Refill Dispensary

Team 18

TA: Jason Paximadas

Professor: Arne Fliflet

29 September, 2022

ECE 445

Michael Blyakhman (mlb11)

Jackson Wiessing (jtw6)

Lyla Zegelstein (lrz2)

Contents

1 Introduction

1.1 Problem

Plastic waste is a massive issue world-wide, particularly as it pertains to the packaging of food and other household goods.The United States Environmental Protection Agency estimates that in 2018, 14 million tons of plastic was consumed for packaging in the USA with about seventy percent of that ending up in landfills[1]. Plastic waste is detrimental to the environment as it doesn't decompose naturally on human time scales. End-user plastic waste is often unnecessary as consumers own containers capable of being reused. Furthermore, consumers are often required to purchase a greater quantity of an item than they need.

We propose a unique vending machine which will dispense precise quantities of goods into reusable containers. The machine can be small or large and still provide variety to customers. Competitor's designs have each type product in the machine having its own dispenser spot. This isn't a space-efficient design. Our machine allows multiple products to share a dispensing hole which helps us to keep the width of the machine to a minimum and build it vertically. Another competitive feature we have is the ability to dispense unrelated products. For example, our competitors are building machines dedicated to dispensing one type of good in multiple brands like laundry detergent. Meanwhile, we are innovating on this concept by designing something that is able to be tailored to different areas and exist in a variety of places. One neighborhood might buy lots of quinoa and rice, but another one may demand pasta and cereal. Different types of goods should all be able to be stored in the same machine and fit the demand of an area. We expect that our machine will be able to be placed in corner stores, gas stations, and more while our competitors are limited to supermarkets due to their lack of variety and large size.

1.2 Visual Aid

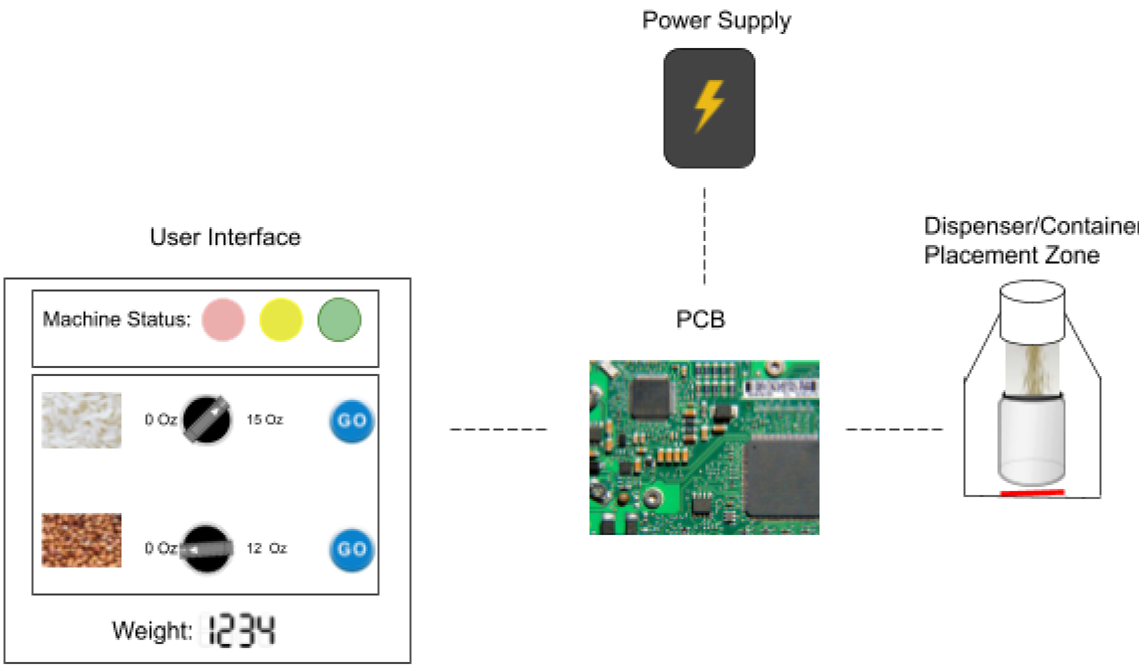

Figure 1: Mockup of Dispensing Machine

1.3 High level requirements:

1. The user is able to choose an item and quantity to get dispensed via buttons and rotary potentiometers. Before placing the order, the user will have some indication of the amount ordered within a tolerance of 10 grams for weights between .1 kg and 2kg.
2. The machine dispenses two different types of low-viscosity liquids and is able to perform 2 orders in succession.
3. The machine can deliver another order properly after 1 of the liquids is switched out or refilled.

