
double run_time = 0; // Operation time since user presses "proceed"; in seconds
int min, sec;
int timer = 0; // Variable used to time the amount of time since the last pill drop - determines whether a cylinder should stop spinning

unsigned short reading, R_threshold = 0x3f0, F_threshold = 0x1c3, L_threshold = 0x1d0;

void printLCD(const char *s1, const char *s2)
{
    __lcd_clear();
```

```c
    __lcd_clear();
    __lcd_home();
    printf("%s", s1);
    __lcd_newline();
    printf("%s", s2);
    return;
}

oid printSc2(void)
{
    __lcd_clear();
    __lcd_home();
    printf("R:%d F:%d L:%d ", presc[0], presc[1], presc[2]);
    __lcd_newline();
    printf("Home     OK");
    return;
}

nt findValue(int val, const int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == val)
            return 1;
    }
    return 0;
}

oid main(void)
{
    TRISA = 0x00;
    TRISB = 0xFF;
    TRISC = 0x00;
    TRISD = 0x00;
    TRISE = 0xFF;

    ADCON0 = 0x00;  // Disable ADC
    ADCON1 = 0b0111; // Enabled analog pins: RA 5,3-0; RE 2-0
    ADCON2bits.ADCS = 0b010; // A/D conversion time: 40000000/32 (make sure the clock is not too fast)
    ADCON2bits.ADFM = 1; // Right justify A/D result

    // Global/Peripheral Interrupts Enable
    INTCONbits.GIE = 1; // Global Interrupt Enable
    INTCONbits.PEIE = 1; // Peripheral Interrupt Enable

```

```
    // External Interrupts Configure (Enable later!!!)
    INTCON2bits.INTEDG0 = 1; // Interrupt on positive edge
    INTCON2bits.INTEDG1 = 1; // Interrupt on positive edge
    INTCON3bits.INT1IE = 1; // Do this to allow for keypad interrupts
    INTCON2bits.INTEDG2 = 1; // Interrupt on positive edge

    // Timer Config/Interrupt Enable
    INTCONbits.TMR0IE = 1; // Enable Timer0 Interrupt (for controlling information display + contact sensors)
    T0CONbits.T08BIT = 0; // Use a 16-bit timer
    T0CONbits.T0CS = 0; // Set to Timer Mode
    T0CONbits.PSA = 0; // Select to use pre-scaler
    T0CONbits.T0PS2 = 1;
    T0CONbits.T0PS1 = 1;
    T0CONbits.T0PS0 = 1; // Select pre-scaler value of 256 (for message display)
    T0CONbits.TMR0ON = 1; // Turn on Timer0 to allow message display sequence. Other times are off atm.

    PIE1bits.TMR1IE = 1; // Enable Timer1 Interrupt (for Timing; turn on timer later)
    T1CONbits.T1RUN = 0; // Select other clock source (not T1 internal clock)
    T1CONbits.TMR1CS = 0; // Select internal clock source
    T1CONbits.T1CKPS1 = 0;
    T1CONbits.T1CKPS0 = 0; // Select pre-scaler value of 1 (for accurate timing)X$

    LATCbits.LATC2 = 1; // Make sure "pinball" servo is oriented so that it will deposit into morning side
    LATCbits.LATC0 = 1; // Make sure "base" servo is in closed position
    LATCbits.LATC7 = 0;
    LATCbits.LATC5 = 0; // Stall the conveyor DC motor
    LATAbits.LATA5 = 0;
    LATAbits.LATA4 = 0; // Stall the R. dispenser motor
    LATAbits.LATA3 = 0;
    LATAbits.LATA2 = 0; // Stall the F. dispenser motor
    LATAbits.LATA1 = 0;
    LATAbits.LATA0 = 0; // Stall the L. dispenser motor

    initLCD();

    while (1)
    {
        if (status == 6)
        {
            // Make the conveyor DC motor run to the first stop
            //LATCbits.LATC7 = 1;
            printLCD("Conveyor motor", "running...");
            while (1)
            {
```

