# TRAIL MIX DISPENSER

By

Mathew Jacob

Andrew Ma

Kanav Kariya

Final Report for ECE 445, Senior Design, Spring 2018

TA: Dongwei Shi

2 May 2018

Project No. 67

## Abstract

This report recapitulates the design, development, prototype, and integration of both the hardware and software to build the Trail Mix Dispenser. The purpose of this machine is to automatically create a custom trail mix that users could take on-the-go to fulfill their nutritional requirements. The fully automated and Bluetooth enabled machine that can receive commands and recipes from an Android device, autonomously dispensing four unique ingredients, weighing each individual ingredient, and mixing the final trail mix together. The machine designed was found to be electrically sufficient, however it suffered from a combination of mechanical issues.

# Contents

# 1. Introduction

## 1.1. Objective

In today's world, an increasing number of people are living fast-paced lifestyle where their time is very valuable. A common example is the college student who has to constantly jump between schoolwork, extracurricular activities, and personal endeavors. These lifestyles leave little time for thinking about, preparing, and maintaining healthy dietary habits. For example, almost half of all college students in Kwangju, South Korea ate out at least once a day [1]. In addition, high levels of stress have been linked to increased eating of sweet and fatty foods [2]. These facts indicate that high stress lifestyles can promote poor eating habits. These eating habits can negatively affect an individual as diet is known to have a demonstrable impact on mental and physical health.

Trail mix, a snack that is highly customizable and can be high in protein, fats, and calories, can be a nutritious addition to a fast-paced and on-the-go diet if prepared correctly. A common problem with home preparation of food is incorrectly estimating the mass, and therefore calories, of ingredients. Many people that are invested in healthy lifestyles use food scales. The use of these scales before meal preparation has been proven to have a positive impact on the outcome of diet plans [3]. Providing people with low time investment, healthier, and customizable food options in their own home would increase their quality of life. In conclusion, a customized and accurately measured trail mix could be a great snack alternative for many people.

The machine covered in this report is an automated and Bluetooth enabled trail mix dispenser. The purpose is to allow a user to remotely create a quick and healthy custom snack that they can take on-the-go. Rather than taking time out of their day to measure and mix trail mix ingredients, the user will be able to remotely, through an Android application and Bluetooth, begin the dispensing of a custom trail mix.

This report will cover the high-level functionality, subsystems and interconnects, equations and simulations, design alternatives, design justifications, costs, schedules, requirements, and verification of the trail mix dispenser. In conclusion, the design decisions made created an electrically functioning prototype that suffers from a few critical mechanical failures.

## 1.2. Review of High-Level Requirements

1. Receive request from Android application to HC-05 Bluetooth module and begin dispensing. (5 points)
2. Provide appropriate levels of power to all circuit elements. (5 points)
3. Weight sensor readings have an accuracy of 15g through the microcontroller. (10 points)
4. Closed loop feedback implemented to dispense individual ingredients with an accuracy of 25g. (15 points)
5. Ability to operate all stepper and DC motors in series. (15 points)

# 2. Design

## 2.1. Block Diagram

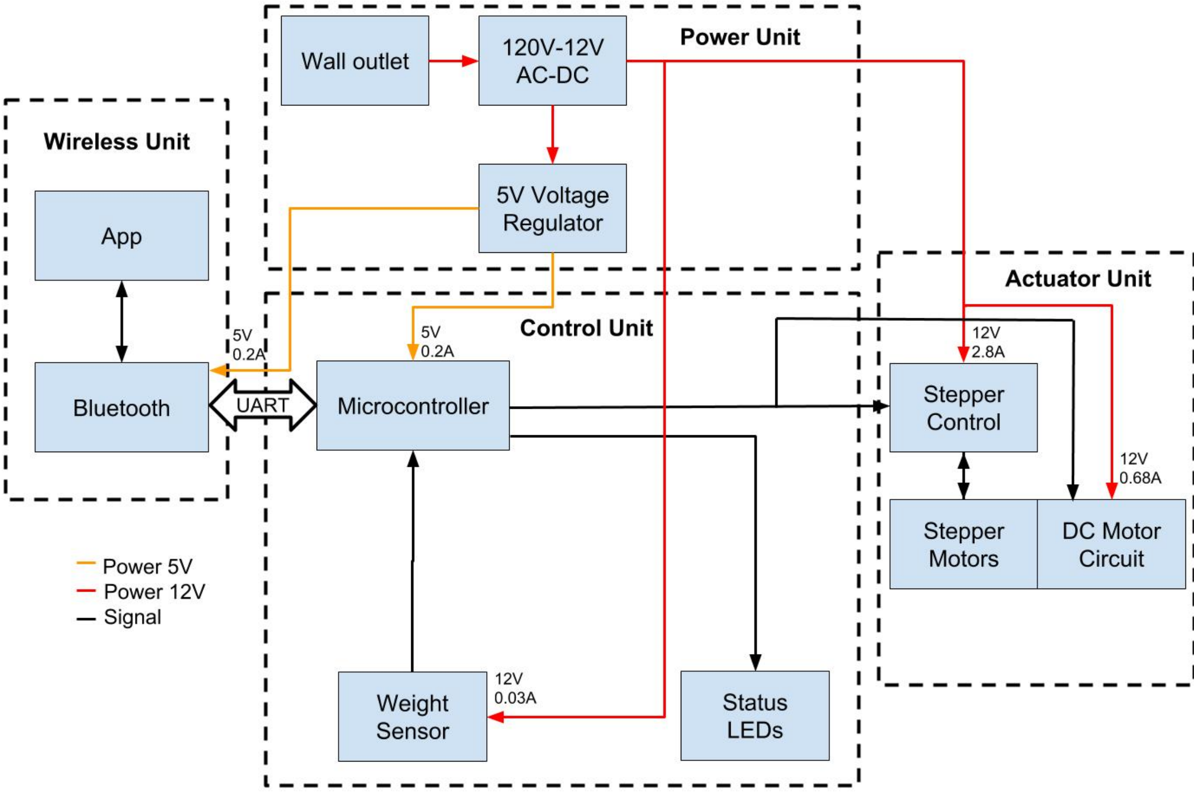


Figure 1 Block diagram for trail mix dispenser

### 2.1.1. Power Unit

The power source for this design is the wall outlet, which is transformed using a 120 V to 12 V AC-DC wall adapter. This 12 V source provides power to the stepper control, dc motor, weight sensor, and 5 V linear voltage regulator. The 5 V source provides power to the microcontroller and Bluetooth module. The Bluetooth module has a built in 3.3 V linear voltage regulator.

### 2.1.2. Control Unit

The microcontroller in this design is an Atmega328p chip programmed using the Arduino IDE. The microcontroller functions as the brain of the system, providing commands to the stepper control, dc motor circuit, and status light emitting diodes (LEDs) using inputs from the Bluetooth module using universal asynchronous receiver-transmitter (UART) and from the weight sensor. The status LEDs provide visual user feedback aside from the Android application. The weight sensor is used to weigh the amount of trail mix that has been dispensed and inform the microcontroller.

### 2.1.3. Wireless Unit

The purpose of this unit is to provide the user with an interface to the system. The Android application allows the user to connect to the Bluetooth module and send commands to dispense the desired mass of each of the four ingredients. The Bluetooth module will then receive these commands and relay them to the microcontroller using UART.