2 Design

2.1 Block Diagram

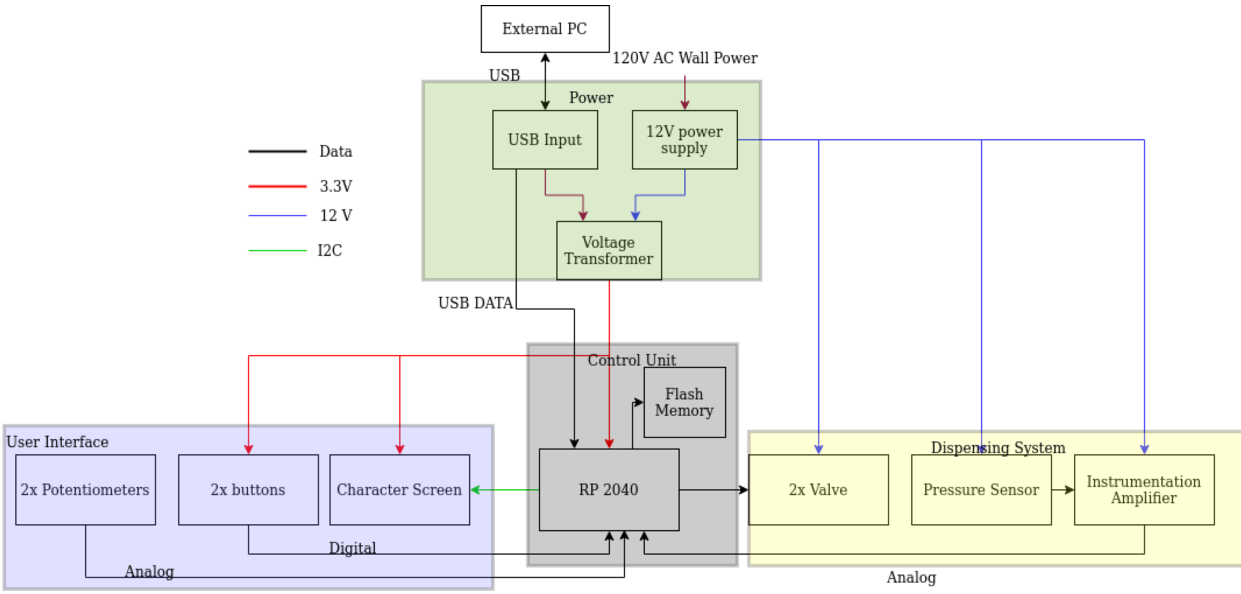

Figure 2: Block diagram for the vending machine

The block diagram has 4 main parts. The user interface interacts with the RP2040 by signaling button presses and potentiometer changes. The RP2040 then will display the number from the potentiometer onto the character screen of the user interface. The dispensing system valves are controlled by the RP2040. Additionally, the pressure sensor in the dispensing system will notify the RP2040 when to start and stop dispensing. The flash memory will be used to store program instructions and data about the containers. Finally, the power subsystem will supply all other subsystems with the proper power type.