```
            {
                prev_conveyorDCstat = conveyorDCstat;
                LATCbits.LATC7 = 1;
                __delay_ms(22);
                LATCbits.LATC7 = 0;
                __delay_ms(3);
                LATCbits.LATC7 = 1;
                __delay_ms(22);
                LATCbits.LATC7 = 0;
                __delay_ms(3);
                conveyorDCstat = PORTBbits.RB7;
                if (conveyorDCstat != prev_conveyorDCstat && conveyorDCstat == 1)
                {
                    LATCbits.LATC7 = 0;
                    printLCD("Conveyor first", "stop reached");
                    break; // Motor arrives at first stop
                }
            }
            __delay_ms(1000);

            /*
            // Read the box orientation only if the pattern will be affected (i.e. only morning / only afternoon)
            if (dTime == 1 || dTime == 3)
            {
                __delay_ms(2000);
                boxOrientation = PORTBbits.RB6;
                if (boxOrientation == 1) // If blue is facing towards the back
                {
                    if (dTime == 1) { dTime = 3; }
                    else if (dTime == 3) { dTime = 1; }
                }
            }
             */

            if (dTime == 3)
            {
                LATCbits.LATC2 = 0; // Negedge = switch pinball servo to deposit into afternoon side
            }

            // Loop to repeat the process of dispense -> move -> close lid 7 times
            for (int i = 0; i < 7; i++)
            {
                // Dispense into a single compartment
                // Note: the interrupt enables/disables are not needed in actual code
                if ((dReps == 1 & i % 2 == 0) || dReps == 2 || (dReps == 3 && i % 2 == 1))
```

```
    if ((dReps == 1 & i % 2 == 0) || dReps == 2 || (dReps == 3 && i % 2 == 1))
    {
        disp_Compartment(); // Dispense pills into the morning compartment for that day
        if (dTime == 2)
        {
            LATCbits.LATC2 = 0; // Negedge = switch pinball servo to deposit into afternoon side
            __delay_ms(2000);
            disp_Compartment(); // Dispense pills into the afternoon compartment for that day
            __delay_ms(2000);
            LATCbits.LATC2 = 1; // Posedge - reset pinball servo into morning position
            __delay_ms(500);
        }
    }

    __delay_ms(1000);

    // Run conveyor until next stop
    //LATCbits.LATC7 = 1;
    if (i != 6)
    {
        printLCD("Conveyor motor", "running...");
        while (1)
        {
            prev_conveyorDCstat = conveyorDCstat;
            LATCbits.LATC7 = 1;
            __delay_ms(22);
            LATCbits.LATC7 = 0;
            __delay_ms(3);
            LATCbits.LATC7 = 1;
            __delay_ms(22);
            LATCbits.LATC7 = 0;
            __delay_ms(3);
            conveyorDCstat = PORTBbits.RB7;
            if (conveyorDCstat != prev_conveyorDCstat && conveyorDCstat == 1)
            {
                LATCbits.LATC7 = 0;
                //if (i == 6)1 { printLCD("Conveyor last", "stop reached"); }
                printLCD("Conveyor next", "stop reached");
                dem_status = 0;
                break; // Motor arrives at next/last stop
            }
        }
    }
}

```

```
}

LATCbits.LATC7 = 1;
__delay_ms(3500);
LATCbits.LATC7 = 0;
printLCD("Conveyor last", "stop reached");
__delay_ms(2000);



