ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Project: GreenCan

IFESI ONUBOGU
(onubogu2@illinois.edu)
MATT WILDENRADT
(miw3@illinois.edu)
MICHAEL OBUNIKE
(obunike2@illinois.edu)

TA: Sainath Barbhai

May 3, 2023

Abstract

This document provides an outline of the design for project Greencan, a can recycling system that takes into consideration its immediate users, end recycling agencies, and the organization it is deployed. Inspired by our concern for the environment and respect for data collection, we designed Greencan to prevent metal shards from harming users while can-crushing is in progress (thus protecting its immediate users); avoid the recycling of non-empty cans (thus ensuring feeder recycling agencies receive optimal inputs) and reports the number of cans to organizations that employ Greencan. The design of Greencan was a long, fun, and collaborative process made more manageable by breaking the system into three main subsystems we were individually in charge of– the can-counting, can-crushing, and control subsystems.

Contents

1 Introduction

1.1 Problem

According to an article by the National Association of Convenience Stores[1], crushing cans before recycling saves space, providing more recyclable material per container and makes transportation more efficient. However, the average person's means of crushing cans before recycling neither has a system to prevent the crushing of non-empty nor a safety guard for its users– what happens if a hand gets stuck in front of the crushing surface some metal shards fly around during the crushing? Some concerned threads outline these concerns in a discussion forum (Health and Safety Tips). So the problem is apparent: improving our current system of recycling cans with safety in mind.

1.2 Solution

We intend to make an Aluminum can recycling machine that prevents the recycling of non-empty Aluminum cans and keeps track of how many cans have been recycled for documentation purposes at larger organizations. This solution encourages large-scale recycling through a user-safe system that prevents the recycling of non-empty and/or pressurized cans.

The machine will use a latch sensor (similar to those used to turn the light on or off in refrigerators) to tell when the can-crushing enclosure's door is shut and a load cell to tell when an empty aluminum can (weighing from 12g to 16g) has been inserted into the machine. When, in addition to the previous two conditions, there is no overload of cans detected in the collection system (i.e. as long as there are not too many cans in the machine's collection bin), and the start button is also pressed, a PCB will send a signal to the motor, which will crush the can. The motor will proceed to crush the can until a potentiometer attached to the motor indicates to the PCB that the can has been crushed to the point where, if the motor retracts, it will fall into the can disposal chute. If, before this point, the current monitor attached to the motor indicates that an irregular amount of current is being exerted to crush the can, the motor will be retracted immediately to prevent damage to the machine. Assuming the can is sufficiently crushed, it will be allowed to fall into the disposal chute by the retracting motor, breaking the beam of an IR sensor placed at the bottom of the chute, and sending a signal to the PCB. The can will proceed to fall into the disposal bin below the device, while the PCB will internally increment its count of the number of cans recycled and display the current number on a small display.

To ensure only empty cans are crushed, our system will monitor two values: the weight of cans placed into the crushing cubicle and the current drawn from the motor. If its weight exceeds the weight of an empty can or the current crosses an experimentally determined threshold, a red LED will glow (indicating to the user that the machine will not crush the can placed inside, sending the machine into a do not accept state). There will be a collection bin for the crushed cans.

At any point in time, the system is one of four internal states: A start state (which it will be frozen in temporarily if it detects an invalid can on the load cell, a can blocking the disposal chute due to a full disposal bin, or an open door to the can insertion area) where the machine can be asked to crush a can, a crush state (which will only be triggered from the start state if none of the freezing conditions are true and if the go button is pressed) where, assuming no problems are detected, the can will be crushed by extending the crushing piston and retracting it once the can has been crushed small enough to fall into the disposal chute, a retraction state (which only occurs to immediately retract the piston if the door is suddenly opened or if the current is detected to be to unsafe) to implement safety measures during crushing, or an increment state (which occurs after the crush state assuming a can is detected by the disposal chute as the piston is retracted) to increment the recorded number of cans crushed and continue retracting the piston. The current state of the machine, including which of the four internal states it is in and the presence of problematic signals (too much weight, too many cans, can door open) might also be indicated by a set of LEDs.

1.3 Functionality

There are three high-level project functionalities. How they align with the goal of the project is explained:

- The system only crushes empty cans when inserted into the can-crushing space: This ensures that only empty cans are crushed by the system and ultimately delivered to recycling agencies. This is important because recycling agencies have automated weight-based can-material-sorting mechanisms rendered inaccurate by liquid-holding cans.

- The crushed cans are collected in the collector until it is full or the collection duct is obstructed (in this case, the machine goes into a mode where it must be serviced to continue operation): This is a way of ensuring can-recycling data is collectible from our system.

- Metal shards are prevented from harming users: our system is designed to be safe for immediate users. To ensure their safety, we have inbuilt safety mechanisms such as a protective door to ensure the users while can-crushing is in progress.

1.4 Subsystem Overview

1.4.1 Can Crushing Subsystem

This subsystem is responsible for ensuring the cans are consistently and adequately crushed. The motor drives the can-crushing platform to be as thin as 0.28cm with a tolerance of 5% to allow the crushed can to pass through the collection chute. Opening the door should stop the motor while crushing within 5 seconds. Force sensors should correctly ensure that only empty cans are crushed with an accuracy of at least 90 percent.