2.2 Physical Design

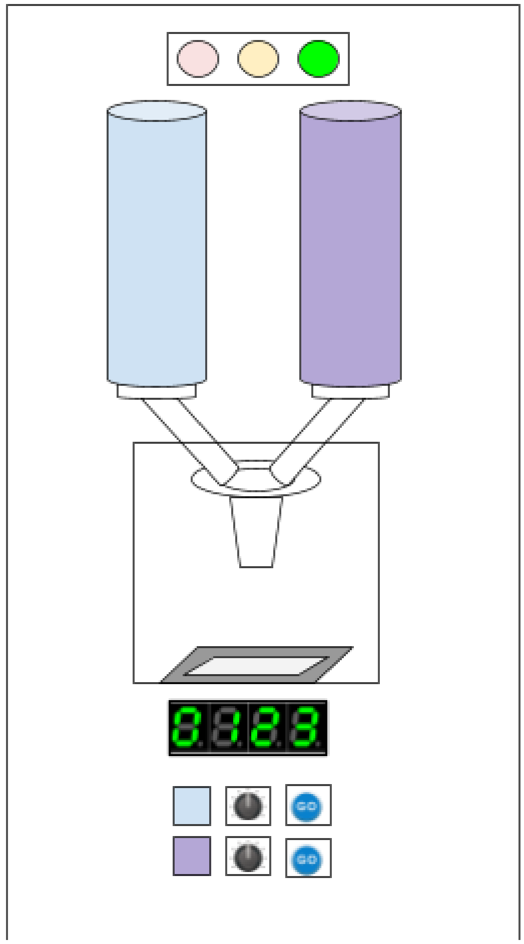

Figure 3: Physical Design Mockup

Description: On the top of the machine is the status bar. In this diagram, the status is green, hence it's bright and the other lights are off. The design mockup consists of 2 bottles with tubes that lead into a funnel which transports items to the outside container spot. The gray rectangle located in the middle of the diagram is the spot in which a container will be placed. This is also where the pressure sensor will be located. Below shows the target weight of the item chosen by the user via the knobs. To select an item to dispense the user presses the corresponding go button.

2.3 Subsystems

2.3.1 Power

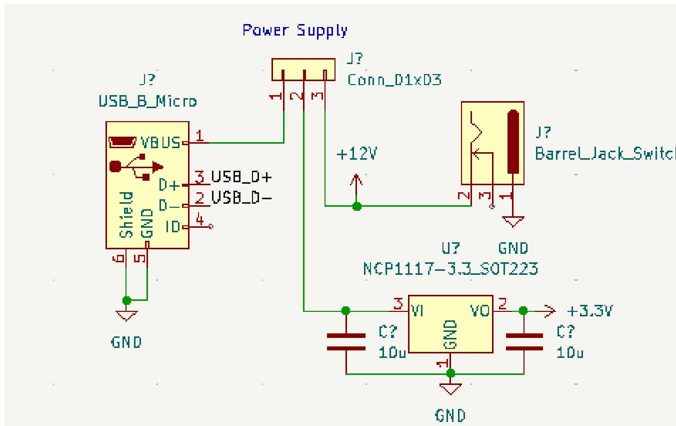

Figure 4: USB and 12V supply fed into voltage regulator

We plan on using a 12V power supply that will be used to power the highest voltage components of our design, and then stepped down to 3.3V to power the rest of the devices. Furthermore, we require access to the microcontroller for initial programming and reprogramming. The choice of voltage regulator, the NCP1117, comes from the very detailed hardware design guide provided for the rp2040[2]. It is fortunate that this regulator will accept voltages in the range of 5-12V, allowing us to use it for both power supplies. We require a current rating for each voltage line. For the 3.3V line, the only significant elements drawing current are the microcontroller, flash memory and the screen. We expect these together to draw less than 250mA of current. To be safe we increase this requirement to 500mA.

The 12V line will be attached to 2 solenoids each rated for a maximum of around 300mA. In the worst case scenario, both solenoids engage, drawing roughly 600mA of current. To be safe and to account for the smaller components on the line, we require a power supply capable of delivering 1A on the 12V line.

Requirement	Verification
Power supply capable of generating 500mA on 3.3V line at +/- 0.2V	While the machine is running, we will measure the voltage at the voltage transformer, as well as measuring the current drawn under load.
Power supply capable of generating 1A on 12V line at +/- 0.5V	While the machine is running, we will measure the voltage at the voltage transformer, as well as measuring the current drawn under load.