// *** CODE FOR RUNNING THE DISPENSERS DRY AT THE END ***
LATCbits.LATC0 = 0; // Negedge on this pin to open the base servo
__delay_ms(2000);
status = 7; // stat = 7: start using the 'timer' variable
while (1) // Run the ROUND dispenser
{
    LATAbits.LATA5 = 1;
    __delay_ms(3);
    LATAbits.LATA5 = 0;
    reading = readADC(5);
    if (reading < R_threshold)
    {
        timer = 0;
        remainingPills[0]++;
        __delay_ms(200);
    }
    if (timer >= 5) // if no pill has been detected for 5 seconds
    {
        timer = 0;
        LATAbits.LATA5 = 0;
        break;
    }
    __delay_ms(22);
}

while (1) // Run the FLAT dispenser
{
    LATAbits.LATA3 = 1;
    __delay_ms(3);
    LATAbits.LATA3 = 0;
    reading = readADC(6);
    if (reading < F_threshold)
    {
        timer = 0;
        remainingPills[1]++;
```

```
                    timer = 0;
                    remainingPills[1]++;
                    __delay_ms(200);
                }
                if (timer >= 5) // if no pill has been detected for 5 seconds
                {
                    timer = 0;
                    LATAbits.LATA3 = 0;
                    break;
                }
                __delay_ms(22);
            }

            while (1) // Run the LONG dispenser
            {
                LATAbits.LATA1 = 1;
                __delay_ms(3);
                LATAbits.LATA1 = 0;
                reading = readADC(7);
                if (reading < L_threshold)
                {
                    timer = 0;
                    remainingPills[2]++;
                    __delay_ms(200);
                }
                if (timer >= 10) // if no pill has been detected for 5 seconds
                {
                    timer = 0;
                    LATAbits.LATA1 = 0;
                    break;
                }
                __delay_ms(22);
            }

            // *** RESETTING THE CONVEYOR ***
            LATCbits.LATC7 = 1;
            __delay_ms(4000);
            LATCbits.LATC7 = 0;

            // Add code for "running the dispensers dry" here
            summary_data_avail = 1;
            if (dTime == 3)
            {
                LATCbits.LATC2 = 1; // posedge - reset pinball servo to default (morning) position
            }
            if (boxOrientation == 0 && (dTime == 1 || dTime == 3)) // Ensure that summary contains correct info
            {
                if (dTime == 1) { dTime = 3; }
                else if (dTime == 3) { dTime = 1; }
            }
            min = floor(run_time/60);
            sec = floor(run_time - 60*min);
            T1CONbits.TMR1ON = 0; // Turn off Timer1 to stop timing the machine
            INTCON3bits.INT1IE = 1; // Re-enable keypad interrupt
            T0CONbits.T0PS2 = 1;
            T0CONbits.T0PS1 = 1;
            T0CONbits.T0PS0 = 1; // Timer0: Switch to 256 prescaler to display summary data
            T0CONbits.TMR0ON = 1; // Turn on Timer0 to display information
            status = 1;
        }
    }
    return;
}

void interrupt interruptHandler(void)
{
    if (INT1IE && INT1IF)
    {
        int key_pressed = PORTB >> 4;
        if (status == 1)
        {
            T0CONbits.T0PS2 = 0;
            T0CONbits.T0PS1 = 1;
            T0CONbits.T0PS0 = 0; // Switch to using a prescaler of 8 for Timer0
            T0CONbits.TMR0ON = 0; // Turn off Timer0 (needed for dispenser contact sensors to work)
            nxt_msg = 0; // Reset the message display sequence
            summary_data_avail = 0; // New round has started; no data available yet
            run_time = 0; // Reset machine timer
            presc[0] = 0; presc[1] = 0; presc[2] = 0;
            dTime = 0;
            dReps = 0;
            printLCD("Please enter", "Prescription...");
            __delay_ms(1500);
            printSc2();
            lcd_set_cursor(2, 0);
            cursor_pos = 1;
            // Reset box orientation here as well
            status = 2;
        }
        else if (status == 2) //Remember to flip to RUN mode!!!
```