### 2.1.4. Actuator Unit

The purpose of this module is to dispense and mix the ingredients using motors. The stepper control receives commands from the microcontroller and power from the 12 V source. This power is then redirected based on the commands from the microcontroller to the steppers. Once each of the stepper motors has serially completed dispensing the desired mass the microcontroller it will tell the DC motor circuit to begin mixing the ingredients.

## 2.2. Power Unit

### 2.2.1. 120 V to 12 V AC-DC Wall Adapter

The main power source for this system was the wall outlet, which was transformed using a 12 V DC wall adapter. This is the power source for the stepper control, DC motor circuit, and the weight sensor. 12 V fell within the recommended excitation voltage for the load cell of 10 V to 15 V and there were a variety of motors available that operated at 12 V. The adapter had a 7 A capacity, which according to Table 1 is plenty of power.

**Table 1 Current drawn by devices operating at 12 V**

| Item | Quantity | Maximum Current Drawn (A) | Total Maximum Current Drawn (A) |
|---|---|---|---|
| Stepper Motor Driver | 4 | 0.7 | 2.8 |
| DC Motor | 1 | 0.68 | 0.68 |
| Weight Sensor | 1 | 0.03 | 0.03 |
| 5 V Linear Voltage Regulator | 1 | 0.72 | 0.72 |
| Total | | | 4.23 |

### 2.2.2. 5 V Linear Voltage Regulator

The 5 V Linear Voltage Regulator is used to provide power to the microcontroller and Bluetooth module. The choice was simple as the Bluetooth module required 5 V to operate its built-in 3.3 V Linear Voltage Regulator and the Atmega328p operates very well at 5 V. The LM7805 was out choice of regulator because it is very straightforward and can support up to 1 A of current, more than the maximum total draw indicated in Table 2. Also, pictured in Figure 2 is the standard operating circuit that was used to implement the LM7805.

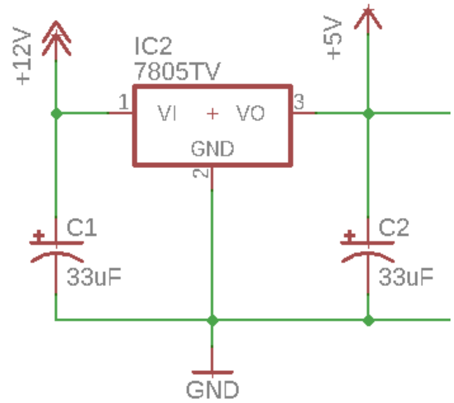


**Figure 2 Standard operating circuit for the LM7805**

**Table 2 Current drawn by devices operating at 5 V**

| Item | Quantity | Maximum Current Drawn (A) | Total Maximum Current Drawn (A) |
|---|---|---|---|
| Atmega328p | 1 | 0.5 | 0.5 |
| Decoder | 1 | 0.02 | 0.02 |
| HC-05 Bluetooth Module | 1 | 0.2 | 0.2 |
| Total | | | 0.72 |

## 2.3. Control Unit

### 2.3.1. Microcontroller

The microcontroller used in this design was the Atmega328p because it can be programmed using the Arduino IDE in the same manner as the SparkFun RedBoard development board that was used for testing. This meant that all the code used in testing could be ran without modification on our printed circuit board (PCB). The Atmega328p ran at 5 V and was connected to an external 16 MHz clock. The reason for the external clock is that for UART communication a stable clock is required and the internal clock of the Atmega328p had too much error.

The Atmega328p also had a 10-bit ADC that was utilized in reading from the weight sensor after amplification. The general-purpose input output (GPIO) pins were used as digital outputs to control the stepper control, status LEDs, and DC motor circuit. We also used a pulse-width modulation (PWM) enabled pin to control the speed of the motors through the stepper control.

The code for the microcontroller consists of three primary functions: dispensing, reading the weight from the load cell, and implementing a closed feedback loop using the two. The microcontroller dispenses for one sixth of a rotation, checks the weight from the load cell, and then decides whether to continue dispensing. A sample from the code to implement this functionality can be seen in Code Sample 1 in Appendix C.

### 2.3.2. Weight Sensor

The weight sensor contained a load cell whose output was fed through an instrumentation amplifier. The selected load cell had a capacity of 3 kg, zero balance of 0, rated output of 2 mV/V, and was excited using the 12 V and GND. This was fine for this system as the maximum amount of trail mix dispensed was to be 1 kg. These values also meant that the theoretical output was from 0 to 24 mV with a slope of 0.008 mV/g. During the process of testing the zero balance had shifted significantly as a result of physical damage to the load cell, however the output remained relatively linear between mass and voltage. The issues this caused will be addressed in 3.2.2. Weight Sensor.

The output of 0 to 24 mV is much too small to receive good resolution from the Atmega328p 10-bit ADC. The solution to this was an instrumentation amplifier, a very popular device used with four-wire sensors such as this load cell. The limit of the Atmega328p ADC is 5 V, so the desired amplification is defined by the equation $G = \frac{5\,V}{24\,mV} = 208.3$. Another option to further increase resolution would have been to increase the gain further such that only 1 kg would be the equivalent of 5 V after amplification, this would result in a gain defined by $G = \frac{5\,V}{24\,mV} * 3 = 625$. This was not included in the final design, as this would require overvoltage protection on the Atmega328p ADC in case a mass greater than 1 kg was applied to the load cell. The final gain that was selected was 160 because it was attainable with the passive components available and close to 208.3 without having to worry about overvoltage protection. The configuration of the instrumentation amplifier can be seen in Figure 3.

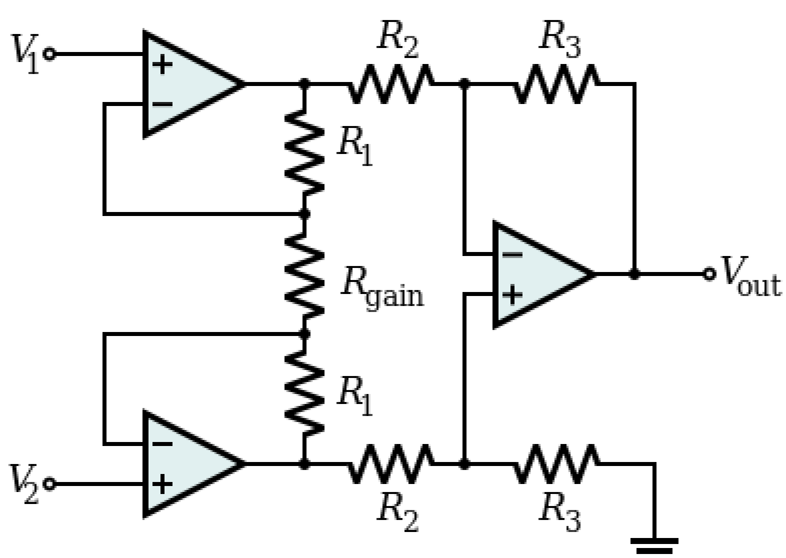


Figure 3 Standard instrumentation amplifier configuration. Source: https://en.wikipedia.org/wiki/Instrumentation_amplifier

The final step is reading the amplified load cell output through the Atmega328p 10-bit ADC. This corresponds to a resolution of 0.0049 mV/count. Theoretically our final resolution was calculated to be

3.81 g/count according to Equation 1. This was sufficient resolution to satisfy the high-level requirement of accuracy of 25 g.