Table 1: Power Requirements and Verification

2.3.2 Dispensing Subsystem

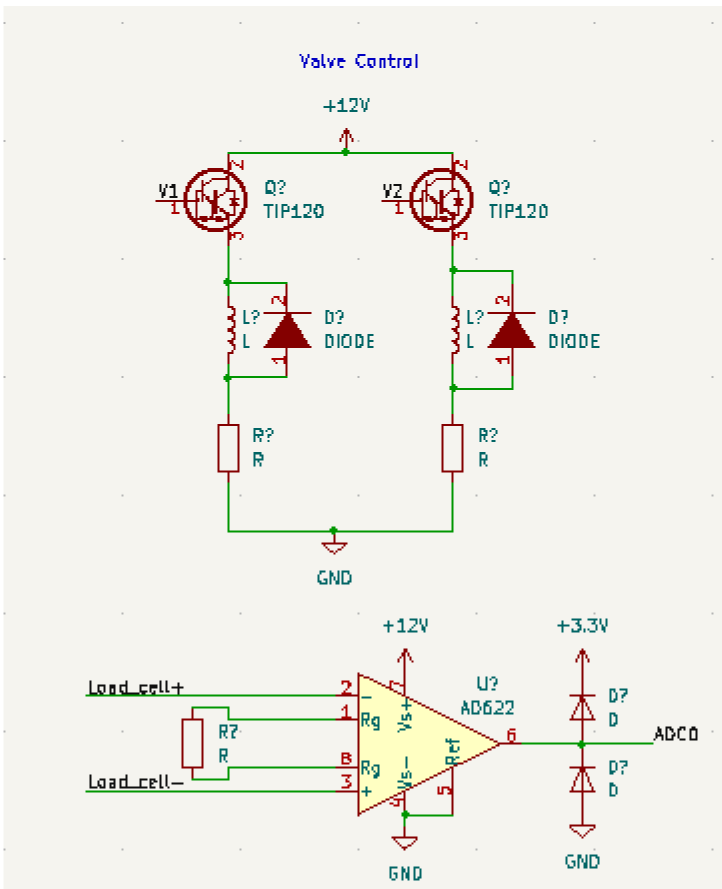

Figure 5: Dispensing subsystem

The control valves should be able to shut off quickly enough to guarantee that the 10% tolerance value for dispensed goods is achieved; factoring in the starting weight of the container, and any material left in the line. The valves will be solenoid valves that are rated for 12 V with ideally low current draw, around 0.5 amps. These will receive data signals from the microcontroller on when to open and close the valve. The microcontroller will be connected to a gate (MOSFET) that will allow it to interface with the higher voltage components while outputting a lower voltage data signal around 3.3V. The solenoid will take inductive kickback into account to make sure that the valves are properly closed.

Our load cell will be connected to a Wheatstone bridge to ensure its correct operation. Then that will be fed through the AD622 instrumentation amplifier. This will boost the signal and allow it to be read by our microcontroller's ADC. Furthermore, in order to not overload the ADC we add an ADC protection circuit at the output of the amplifier.

Requirement	Verification
Voltage across inductor should always be less than the specified maximum voltage of the MOSFET	Measure the voltage across the inductor before and after closing the MOSFET switch.
The load cell should be able to weigh items with a tolerance of 10g for weights up to 5 kg.	Check if the weight detected by the load cell is within the tolerance of the item's actual weight as recorded on a full scale.