```
        else if (status == 2) //Remember to flip to RUN mode!!!
        {
            switch (cursor_pos)
            {
                case 1:
                    if (findValue(key_pressed, move_keys, 4))
                    {
                        if (key_pressed == 10) { lcd_set_cursor(6, 0); cursor_pos = 2; }
                        else if (key_pressed == 9) { lcd_set_cursor(0, 1); cursor_pos = 4; }
                    }
                    else if (key_pressed == 0 || key_pressed == 1 || key_pressed == 13)
                    {
                        if (key_pressed == 13) { key_pressed = -1; }
                        if (presc[1] + presc[2] + key_pressed >= 4)
                        {
                            printLCD("Maximum 4 Pills", "Per Dosage.");
                            __delay_ms(1500);
                            printSc2();
                        }
                        else
                        {
                            presc[0] = key_pressed + 1;
                            lcd_set_cursor(0, 0);
                            printf("R:%d F:%d L:%d ", presc[0], presc[1], presc[2]);
                        }
                        lcd_set_cursor(2, 0);
                    }
                    else
                    {
                        printLCD("Maximum 2 Round", "Pills.");
                        __delay_ms(1500);
                        printSc2();
                        lcd_set_cursor(2, 0);
                    }
                    break;
                case 2:
                    if (findValue(key_pressed, move_keys, 4))
                    {
                        if (key_pressed == 10) { lcd_set_cursor(10, 0); cursor_pos = 3; }
                        else if (key_pressed == 9) { lcd_set_cursor(8, 1); cursor_pos = 5; }
                        else if (key_pressed == 8) { lcd_set_cursor(2, 0); cursor_pos = 1; }
                    }
                    else if (key_pressed == 0 || key_pressed == 1 || key_pressed == 13)
                    {
                        if (key_pressed == 13) { key_pressed = -1; }
                        if (presc[0] + presc[2] + key_pressed >= 4)
                        {
                            printLCD("Maximum 4 Pills", "Per Dosage.");
                            __delay_ms(1500);
                            printSc2();
                        }
                        else
                        {
                            presc[1] = key_pressed + 1;
                            lcd_set_cursor(0, 0);
                            printf("R:%d F:%d L:%d ", presc[0], presc[1], presc[2]);
                        }
                        lcd_set_cursor(6, 0);
                    }
                    else
                    {
                        printLCD("Maximum 2 Flat", "Pills.");
                        __delay_ms(1500);
                        printSc2();
                        lcd_set_cursor(6, 0);
                    }
                    break;
                case 3:
                    if (findValue(key_pressed, move_keys, 4))
                    {
                        if (key_pressed == 9) { lcd_set_cursor(8, 1); cursor_pos = 5; }
                        else if (key_pressed == 8) { lcd_set_cursor(6, 0); cursor_pos = 2; }
                    }
                    else if (key_pressed == 0 || key_pressed == 1 || key_pressed == 2 || key_pressed == 13)
                    {
                        if (key_pressed == 13) { key_pressed = -1; }
                        if (presc[0] + presc[1] + key_pressed >= 4)
                        {
                            printLCD("Maximum 4 Pills", "Per Dosage.");
                            __delay_ms(1500);
                            printSc2();
                        }
                        else
                        {
                            presc[2] = key_pressed + 1;
                            lcd_set_cursor(0, 0);
                            printf("R:%d F:%d L:%d ", presc[0], presc[1], presc[2]);
                        }
                        lcd_set_cursor(10, 0);
                    }
```