Equation 1 Resolution of weight sensor readings from Atmega328p 10-bit ADC

$$\text{Resolution} = \frac{\mathbf{3}\text{ kg}}{\mathbf{160} * \mathbf{24}\text{ mV}} * \frac{\mathbf{5}\text{ V}}{\mathbf{1024}\text{ count}} = \mathbf{3.81}\text{ g/count}$$

The final physical design of the weight sensor was completed by the ECE Machine Shop and can be seen in Figure 4. Because the platform applies a mass to the load cell, the zero balance becomes negligible and from this point onward zero balance will refer to an adjusted zero balance with reference to the mass of the platform.

Figure 4 Picture of the load cell with platform built by the ECE Machine Shop.

### 2.3.4. Status LEDs

Status LEDs are a very common way for systems to give users visual feedback. There were two LEDs included in the design; a red LED was meant to signal whether the circuit was powered and controlled using the 5 V voltage regulator while the yellow LED was meant to signal whether the system was in the process of dispensing and powered and controlled using the a GPIO pin from the Atmega328p. They were each placed in series of a 1 kΩ resistor.

## 2.4. Wireless Unit

### 2.4.1. Android Application

Android was chosen as the development platform since it is a widely used mobile operating system. Its widespread use made a large market share of mobile phone and tablet users accessible. In addition, Android application development, within the scope of the project, is simple and straightforward. The app is designed to communicate the weights of each ingredient to be dispensed to the Atmega328p via Bluetooth with the assistance of the HC-05 module. The application also features a “Show Log” button that can be used for debugging. A screenshot of the application’s user interface (UI) running can be seen in Figure 5.


Figure 5 Screenshot of the Android application.

### 2.4.2. Bluetooth Module

The HC-05 module was chosen due to its ability to communicate with the Atmega328p over UART and its simplicity of operation. An HC-05 module does not require any extra libraries or components beyond a few resistors to function properly. The module ran on 5 V, however it has an internal 3.3 V linear voltage regulator. This meant that the input and output, RX and TX, pins operated at 3.3 V. Because signals coming from the Atmega328p were 5 V, a voltage divider circuit was required to provide overvoltage protection for the HC-05.

## 2.5. Actuator Unit

### 2.5.1. Stepper Control

The purpose of the stepper control design was to reduce the number of GPIO pins needed from the Atmega328P microcontroller. It takes three inputs from the Atmega328P, two select inputs and one enable input. The extra select and active-low enable inputs are attached to GND. The two select inputs select between the four motors, and the 1 enable input triggers the output to the CLK input of the stepper drivers, triggering the selected motor to rotate one full step.

The stepper drivers used were the MC3479P, which take 12 V and supply the proper currents to the phases of the bipolar motor. It triggers a full step on the rising edge of the CLK signal and is capable of half stepping as well as both clockwise and counterclockwise rotation. In our design full stepping was selected as it would provide the most torque and only a single direction of rotation was selected in all motors.

If these four motors were driven directly without the stepper control then they would take four signals from the Atmega328p, one more than is used right now. This is not a very large increase in pin efficiency, however in the case of more and more motors the stepper control becomes increasingly pin efficient. It would only take four pins to drive eight motors as opposed to eight pins to drive them directly.

### 2.5.2. Stepper Motors

Stepper motors were ideal for the project because they allowed for reasonable precision with high torque. Specifically, the Adafruit 324 was chosen because it operates at 12 V, which is the supply voltage, and had 20 N-cm of torque. This was considered sufficient torque at the time based on theoretical calculations. A lack of an empirically measured here lead to difficulties in physical implementation which will be discussed in 3.4.2. Stepper Motors.

### 2.5.3. DC Motor

A DC Motor was necessary to mix the trail mix after dispensing. The FIT0492-A was selected because it operates at 12 V, is geared for high torque, and draws less than 680 mA of current. This part of the design was to be physically mounted to a bowl and create a more homogenous trail mix.

# 3. Design Verification

## 3.1. Power Unit

### 3.1.1. 120 V to 12 V AC-DC Wall Adapter

The power supply was verified to supply 12.20 V using a multimeter, which is within the ±0.5 V margin of error. This test was performed as indicated in Table 5.

### 3.1.2. 5 V Linear Voltage Regulator

The linear voltage regulator was verified to supply 5.04V using a multimeter which is within the ±0.5 V margin of error. This test was performed as indicated in Table 5.

## 3.2. Control Unit

### 3.2.1. Microcontroller

The digital output LOW was verified to be at 0.04 V and the digital output HIGH was verified to be at 4.98 V using a multimeter. The theoretical value of the slope should be $\frac{210\ count}{5\ V} = 204.8\ count/V$. The slope, as seen in Figure 6, is 204.82 which is a negligible error. Note that the regression was performed with a fixed intercept (y = 0). These tests were performed as indicated in Table 5.

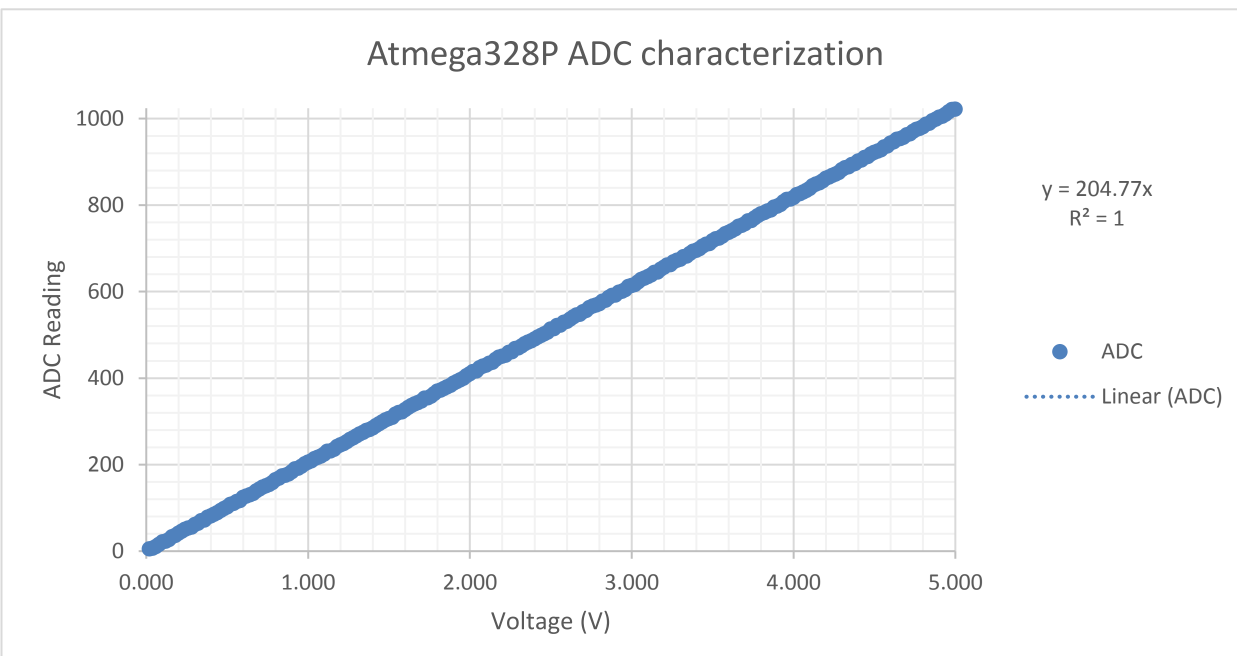


Figure 6 Atmega328P ADC Characterization

### 3.2.2. Weight Sensor

The weight sensor passed all verification tests but had additional issues (i.e. zero balance) needing to be addressed. Various incidents that involved applying excessive force on the load cell led to the zero balance gradually increasing from 0mV to 58mV