Table 2: User Interface Requirements and Verification

2.3.3 User Interface Subsystem

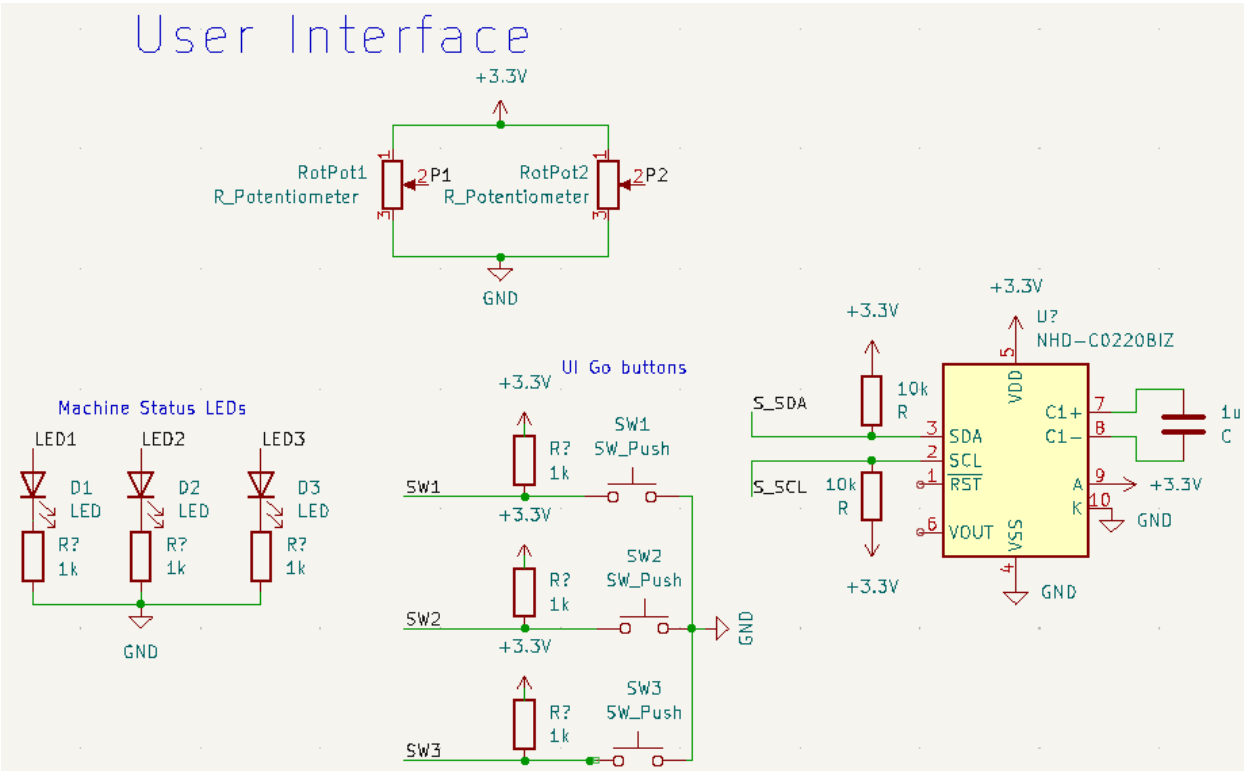

Figure 6: User Interface Schematic

The user interface will consist of potentiometers, a screen, push-buttons, and machine status LEDs. The potentiometers will allow the user to choose the amount they wish to dispense and display that value on the screen. To do this, the potentiometer will feed in a value to the microcontroller which will then display it on the screen. The set of buttons will correspond to the items being dispensed as well as a row of status LEDs to show what process the vending machine is currently in.

We have chosen the NHD-C0220BIZ screen as it interfaces nicely with I2C, and consumes little power. This screen should display the current item selected as well as the amount selected.

Requirement	Verification
The screen should use an I2C connection and be capable of displaying the item to be dispensed as well as its quantity with appropriate units.	Check that the correct text is displayed on the screen in 1 second after a button is pressed or a potentiometer is moved
Status LEDS should match machine state within 1s of machine state changes	Check the current machine state and see if the LED is the correct output. The amount of time required for the correct LED to turn on should be within 1 second of the state change.
The machine should dispense the proper item.	Pressing a button will cause the machine to attempt to dispense the proper item.