```
                        lcd_set_cursor(10, 0);
                    }
                    else
                    {
                        printLCD("Maximum 3 Flat", "Pills.");
                        __delay_ms(1500);
                        printSc2();
                        lcd_set_cursor(10, 0);
                    }
                    break;
                case 4:
                    if (key_pressed == 10) { lcd_set_cursor(8, 1); cursor_pos = 5; }
                    else if (key_pressed == 5) { lcd_set_cursor(2, 0); cursor_pos = 1; }
                    else if (key_pressed == 12)
                    {
                        status = 1;
                        T0CONbits.T0PS2 = 1;
                        T0CONbits.T0PS1 = 1;
                        T0CONbits.T0PS0 = 1; // switch to 256 prescaler for Timer0
                        T0CONbits.TMR0ON = 1; // Turn on Timer0 to display messages again
                        presc[0] = 0; presc[1] = 0; presc[2] = 0;
                        dTime = 0;
                        dReps = 0;
                    }
                    break;
                case 5:
                    if (key_pressed == 5) { lcd_set_cursor(10, 0); cursor_pos = 3; }
                    else if (key_pressed == 8) { lcd_set_cursor(0, 1); cursor_pos = 4; }
                    else if (key_pressed == 12)
                    {
                        if (presc[0] == 0 && presc[1] == 0 && presc[2] == 0)
                        {
                            printLCD("No pills were", "selected.");
                            __delay_ms(1500);
                            printSc2();
                            lcd_set_cursor(8, 1);
                        }
                        else
                        {
                            printLCD("Please enter", "dosage times...");
                            __delay_ms(1500);
                            printLCD("Morning    Both", "Afternoon  Back");
                            lcd_set_cursor(0, 0);
                            cursor_pos = 1;
                            status = 3;
```


```
                            status = 3;
```


```
                        }
                    }
            }
        }
        else if (status == 3)
        {
            if (key_pressed == 12)
            {
                if (cursor_pos == 4)
                {
                    printSc2();
                    lcd_set_cursor(2, 0);
                    cursor_pos = 1;
                    status = 2;
                }
                else
                {
                    dTime = cursor_pos;
                    printLCD("Please enter", "dosage pattern..");
                    __delay_ms(1500);
                    printLCD("Alt.S   Everyday", "Alt.M   Back");
                    lcd_set_cursor(0, 0);
                    cursor_pos = 1;
                    status = 4;
                }
            }
            else
            {
                switch (cursor_pos)
                {
                    case 1:
                        if (key_pressed == 10) { lcd_set_cursor(11, 0); cursor_pos = 2; }
                        else if (key_pressed == 9) { lcd_set_cursor(0, 1); cursor_pos = 3; }
                        break;
                    case 2:
                        if (key_pressed == 8) { lcd_set_cursor(0, 0); cursor_pos = 1; }
                        else if (key_pressed == 9) { lcd_set_cursor(11, 1); cursor_pos = 4; }
                        break;
                    case 3:
                        if (key_pressed == 10) { lcd_set_cursor(11, 1); cursor_pos = 4; }
                        else if (key_pressed == 5) { lcd_set_cursor(0, 0); cursor_pos = 1; }
                        break;
                    case 4:
                        if (key_pressed == 8) { lcd_set_cursor(0, 1); cursor_pos = 3; }
```

```
                else if (key_pressed == 5) { lcd_set_cursor(11, 0); cursor_pos = 2; }
                break;
        }
    }
}
else if (status == 4)
{
    if (key_pressed == 12)
    {
        if (cursor_pos == 4)
        {
            printLCD("Morning    Both", "Afternoon  Back");
            lcd_set_cursor(0, 0);
            cursor_pos = 1;
            status = 3;
        }
        else
        {
            dReps = cursor_pos;
            printLCD("Prescription", "Summary:");
            __delay_ms(1500);

            __lcd_clear();
            __lcd_home();
            printf("Round:%d  Flat:%d", presc[0], presc[1]);
            __lcd_newline();
            printf("Long:%d", presc[2]);
            __delay_ms(1500);

            if (dTime == 1) { printLCD("Mornings Only", ""); }
            else if (dTime == 2) { printLCD("Both Mornings", "And Afternoons"); }
            else if (dTime == 3) { printLCD("Afternoons Only", ""); }
            __delay_ms(1500);

            if (dReps == 1) { printLCD("Alternate", "Starting Sunday"); }
            else if (dReps == 2) { printLCD("Everyday", ""); }
            else if (dReps == 3) { printLCD("Alternate", "Starting Monday"); }
            __delay_ms(1500);