When the load cell was first received, it was excited with 12 V and was measured with a multimeter to have a zero balance of 0 mV and a peak voltage of 24 mV. Shortly after the load cell was physically damaged in transport, it was characterized using a 0.01 g precision scale and a multimeter as shown in Figure 7. The slope of 0.0078 mV/g has a 2.5% error from the theoretical slope of 0.008 mV/g which was calculated in 2.3.2. Weight Sensor. Despite zero balance shifting to about 19 mV, the response of the load cell remained linear with an $R^2$ of 0.9841.

Then, the load cell was connected to an instrumentation amplifier with a gain of 160 and characterized as shown in Figure 8, again using a 0.01 g precision scale and a multimeter. The theoretical slope was $160 * 0.0078 \text{ mV/g} = 0.001248 \text{ V/g}$. The slope in Figure 8 is 0.0011 V/g, giving a 12% error and an experimental gain of 141. This error, however, is constant considering that the $R^2$ is 0.9877.

Finally, the instrumentation amplifier was connected to a voltage divider circuit (scaled voltage by a factor of 0.403) for overvoltage protection and then to the Atmega328P. The load cell was again characterized as shown in Figure 9, this time using a 0.01 g precision scale and recording ADC readings over serial. The zero balance at this point is calculated in Equation 2 to be 57.65 mV using the initial reading in **Error! Reference source not found.** which was 671.

$$671 * \frac{1}{0.403 * 141} * \frac{5 \text{ V}}{1024 \text{ counts}} = 57.65 \text{ mV}$$

**Equation 2 Final Load Cell Zero Balance**

The zero balance more than exceeded double our initial maximum reading for the load cell, limiting our ability to increase the gain, and thus resolution, of our readings. As a result, our final resolution was about $\frac{1}{0.0987} = 10.13 \text{ g/count}$. This, alongside an ADC inaccuracy of ±1 count, provides an accuracy of approximately 20 g. This is just barely within the limit that was specified within Table 5.