Table 3: User Interface Requirements and Verification

2.3.4 Control Subsystem

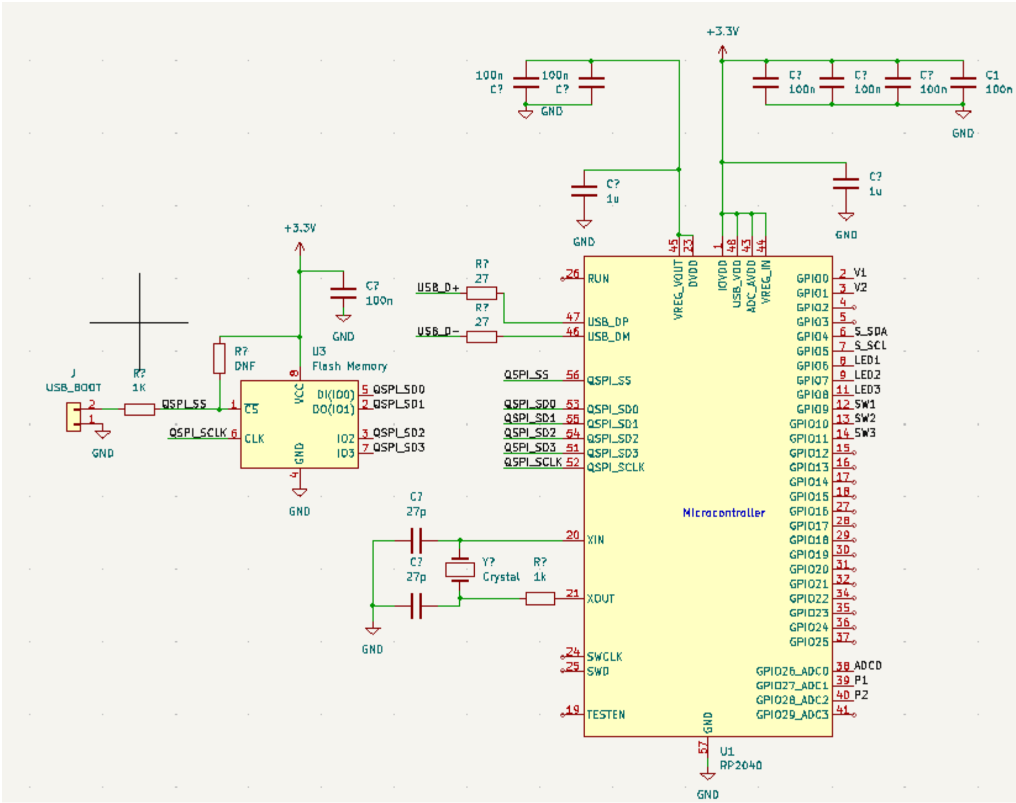

Figure 7: Microcontroller and Flash Memory

An RP2040 will be used as the main control unit in the dispensing machine. We chose this for a number of reasons. Firstly, the processor is cheap (around $1) and readily available, reducing the cost of making the vending machine. Secondly, we require one I2C connection as well as a max of 10 GPIO pins which the RP2040 is capable of handling. The design drawn above is heavily inspired by the one described in the rp2040's hardware design document [2].

The RP2040 requires flash QSPI flash memory in order to store programs and data. We have chosen a 32MB flash chip. This is surely more than we will ever need, but it allows for further enhancements on the design. Though the rp2040 has a built-in oscillator, it is not precise enough to properly communicate over USB. Therefore, we have added a 12Mhz oscillator to facilitate this.

Requirement	Verification
The microcontroller's ADC should be precise enough to measure 250 unique values from the load cell	The microcontroller outputs weight values with increments of ~10 grams for an average item load of 2 kg
Microcontroller sends proper signals to other subsystems.	The correct text displays on the screen, the right item is chosen, machine status gets updated on the UI, and the right item gets dispensed.

Table 4: Control Requirements and Verification

2.4 Tolerance Analysis

The accuracy of the load cell in the Container Placement Zone will determine how well we deliver on the requested weight of the order from the user. We will also need to take into account how long it takes the dispensing subsystem to shut off and the weight added during that process. The equation for this is simply written as follows:

$$M_{Total} = M_{Container} + M_{Dispensed} + M_{In\ Line}$$

Equation 1: Total Mass of container

If we know the total mass as well as the mass in the container before anything is dispensed, and the mass that has been dispensed, we know how much is left to be dispensed. We want to close the valve before we have reached the target weight on the scale as the material in the line will still flow into the container.

Flow rate(Kg/s) * Time(s) = Dispensed Mass(Kg)

Equation 2: Flow rate equation

If we can calibrate for each material's flow rate, we can know the time left for the material to flow out of the container. However, the flow rate may not be consistent for any given material, especially for viscous liquids and bigger solids.