            printLCD("Proceed?", "No     Yes");
            lcd_set_cursor(0, 1);
            cursor_pos = 1;
            status = 5;
        }
    }
    else
    {
        switch (cursor_pos)
        {
            case 1:
                if (key_pressed == 10) { lcd_set_cursor(8, 0); cursor_pos = 2; }
                else if (key_pressed == 9) { lcd_set_cursor(0, 1); cursor_pos = 3; }
                break;
            case 2:
                if (key_pressed == 8) { lcd_set_cursor(0, 0); cursor_pos = 1; }
                else if (key_pressed == 9) { lcd_set_cursor(8, 1); cursor_pos = 4; }
                break;
            case 3:
                if (key_pressed == 10) { lcd_set_cursor(8, 1); cursor_pos = 4; }
                else if (key_pressed == 5) { lcd_set_cursor(0, 0); cursor_pos = 1; }
                break;
            case 4:
                if (key_pressed == 8) { lcd_set_cursor(0, 1); cursor_pos = 3; }
                else if (key_pressed == 5) { lcd_set_cursor(8, 0); cursor_pos = 2; }
                break;
        }
    }
}
else if (status == 5)
{
    switch (cursor_pos)
    {
        case 1:
            if (key_pressed == 10) { lcd_set_cursor(7, 1); cursor_pos = 2; }
            else if (key_pressed == 12)
            {
                printLCD("Alt.S   Everyday", "Alt.M   Back");
                lcd_set_cursor(0, 0);
                cursor_pos = 1;
                status = 4;
            }
            break;
        case 2:
            if (key_pressed == 8) { lcd_set_cursor(0, 1); cursor_pos = 1; }
            else if (key_pressed == 12)
            {
                //Complete! set status to "running status" here (done)
                printLCD("Proceeding...", "");
                T1CONbits.TMR1ON = 1; // Turn on Timer1 to begin timing the operation
                __delay_ms(1000);
```

```c
                    T1CONbits.TMR1ON = 1; // Turn on Timer1 to begin timing the operation
                    __delay_ms(1000);
                    status = 6;
                }
                break;
            }
        }
        /*
        else if (status == 6)
        {
            T0CONbits.TMR0ON = 1; // Start Timer0
        }
         */
        INT1IF = 0;
    }
    /*
    else if (INTCONbits.INT0IE && INTCONbits.INT0IF)
    {
        T0CONbits.TMR0ON = 1; // Start Timer0
        INTCONbits.INT0IF = 0;
    }
    else if (INTCON3bits.INT2IE && INTCON3bits.INT2IF)
    {
        T0CONbits.TMR0ON = 1; // Start Timer0
        INTCON3bits.INT2IF = 0;
    }
     */

    // Timer Interrupts
    else if (INTCONbits.TMR0IE && INTCONbits.TMR0IF) // Timer0 Interrupt
    {
        if (status == 1)
        {
            switch (nxt_msg)
            {
                case 0:
                    printLCD("Press any key to", "begin...");
                    break;
                case 1:
                    printLCD("Operation", "Summary:");
                    break;
                case 2:
                    if (summary_data_avail == 1)
                    {
                        __lcd_clear();
```

```c
                    if (summary_data_avail == 1)
                    {
                        __lcd_clear();
                        __lcd_home();
                        printf("Round:%d  Flat:%d", presc[0], presc[1]);
                        __lcd_newline();
                        printf("Long:%d", presc[2]);
                    }
                    else printLCD("None", "");
                    break;
                case 3:
                    if (dTime == 1) { printLCD("Mornings Only", ""); }
                    else if (dTime == 2) { printLCD("Both Mornings", "And Afternoons"); }
                    else if (dTime == 3) { printLCD("Afternoons Only", ""); }
                    break;
                case 4:
                    if (dReps == 1) { printLCD("Alternate", "Starting Sunday"); }
                    else if (dReps == 2) { printLCD("Everyday", ""); }
                    else if (dReps == 3) { printLCD("Alternate", "Starting Monday"); }
                    break;
                case 5:
                    __lcd_clear();
                    __lcd_home();
                    printf("Operation time:");
                    __lcd_newline();
                    printf("%dmin %dsec", min, sec);
                    break;
                case 6:
                    __lcd_clear();
                    __lcd_home();
                    printf("Leftover pills:");
                    break;
                case 7:
                    __lcd_clear();
                    __lcd_home();
                    printf("R:%d  F:%d", remainingPills[0], remainingPills[1]);
                    __lcd_newline();
                    printf("L:%d", remainingPills[2]);
            }
            if ((summary_data_avail == 1 && nxt_msg == 7) || (summary_data_avail == 0 && nxt_msg == 2))
            {
                nxt_msg = 0;
            }
            else nxt_msg++;
        }
```