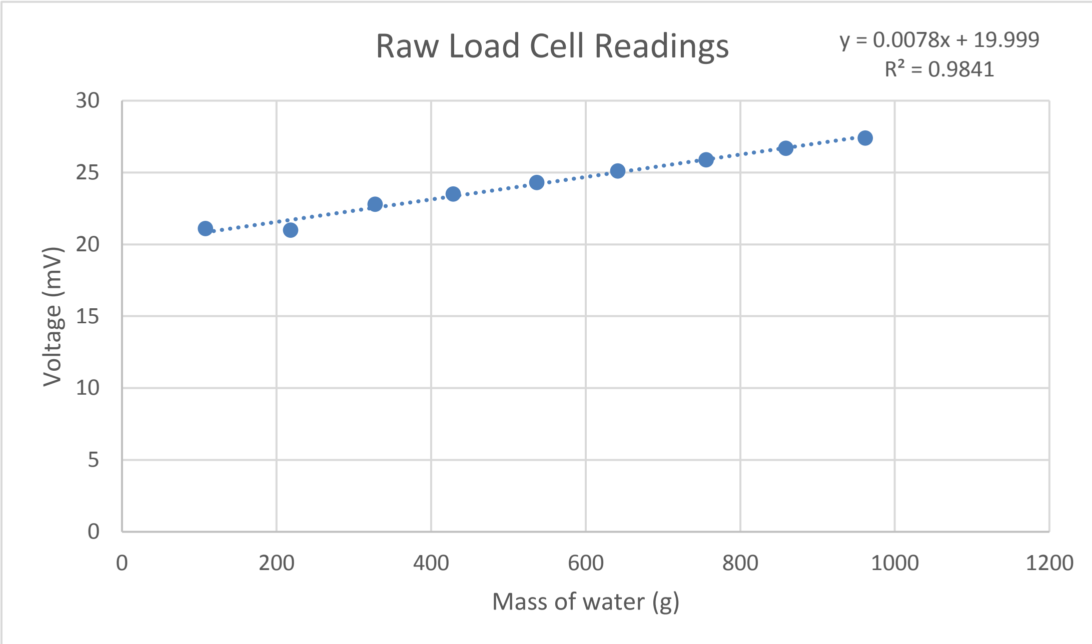


**Figure 7 Raw Load Cell Readings**

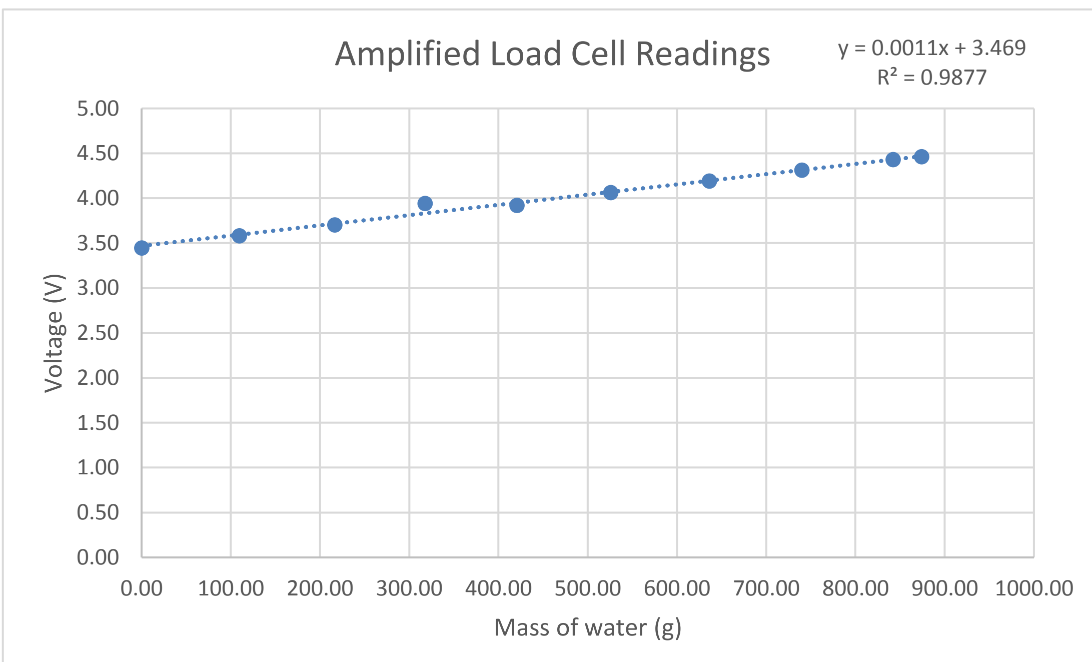


**Figure 8 Amplified Load Cell Readings**

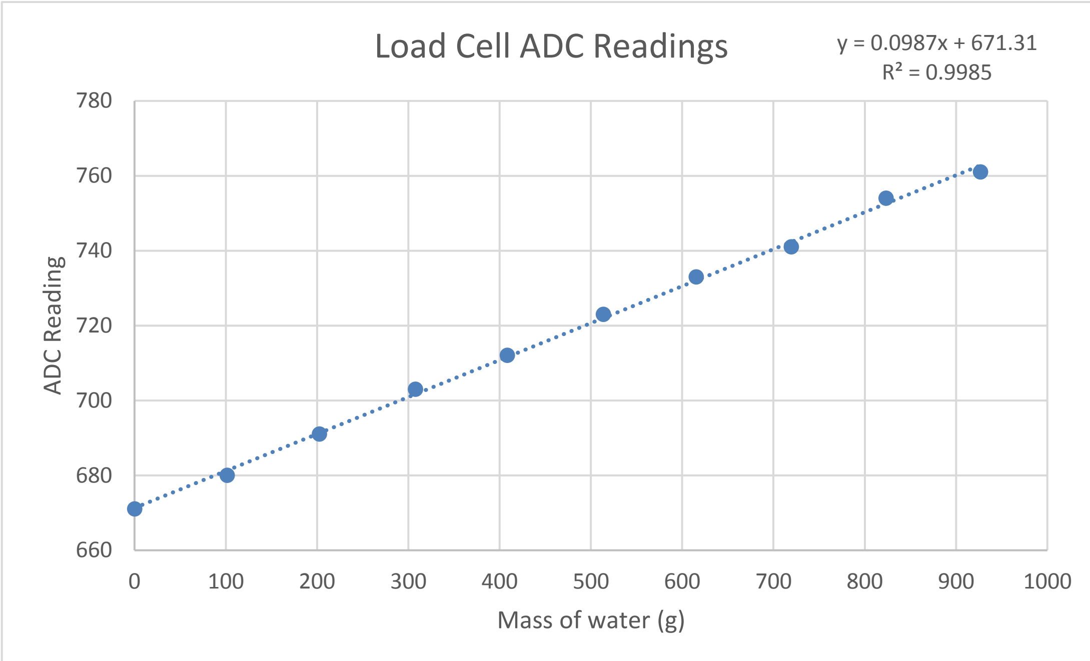


Figure 9 Load Cell ADC Readings

#### 3.2.3. Status LEDs

The status LEDs passed verification, being visible from over three meters away when assembled as specified in Table 5.

### 3.3. Wireless Unit

#### 3.3.1. Android Application

The application was able to send messages over Bluetooth to the HC-05 module from over ten meters away, far exceeding the 5 m requirement. This test was performed as indicated in Table 5.

#### 3.3.2. Bluetooth Module

The Bluetooth module connected via UART to the SparkFun RedBoard and the Atmega328P without issue. In addition, it was usable from over ten meters away, far exceeding its 5 m requirement. This test was performed as indicated in Table 5.

### 3.4. Actuator Unit

#### 3.4.1. Stepper Control

The stepper control was able to correctly control all four stepper motors, one at a time, and operate them at frequencies from 10 Hz up to 10 kHz without issue. There was an issue which arose with the stepper motors, in which the motors would provide a back electromotive force (EMF), affecting the 12 V supply voltage which was also used as the excitation voltage for the load cell. This led to a variance of about ±40 g in load cell readings while a motor was stalling, leading to very unreliable dispensing. The solution to this problem is further explained in 5.4. Future work. This test was performed as indicated in Table 5.

### 3.4.2. Stepper Motors

The steppers never drew more than 0.500 A from the power supply, so they met the power requirement. The requirement for torque, based off theoretical calculations was far insufficient to properly dispense trail mix ingredients. While the steppers were specified to meet the torque requirement on the data sheet, they often stalled while dispensing. As a result of this stalling, a back EMF was applied to the 12 V power line. The solution to this problem is further explained in 5.4. Future work. These tests were first performed as indicated in Table 5. The stalling occurred after they were coupled to the physical design and opposing force from the mass of the ingredients was applied.

### 3.4.3. DC Motor

The DC motor circuit consisted of a MOSFET with one end of the DC motor attached to the drain and the source attached to GND. The other end of the DC motor was attached to the 12 V power line. A fly back diode was placed in parallel with the DC motor to prevent a back EMF from affecting the 12 V power line. The motor operated as expected when set up as indicated in Table 5. However, there was not enough time to properly mechanically integrate the DC motor and mixing fins, as a result the final mixture was not mixed by the DC motor but by the user themselves.

# 4. Costs

## 4.1. Parts

Table 3 Summary of cost of parts used in the design.

| Part | Manufacturer | Retail Cost ($) | Bulk Purchase Cost ($) | Actual Cost ($) |
|---|---|---|---|---|
| LM358N Dual Op-Amp | TI | 0.52 | 0.2 | 0.52 |
| UA741 Op-Amp | TI | 0.58 | 0.22 | 0.58 |
| Load Cell | Seeed Technology Co., Ltd | 10.15 | 9.135 | 10.15 |
| Stepper x4 | Adafruit Industries LLC | 56 | 50.4 | 56 |
| DC Motor | DFRobot | 11.9 | 10.71 | 11.9 |
| Bipolar Stepper Driver x4 | Allegro MicroSystems, LLC | 14 | 8.76 | 14 |
| 3x8 Decoder | Toshiba Semiconductor and Storage | 0.41 | 0.113 | 0.41 |
| 12V 7A Power supply | Advantech Corp | 30.88 | 25.27 | 30.88 |
| Atmega328 | Microchip Technology | 2.01 | 1.627 | 2.01 |
| Bluetooth Module | HiLetgo | 7.69 | 2.6 | 7.69 |
| 3.3V regulator | Diodes Incorporated | 0.58 | 0.112 | 0.58 |
| 3-pack Cereal Dispensers | Zevro | 27.99 | 27.99 | 27.99 |
| 1-pack Cereal Dispenser | Zevro | 18.64 | 9.33 | 18.64 |
| DC Barrel Jack | CUI Inc. | 1.25 | 0.68 | 1.25 |
| 5V regulator | ON Semiconductor | 0.95 | 0.43125 | 0.95 |
| 4 x MC3479P | ON Semiconductor | 2.6 | 2.54 | 2.6 |
| **Total** | | 186.15 | 150.11825 | 186.15 |

Passive components such as resistors and capacitors are not included in Table 3 as they were provided by the ECE445 lab.

## 4.2. Labor

Table 4 Summary of estimated labor costs through the design process.

| Component | Cost($) |
|---|---:|
| Average hourly rate | 40.00 |
| Number of hours to complete per partner | 60.00 |
| Machine Shop Hours | 10 |
| Total Hours | 70 |
| **Total Labor Cost** | **7,000.00** |

A summary of the costs has been provided in Table 3 and Table 4. The sum of these two costs provides an estimate of the total cost of designing and prototyping the design and can be seen in Equation 3.