2.5 Software Flowchart

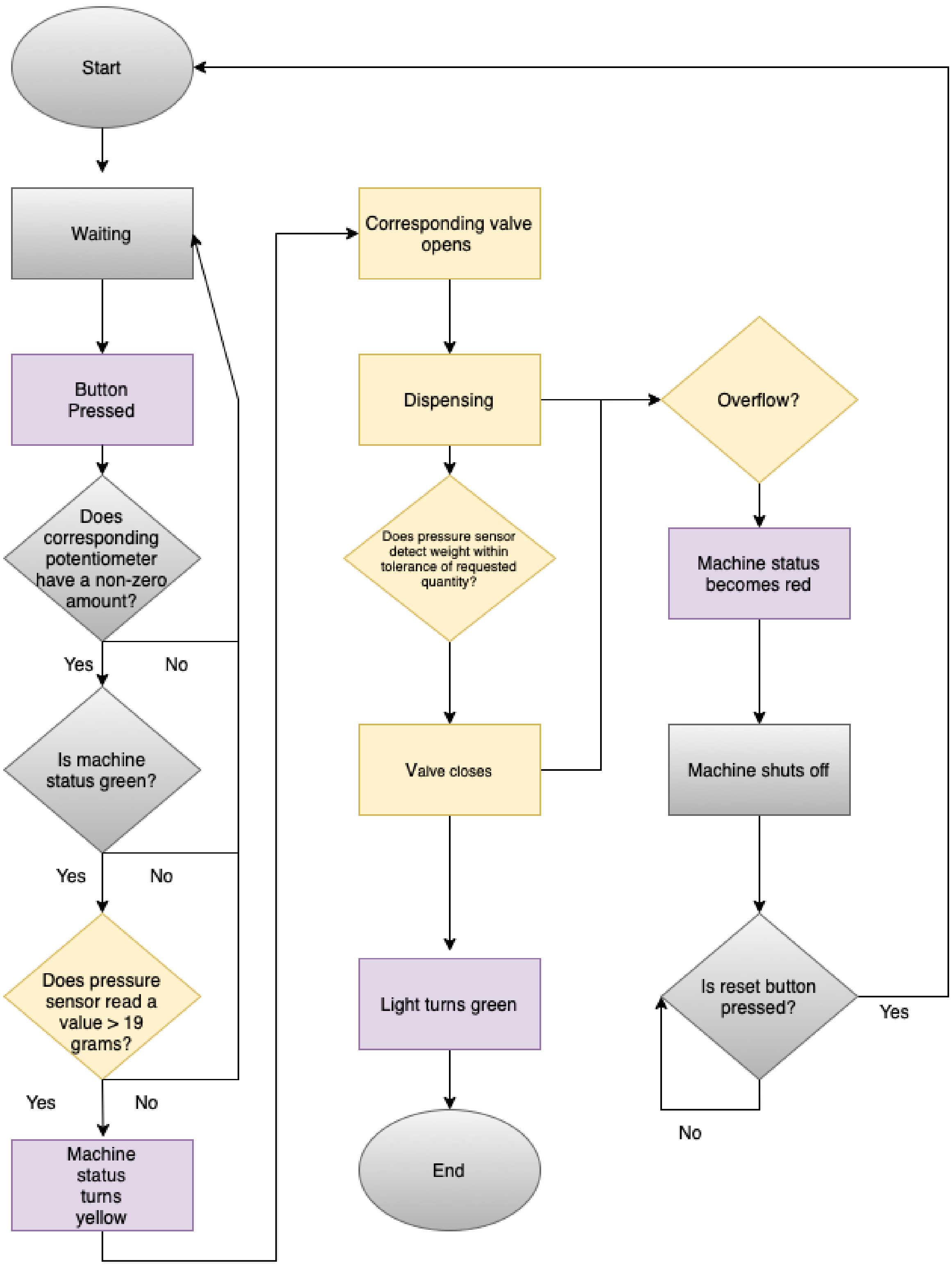

Figure 8: Software Flowchart

3 Cost & Schedule

3.1 Cost Analysis

3.1.1 Labor

From the 2020-2021 Illinois success report, there were 128 computer engineering graduates reporting an average salary of $105,352, and 65 electrical engineering graduates reporting an average salary of $80,296 [3]. Averaging these together we get an average salary for ECE graduates at $96,913. Assuming this salary is paid hourly and a normal 40-hour work week, we come to an average hourly wage of $46.59. Furthermore, we assume that each partner will work on this project for 10 hours each week. Therefore, the total cost of our labor comes out to $46.59 / hour * 10 hours/week * 10 weeks * 3 People * 2.5 Overhead Costs = **$34,942.50**

We were quoted a price of $400 and a time estimate of 3 - 4 days to build the machine from the Machine Shop.

This brings our total cost for labor to **$35,342.50**

3.1.2 Parts

Description	Part	Manufacturer	Quantity	Extended Price	Link
Microcontroller	RP2040	Raspberry Pi	1	$1.00	Chicago Electronics
I2C Screen	ERM2004-1	Buy Display	1	$14.46	DigiKey
Instrumentation amplifier	AD622	Analog Devices	1	$8.59	ECE Supply Center
Red LED	HLMP3.301 - Red	Avago Technologies	1	$0.16	ECE Supply Center
Green LED	HLMP3507 - Green	Avago Technologies	1	$0.18	ECE Supply Center
Yellow LED	HLMP3401 - Yellow	Avago Technologies	1	$0.21	ECE Supply Center
Memory	IC FLASH	Winbond	1	$0.94	Digi-Key

	32MBIT SPI/QUAD 8SOIC	Electronics			
Barrel Jack	Breadboard-friendly 2.1mm DC barrel jack	Adafruit	1	$0.95	Adafruit
Micro USB Plug	Micro USB Plug Female	Adafruit	1	$0.95	Adafruit
Voltage Regulator	NCP1117	OnSemi	1	$0.67	DigiKey
Push Buttons	0661273664773	Cyclewet	3	$6.29	Amazon
Rotary Potentiometer	CA-WH148-10K BK	TWTADE	2	$10.99	Amazon
Diode	IN4001	ON Semiconductor	4	$1.50	Adafruit
Transistor	TIP120	Fairchild	2	$2.50	Adafruit
100 nF Decoupling Capacitor	1C20Z5U103 M050B	Sprague	4	$0.92	ECE Supply Center
Solenoid	Plastic Water Solenoid Valve - 12V - 1/2" Nominal	Adafruit	2	$13.90	Adafruit
12V Power Supply	ALITOVE DC 12V 5A Power Supply Adapter	ALITOVE	1	$12.99	Amazon
Load Cell	4541	Adafruit	1	$3.95	Adafruit
27Ohm Resistor	CMP0805-FX-27R0ELF	Bourns	2	$0.44	Mouser