```
        }

        INTCONbits.TMR0IF = 0;
    }
    else if (PIE1bits.TMR1IE && PIR1bits.TMR1IF) // Timer1 Interrupt
    {
        run_time += 0.0065536;
        if (status == 7) timer += 0.0065536; // If we are currently running the dispensers dry
        PIR1bits.TMR1IF = 0;
    }




nsigned short readADC(char channel)

    /* Reads the analog input from the specified analog channel.
     *
     * Arguments: channel, the byte corresponding to the channel to read
     *
     * Returns: the 10-bit value corresponding to the voltage read from
     *          the specified channel
     */

    ADCON0 = (channel & 0x0F) << 2; // Select ADC channel (i.e. pin)
    ADON = 1; // Enable module
    ADCON0bits.GO = 1; // Initiate sampling
    while(ADCON0bits.GO_NOT_DONE){  continue;   } // Poll for acquisition completion
    return (ADRESH << 8) | ADRESL; // Return result as a 16-bit value


oid disp_Compartment(void) // Function that commands dispensing of right # of each pill into ONE COMPARTMENT

    if (presc[0] != 0) // Dispense ROUND pills, if they are part of the prescription
    {
        while (1)
        {
            /*
            reading = readADC(5);
            __lcd_clear();
            __lcd_home();
            printf("%.3x", reading);
             */



```

```


            LATAbits.LATA5 = 1;
            __delay_ms(3);
            LATAbits.LATA5 = 0;
            reading = readADC(5);
            if (reading < R_threshold)
            {
                dispCounters[0]++;
                dispStat = 0;
                if (dispCounters[0] >= presc[0])
                {
                    LATAbits.LATA5 = 0;
                    break;
                }
            }
            if (dispStat == 0)
            {
                __lcd_clear();
                __lcd_home();
                printf("Round:%d Flat:%d", dispCounters[0], dispCounters[1]);
                __lcd_newline();
                printf("Long:%d", dispCounters[2]);
                dispStat = 1;
                __delay_ms(200);
            }
            __delay_ms(22);

        }
    }
    if (presc[1] != 0) // Dispense FLAT pills, if they are part of the prescription
    {
        while (1)
        {
            /*
            reading = readADC(6);
            __lcd_clear();
            __lcd_home();
            printf("%.3x", reading);

            __delay_ms(200);
             */



```

```


                LATAbits.LATA3 = 1;
                __delay_ms(3);
                LATAbits.LATA3 = 0;
                reading = readADC(6);
                if (reading < F_threshold)
                {
                    dispCounters[1]++;
                    dispStat = 0;
                    if (dispCounters[1] >= presc[1])
                    {
                        LATAbits.LATA3 = 0;
                        break;
                    }
                }
                if (dispStat == 0)
                {
                    __lcd_clear();
                    __lcd_home();
                    printf("Round:%d Flat:%d", dispCounters[0], dispCounters[1]);
                    __lcd_newline();
                    printf("Long:%d", dispCounters[2]);
                    dispStat = 1;
                    __delay_ms(200);
                }
                __delay_ms(22);