Equation 3 Calculation for estimated cost of designing system.

$$\textbf{Total Cost} = \$\mathbf{7000} + \$\mathbf{186.15} = \$\mathbf{7186.15}$$

# 5. Conclusion

## 5.1. Accomplishments

The major accomplishment in the design was that the entire system electrically functioned as necessary. The Android application was able to communicate to the HC-05 Bluetooth module, which was then read via UART by the microcontroller on the PCB, which can be seen in Figure 10. Once the command was received the stepper motors were each able to operate in series based on readings from the weight sensor. This was a major success for the designed stepper control circuit. Following this, the DC motor spun for a few seconds after, being completely electrically functional. However, as referenced in 2.5.3. DC Motor it was not physically incorporated.

## 5.2. Uncertainties

The major failures in the design resulted from a poorly designed requirement, specifically the stepper motor torque. As a result of lacking torque, the motors would stall when faced with enough opposing force from ingredients. This resulted in a back EMF that caused noise in the 12 V line. This fluctuation triggered a drop in the voltage that was used to excite the load cell. This caused the readings from the weight sensor to the Atmega328p to fluctuate, causing the stepper motors to not operate as intended. Solutions to this are addressed in the following section 5.4. Future work. The final physical design can be seen in Figure 11.

## 5.3. Ethical considerations

Safety concerns in the design process were relatively limited. The most dangerous unit in the design was the 120 V wall outlet power. As a result, whenever testing, a power strip with circuit protection was used between the wall outlet and the 120 V – 12 V wall adapter to allow for quickly and safely cutting power. Also, before any power was provided for the circuit, the PCB was checked for any damaged, frayed, or shorted wiring. The PCB was also placed in a position where no food from the dispensing process would come into contact, preventing any accidental damage.

Because this system is being used for food preparation, it is important to disclose any inaccuracies in the design. In accordance to Section 1 and Section 3 of the IEEE Code of Ethics, in this paper the inaccuracy in the dispensed mass is disclosed [4]. This ensures that users will use discretion before using our design as the sole source for their dietary information and decisions.

In conclusion, the design team is responsible for all design decisions, meaning that all issues that may endanger the user, both in the short and long term, must be disclosed.

## 5.4. Future work

In the short term, the road ahead would include using stepper motors with higher torque, diodes attached to the stepper motor circuit, a new load cell, and various physical adjustments.

The reason for stepper motors with higher torque is to prevent stalling on larger, heavier, or even more oddly shaped ingredients. A major issue that was encountered during the testing of this design was that the motors would not be powerful enough to dispense ingredients onto the load cell, requiring users to manually assist the stepper motor in dispensing.

Whenever the motors stalled, a small back EMF would cause a shift in the 12 V power line, which was attached to the input of our closed feedback loop, the weight sensor. Because of this fluctuation, the weight sensor readings would fluctuate significantly causing the microcontroller to receive inaccurate readings and continue to attempt dispensing improperly.

The load cell also had taken a lot of physical wear across the course of testing. As a result, the zero balance had shifted significantly. This required the use of a lower gain to ensure that 1 kg of trail mix would not saturate the amplifier. As a result, our resolution of 10.13 g/count could be increased to the theoretical 3.81 g/count.

The final short-term adjustment would be a more concise and compact physical design including, but not limited to, proper shielding for the load cell extended cable, proper mounting bracket for the PCB and the wall adapter, a better bracket for installment, physical implementation of the DC motor, and better funneling for ingredients. All of these would make the product significantly more user friendly.

In the long-term, some adjustments would include providing usage statistics and a cleaner UI for user friendliness, including a watchdog timer so the system can sleep and reduce power consumption when not in use, applying load cells under each ingredient dispenser such that the user can be notified when they are low on ingredients, and nutritional information tracking within the application so the user has more information about their dietary habits available.

## References

[1] H.-K. Ro, "Factors in Food Selection and Eating Out Behavior of College Students," *Journal of the Korean Society of Food Culture,* vol. 14, no. 3, pp. 241-249, 1999.

[2] O. G, "Stress and food choice: a laboratory study.," *Psychosomatic Medicine,* vol. 62, no. 6, pp. 853-865, 2000.

[3] L. Lipsky, "Power of Food Scale in association with weight outcomes and dieting in a nationally representative cohort of U.S. young adults," *Appetite,* vol. 105, no. 1, pp. 385-391, 2016.

[4] IEEE, "IEEE Code of Ethics," IEEE, [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 2 May 2018].

# Appendix A. Requirement and Verification Table

Table 5 Requirements and Verification

| Requirement | Verification | Verification status (Y or N) |
|---|---|---|
| **12 V AC-DC Wall Adapter**<br>1. $V_{out}$ = 12 V ± 0.5 V | **12 V AC-DC Wall Adapter**<br>1. Measure the voltage between $V_{out}$ and GND and ensure it falls within the required range<br>(a) Plug wall adapter into the wall outlet.<br>(b) Measure the voltage between $V_{out}$ and GND and ensure it falls within the required range. | 1. Y |
| **5 V Voltage Regulator**<br>1. $V_{out}$ = 5 V ± 0.3 V | **5 V Voltage Regulator**<br>1. Verification Process for Item 1<br>(a) Power the 5 V Voltage Regulator using the 12 V and GND from the lab kit.<br>(b) Measure the voltage between Vout and GND and ensure it falls within the required range. | 1. Y |
| **Microcontroller**<br>1. Digital LOW corresponds to $V_{out} \leq 1$ V<br>2. Digital HIGH corresponds to $V_{out} \geq 3.5$ V<br>3. ADC must have at least 10-bit resolution quantizing from 0 to 1023 | **Microcontroller**<br>1. Verification Process for Item 1<br>(a) Power the microcontroller with 5 V and GND from the power supply in the lab kit.<br>(b) Upload code setting all digital pins to OUTPUT and LOW<br>(c) Probe each pin using lab kit DMM ensuring the voltage falls in the required range.<br>2. Verification Process for Item 2<br>(a) Power the microcontroller with 5 V and GND from the power supply in the lab kit.<br>(b) Upload code setting all digital pins to OUTPUT and HIGH.<br>(c) Probe each pin using lab kit DMM ensuring the voltage falls in the required range.<br>3. Verification Process for Item 3<br>(a) Power the microcontroller with 5 V and GND from the power supply in the lab kit.<br>(b) Upload code to print the read value from the analog input pin | 1. Y<br>2. Y<br>3. Y |

| | | |
|---|---|---|
| | via Serial.<br>(c) Connect the analog input pin to a DC power supply and turn the DC power supply on.<br>(d) Increment the voltage by a multiple 0.02 V.<br>(e) Ensure the value printed to serial increments by the multiple.<br>(f) Repeat steps (d) and (e) until the voltage is at 5 V. | |
| **Weight Sensor**<br>1. Must have an output range within 0 to 5 V<br>2. Must be able to support 1 kg<br>3. Must have an accuracy of 25 g | 1. Verification Process for Item 1<br>(a) Power the load cell and amplifiers with 12 V and GND from the power supply in the lab kit.<br>(b) Attach the measurement leads to a DMM.<br>(c) Measure the output voltage with no load.<br>(d) Measure the output voltage with a 3 kg load.<br>(e) Ensure that the outputs fall within the required range.<br>2. Verification Process for Item 2<br>(a) Power the load cell and amplifiers with 12 V and GND from the power supply in the lab kit.<br>(b) Attach the measurement leads to a DMM.<br>(c) Measure the output voltage with 1 kg and ensure it falls below 5 V.<br>3. Verification Process for Item 3<br>(a) Power the load cell and amplifiers with 12 V and GND from the power supply in the lab kit.<br>(b) Attach the measurement leads to a DMM.<br>(c) Apply a known mass and record the measured voltage and mass.<br>(d) Continue adding mass until 1 kg has been placed on the load cell.<br>(e) Perform a linear regression to find the voltage to mass ratio.<br>(f) Ensure that all the measured | 1. Y<br>2. Y<br>3. Y |