1K Ohm Resistor	ERJ-1RHD1001C	Panasonic	6	$0.90	Mouser
10k Ohm Resistor	CRGCQ0603J10K	TE Connectivity / Holsworthy	2	$0.24	Mouser
27pico Farad Capacitor	C0805X270J5GAC7800	Kemet	2	$0.68	Mouser
1u Capacitor	TMR107B7105KA-T	Taiyo Yuden	2	$0.38	Mouser
10u Capacitor	CL32Y106KCVZNWE	Samsung	2	$2.24	Mouser
100n Capacitor	C1005X7R1C104K050BC	TDK	6	$0.60	Mouser
12MHz Oscillator	ABLS-12.000MHZ-B4-T	Abracon	1	$0.22	Mouser
Total Cost				$85	

Table 5: Parts List

Grand total = $35,427.50

3.2 Work Distribution

	Michael	Jackson	Lyla
9/19	Design Document	Design Document	Design Document
9/26	Design Document	Design Document	Design Document
10/3	Start PCB	Order Components	Itemize
10/10	Finalize PCB	Finalize PCB	Finalize PCB
10/17	Assemble Load cell and valve arrangement	Refine State machine	I2C code & create UI
10/24	Solder PCB/ Measure output voltages and currents for RP2040, memory, solenoids, load cell, and screen	Solder PCB/Check RP2040 output signals and weight increments	Solder PCB/ Confirm that screen and LEDs display properly
10/31	Create and test circuit for refill mechanism	Debug state machine and circuit python code	Check refill mechanism's physical operation
11/14	Mock Demo	Mock Demo	Mock Demo
11/21	Fall Break	Fall Break	Fall Break
11/28	Final Demo	Final Demo	Final Demo
12/5	Presentation/Report	Presentation/Report	Presentation/Report

Table 6: Work Schedule

4 Ethics and Safety

4.1 Ethics

Ethics are of paramount importance to our project. Our design aims to reduce plastic waste in the environment, thereby complying with the sustainability clause of the IEEE code of ethics [4]. However, we recognize that shortcomings on our part may violate other provisions of the code. To that end, we promise to ensure that we will credit any work we may wish to build from.

Additionally, we're going to ensure that the environment where the machine is assembled demonstrates ethical behaviors such as obeying all rules and regulations, communicating effectively, being professional, and maintaining core group values such as trust, respect, and accountability [5]. At all times, members of the group will be open to hearing feedback.

Although in a perfect world, we would only order from vendors that are environmentally sustainable and have standards of ethics that we do, for this project that wasn't possible as we need parts in a timely manner and for low costs.

4.2 Safety

Since the machine lacks a Food Sanitation Certification and Food Handler Training, we will not permit anyone to consume anything coming out of the machine. Our machine has a status bar which indicates to the user whether it may be used or not. There will also be directions posted on the machine for how to use it. On the exterior of the machine, warnings will be posted that include keeping the dispensing area clear, not hitting the *go* button until the container is placed and ensuring no body parts or clothing are in the Container Placement Spot etc. On the backend, we will have a series of checks that ensure there are no leaks or spillage that could damage the food inside.

Additionally, in our design, we will make sure that food and wires or electricity do not mix. Finally, we will have some emergency shut-off button on the inside, in case a machine stocker runs into any issues.

Works Cited

[1]"Containers and Packaging: Product-Specific Data," *EPA*. [Online]. Available: https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/containers-and-packaging-product-specific#PlasticC&P. [Accessed: 14-Sep-2022]

[2] "Hardware Design with rp2040," *Raspberry Pi Foundation*. [Online]. Available: https://datasheets.raspberrypi.com/rp2040/hardware-design-with-rp2040.pdf. [Accessed: 25-Sep-2022].

[3]"2020-2021 Illinois Success Report," *Illini Success*, 2021. [Online]. Available: https://illinisuccess.illinois.edu/annual-reports/. [Accessed: 26-Sep-2022].

[4]"IEEE code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 14-Sep-2022].

[5] "23 Ethical & Unethical Behavior Examples in Workplace," Formplus Blog. [Online]. Available: https://www.formpl.us/blog/workplace-ethics [Accessed: 29-Sep-2022].