            }
        }
        if (presc[2] != 0) // Dispense LONG pills, if they are part of the prescription
        {
            while (1)
            {
                LATAbits.LATA1 = 1;
                __delay_ms(3);
                LATAbits.LATA1 = 0;
                reading = readADC(7);
                if (reading < L_threshold)
                {
                    dispCounters[2]++;
                    dispStat = 0;
                    if (dispCounters[2] >= presc[2])
                    {
                        LATAbits.LATA1 = 0;
                        break;
                    }
                }
                if (dispStat == 0)
                {
                    __lcd_clear();
                    __lcd_home();
                    printf("Round:%d Flat:%d", dispCounters[0], dispCounters[1]);
                    __lcd_newline();
                    printf("Long:%d", dispCounters[2]);
                    dispStat = 1;
                    __delay_ms(200);
                }
                __delay_ms(22);
            }
        }
        for (int j = 0; j < 3; j++) { dispCounters[j] = 0; } // Reset # of dispensed pills for next compartment
    }
```

## Code for Arduino Nano

## Servo Control:

```
#include <Servo.h>

Servo servo_base;
```

```
Servo servo_pinball; //initialize a servo object for the connected servo

int curr_sig_3 = 0, prev_sig_3 = 0; //for base motor
int curr_sig_5 = 0, prev_sig_5 = 0; //for pinball motor
int begin_ball = 20, end_ball = 107; //These are magic numbers. Don't touch the magic numbers
int begin_base = 50, end_base = 180;
void setup()
{
  servo_base.attach(13); //d13/7 servo protocol, 3/5 for servo control, 9 for dispenser motor
PWM
  servo_pinball.attach(7);
  pinMode(3, INPUT);
  pinMode(5, INPUT);
  pinMode(9, OUTPUT);
}

void ccw(int begin_base, int end_base, Servo servo){
      for (int i = begin_base; i < end_base; i++)
      {
        servo.write(i);
        delay(10);
      }
}

void cw(int begin_base, int end_base, Servo servo){
   for (int i = end_base; i > begin_base; i--){
        servo.write(i);
        delay(10);
   }
}

void loop()
{
  digitalWrite(9, HIGH);
  prev_sig_3 = curr_sig_3;
  prev_sig_5 = curr_sig_5;
  delay(30); //Debounce/duration of high
  digitalWrite(9, LOW);
  delay(250);
```

```
  curr_sig_3 = digitalRead(3);
  curr_sig_5 = digitalRead(5);
  if (curr_sig_3 != prev_sig_3){
    if (curr_sig_3 == 1){
      ccw(begin_base, end_base, servo_base); //Checking inputs
    }
    else{
     cw(begin_base, end_base, servo_base);
    }
  }
  if (curr_sig_5 != prev_sig_5){
    if (curr_sig_5 == 1){
      ccw(begin_ball, end_ball, servo_pinball);
    }
    else{
     cw(begin_ball, end_ball, servo_pinball);
    }
  }
}
```

**TCS34725 Color Sensor:**

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"

Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_4X);

void setup(void) {
  Serial.begin(9600);

  if (tcs.begin()) {
    Serial.println("Found sensor");
  } else {
    Serial.println("No TCS34725 found ... check your connections");
    while (1);
  }

  pinMode(12, OUTPUT);

  // Now we're ready to get readings!
}

void loop(void) {
  uint16_t r, g, b, c;
  int avR = 0, avB = 0 /*, colorTemp, lux*/;


  for (int i = 0; i < 5; i++)
  {
    tcs.getRawData(&r, &g, &b, &c);
    avR = ((avR*i)+r)/(i+1);
    avB = ((avB*i)+b)/(i+1);
  }

  if (avR > 3000 && avB > 3000 && avB - avR > 2000)
  {
    digitalWrite(12, HIGH);
  }
  else {
    digitalWrite(12, LOW);
  }

  int pinState = digitalRead(12);
  Serial.print(pinState); Serial.print(" ");
  Serial.print("avR: "); Serial.print(avR, DEC); Serial.print(" ");
  Serial.print("avB: "); Serial.print(avB, DEC); Serial.print(" ");
  Serial.print("Diff: "); Serial.print((avB - avR), DEC); Serial.print(" ");
  Serial.println(" ");
}
```