| | | |
|---|---|---|
| | data points have an accuracy of 25 g from the regression line using the voltage to mass ratio. | |
| **Status LEDs**<br>1. LED illuminates with 5 V and a 1 kΩ resistor | 1. Verification Process for Item 1<br>(a) Connect an LED and 1 kΩ resistor in series across 5 V and GND from the power supply in the lab kit.<br>(b) Verify that the LED is visible from 3 m away with no obstruction. | 1. Y |
| **Stepper Control**<br>1. Must be able to control four stepper motors in series<br>2. Must be able to provide a STEP signal to the motor varying from 10 Hz to 1 kHz | 1. Verification Process for Item 1<br>(a) Power the stepper motor control with 12 V, 5 V, and GND from the power supply in the lab kit and connect output leads to the stepper motors.<br>(b) Provide the appropriate select bits for a motor with 5 V and GND. Code Sample 4 may be used as a reference.<br>(c) Provide a 20 Hz 5 V peak-to-peak 2.5 V offset square wave for 10 seconds to the STEP input.<br>(d) Ensure that the motor shafts rotate.<br>(e) Repeat steps 2-4 for the remaining motors.<br>2. Verification Process for Item 2<br>(a) Power the stepper motor control with 12 V, 5 V, and GND from the power supply in the lab kit and connect output leads to the stepper motors.<br>(b) Provide the appropriate select bits for a motor with 5 V and GND.<br>(c) Provide a 20 Hz 5 V peak-to-peak 2.5 V offset square wave for 10 seconds to the STEP input.<br>(d) Ensure that the motor shafts rotate.<br>(e) Repeat steps 2-4 increasing the frequency by increments of less than 100 Hz.<br>(f) Repeat steps 2-5 for the | 1. Y<br>2. Y |

| | | |
|---|---|---|
| | remaining motors. | |
| **Stepper Motors**<br>1. Draws ≤ 1 A of current when static.<br>2. Must have a torque ≥ 16 oz-in at an angular speed of 10 rpm. | 1. Verification for Item 1<br>(a) Power the stepper motor control with 12 V, 5 V, and GND from the power supply in the lab kit and connect the output leads to the motor.<br>(b) Using a multimeter, measure the current drawn to the motor drivers and ensure it is in the required range.<br>2. Verification for Item 2<br>(a) Power the stepper motor control with 12 V, 5 V, and GND from the power supply in the lab kit and connect the output leads to the motor.<br>(b) Provide the appropriate select bits for a motor with 5 V and GND.<br>(c) Provide a 2 kHz 5 V peak-to-peak 2.5 V offset square wave for 10 seconds to the STEP input.<br>(d) Measure the torque of the motor using a torque wrench | 1. Y<br>2. N |
| **DC Motor**<br>1. Must be able to control using digital output from microcontroller | 2. Verification for Item 1<br>(a) Connect the DC motor across 12 V from the power supply in the lab kit and the drain pin of a properly rated MOSFET.<br>(b) Connect the source to GND from the power supply in the lab kit. Connect the gate to a GPIO pin on the microcontroller.<br>(c) Power the microcontroller with 5 V and GND from the power supply in the lab kit.<br>(d) Upload Code Sample 3 to microcontroller.<br>(e) Measure the gate voltage using a DMM.<br>(f) Ensure that the motor is turning on and off corresponding to the gate | 1. Y |
| **Bluetooth**<br>1. Can connect to the MCU via UART | 1. Verification for Item 1<br>(a) Connect the Bluetooth module | 1. Y<br>2. Y |

| | | |
|---|---|---|
| 2. Can be used from up to 5 m away. | pin TX to the microcontroller<br>(b) Connect VCC to 5 V from the power supply in the lab kit<br>(c) Connect GND to GND from the power supply in the lab kit<br>(d) Connect RX to the microcontroller with a resistor divider circuit such that the voltage at the microcontroller is 5 V when the Bluetooth module provides 3.3 V.<br>(e) Power on the microcontroller with Code Sample 2.<br>(f) Connect to the HC-05 from a computer over Bluetooth.<br>(g) Open a serial session on the computer and type a message.<br>(h) Verify that the message was received by the microcontroller. | |
| **Application**<br>1. Communicates accurately with the Bluetooth module from up to 5 m away. | 1. Verification for Item 1<br>(a) Setup HC-05 Bluetooth module with the microcontroller using the power supply in the lab kit.<br>(b) Program microcontroller with code from Code Sample 2.<br>(c) Move 5 m away, using a measuring tape to confirm distance, and establish a Bluetooth connection between a mobile device and the module.<br>(d) Open the application and transmit orders.<br>(e) Verify that the order was echoed back via serial | 1. Y |

# Appendix B. Additional Figures

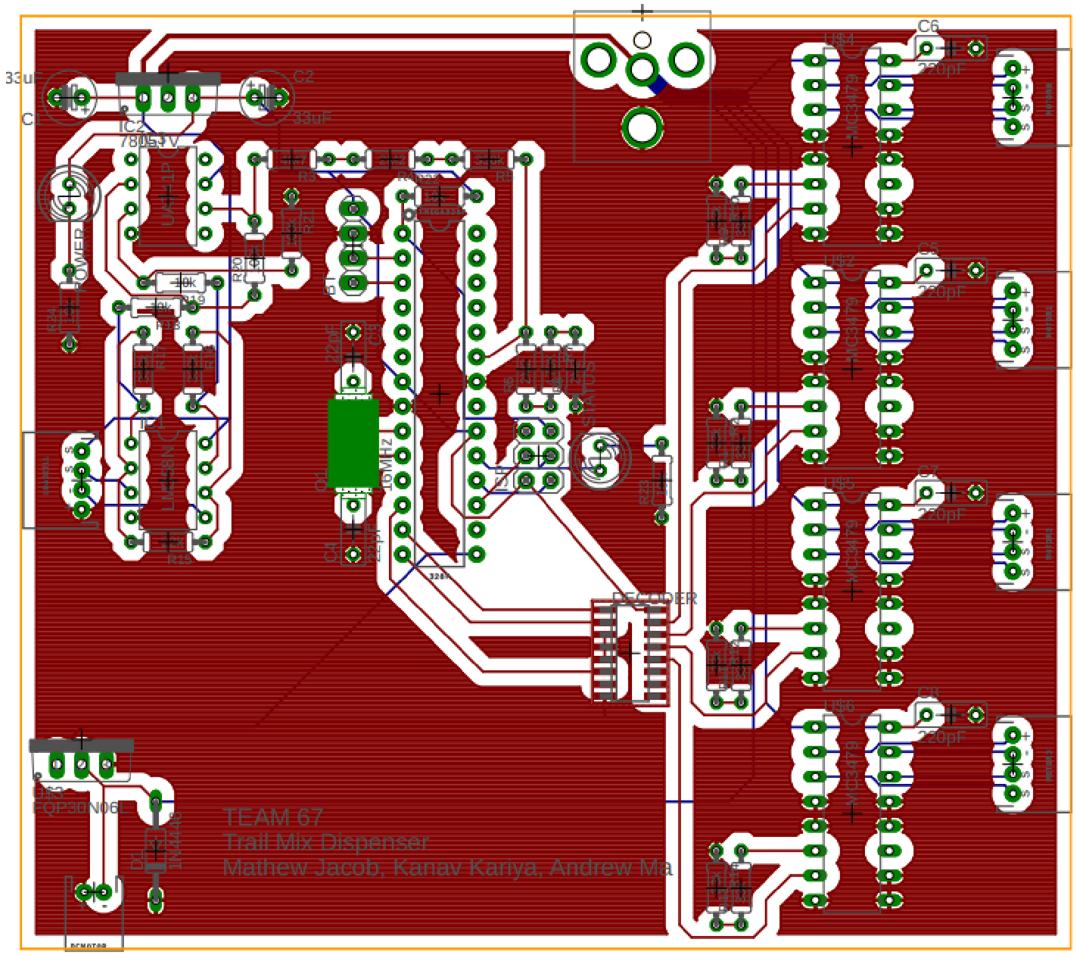


Figure 10 Image of the final PCB layout

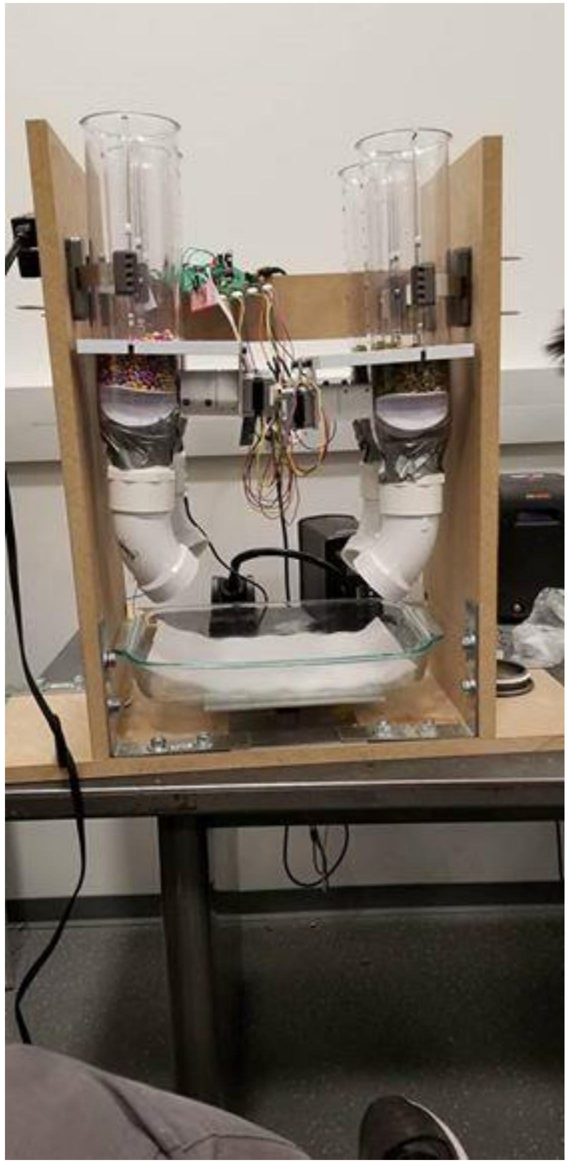

Figure 11 Picture of final physical implementation of the full design.

## Appendix C. Code Samples

```
void dispense(int stepper){
  digitalWrite(DECODER_ENABLE, LOW);
  digitalWrite(DECODER_SELECT[0], stepper & 0x01);
  digitalWrite(DECODER_SELECT[1], stepper & 0x02);

  tone(DECODER_ENABLE, 31, 500);
}

float readWeight(){
  int weightReading = analogRead(LOAD_CELL);
  int averageWeight = ema_filter(weightReading, averagePtr);
  float weight = ((float)averageWeight-baseWeightReading) / 0.0987;
  return weight;
}

void dispenseWeight(int stepper, float dispenseWeight){
  float weight = readWeight();
  Serial.println("Dispensing " + String(dispenseWeight) + " grams from 
container " + String(stepper));
  float initialWeight = weight;
  while(weight+0.5 < initialWeight + dispenseWeight){
    Serial.println("start loop" + String(weight));
    dispense(stepper);
    for(int i = 0; i < 5000; i++){
      readWeight();
      delay(1);
    }
    weight = readWeight();
    Serial.println(weight);
  }
}
```

Code Sample 1 Excerpts from Final Code

```
// test outputs
const int TEST_LED = 13;

// globals
String message;
int led_status;

// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin 13 as an output.
  Serial.begin(9600); //set baud rate
  led_status = HIGH;
  message = "";
}

// the loop function runs over and over again forever
void loop() {
  if(Serial.available())
  {//while there is data available on the serial monitor
    char c   = char(Serial.read());//store string from serial command
    message += c;
  }
  else
  {
    if(message!=""){
      if(message[message.length()-1] == '\r')
      {
        message.trim();
        Serial.println("Got " + message);
        if(led_status == HIGH){
          led_status = LOW;
        } else {
          led_status = HIGH;
        }

        digitalWrite(TEST_LED, led_status);
        msg = "";
      }
    }
  }
}
```

Code Sample 2 Bluetooth Test

```
const int TEST_LED = 13;


// outputs
const int DC_MOTOR = 9;

// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin 13 as an output.

  pinMode(DC_MOTOR, OUTPUT); //  DC Motor
  pinMode(TEST_LED, OUTPUT); // LED
}

// the loop function runs over and over again forever
void loop() {
    delay(1000);
    digitalWrite(DC_MOTOR, LOW);
    digitalWrite(TEST_LED, LOW);
    delay(1000);
    digitalWrite(DC_MOTOR, HIGH);
    digitalWrite(TEST_LED, HIGH);
}
```

**Code Sample 3 DC Motor Test**

```
// test outputs
const int TEST_LED = 13;

// outputs
const int DECODER_ENABLE = 6;
const int DECODER_SELECT[] = {7,8};

// globals

void writeStepper(int stepper, int m_speed){
  digitalWrite(DECODER_ENABLE, LOW);
  digitalWrite(DECODER_SELECT[0], stepper & 0x01);
  digitalWrite(DECODER_SELECT[1], stepper & 0x02);

  tone(DECODER_ENABLE, m_speed);
}

void disableSteppers(){
  digitalWrite(DECODER_ENABLE, LOW);
}

// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin 13 as an output.
  Serial.begin(9600); //set baud rate

  pinMode(DECODER_SELECT[0], OUTPUT); // Decoder select pins
  pinMode(DECODER_SELECT[1], OUTPUT);
  pinMode(DECODER_ENABLE, OUTPUT); // Decoder enable
  pinMode(TEST_LED, OUTPUT); // LED
  disableSteppers();
}

// the loop function runs over and over again forever
void loop() {
  for(int i = 0; i < 4; i++){
    writeStepper(i, 70);
    digitalWrite(TEST_LED, HIGH);
    delay(1000);
    disableSteppers();
    digitalWrite(TEST_LED, LOW);
    delay(1000);
  }
}
```

Code Sample 4 Stepper Motor Test