1.4.2 Control Subsystem

This ensures the proper control signals are sent out to the can-crushing subsystems and act on inputs from the can-counting subsystem to display the crushed can count. It runs a finite state machine explained in section 2.
The PCB should stop the motor from running when it is stalling. The system also prevents the motor from crushing if the force sensor detects non-empty cans. The PCB should internally keep track of the number of cans crushed in the current service cycle.

1.4.3 Can counting Subsystem

This uses an IR sensor to detect when a crushed can falls the collection chute and is responsible for detecting blockages in the collection chute.
The system, via its IR sensor, sends a signal to PCB to increase the number of cans crushed internally. The system indicates to the PCB when there is a blockage in the collector chute, which needs to be serviced before the system accepts subsequent cans.

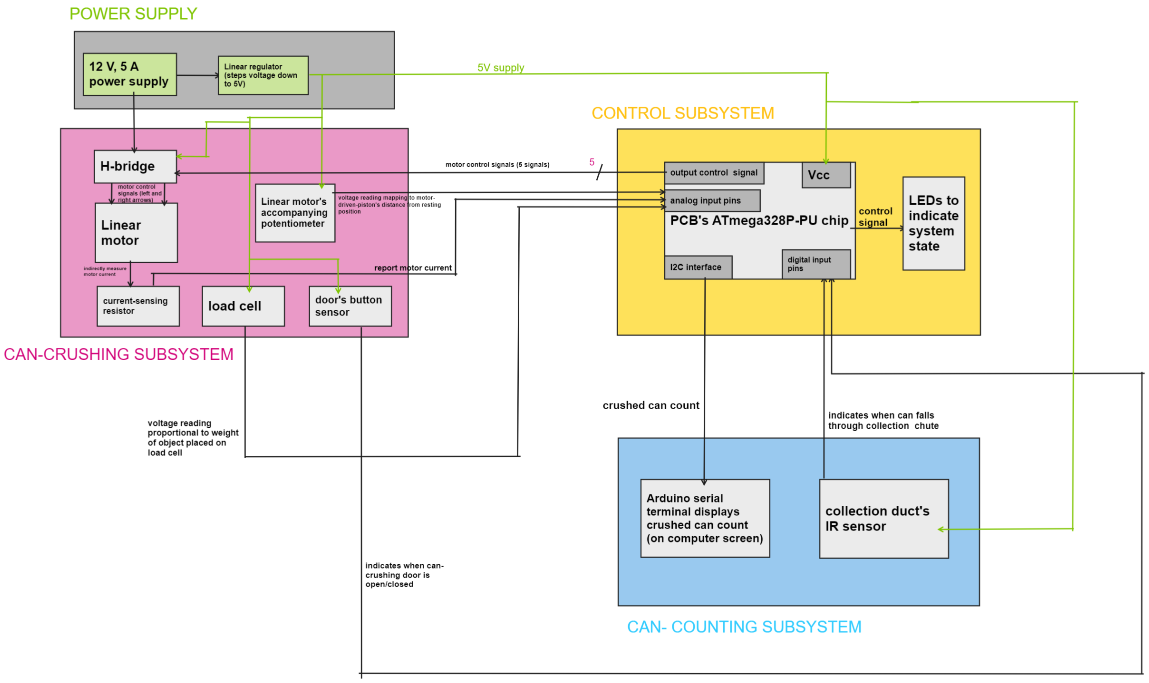

POWER SUPPLY
12 V, 5 A power supply
Linear regulator (steps voltage down to 5V)
5V supply
CONTROL SUBSYSTEM
H-bridge
motor control signals (left and right arrows)
Linear motor
indirectly measure motor current
Linear motor's accompanying potentiometer
voltage reading mapping to motor-driven-piston's distance from resting position
motor control signals (5 signals)
5
report motor current
current-sensing resistor
load cell
door's button sensor
CAN-CRUSHING SUBSYSTEM
output control signal
Vcc
analog input pins
PCB's ATmega328P-PU chip
control signal
LEDs to indicate system state
I2C interface
digital input pins
crushed can count
indicates when can falls through collection chute
voltage reading proportional to weight of object placed on load cell
Arduino serial terminal displays crushed can count (on computer screen)
collection duct's IR sensor
indicates when can-crushing door is open/closed
CAN- COUNTING SUBSYSTEM

Figure 1: Block Diagram of Machine

2 Design

The design of our machine could be broken down into two categories, the three subsystems as highlighted by the block diagram and the hardware/software design. All of these will be discussed to show the importance of each part of the design and how each part connects with one another.

2.1 Block Diagram

The block diagram highlights the three main subsystems of the machine. These subsystems are the can-crushing system, control system (PCB), and collection system. The can-crushing system which does the actual crushing of the can as the name implies houses the H-bridge which sends motor signals to the linear motor, the linear motor which drives the piston and the can-crushing surface forward and backward as required, the potentiometer attached to the linear motor that sends signals to the PCB to indicate the location of the motor-driven piston at any point in time to either extend or retract the piston for the purpose of crushing the cans to the correct thickness as advised by recycling agencies, the door sensor which communicates with the PCB via signals to indicate when the safety door is open or closed, the current sensing resistor which communicates current readings to the control system and the force sensor which lies on the crushing platform detecting changes in pressure applied to it for the purpose of sending signals to the PCB to distinguish between an empty and a non-empty cans.

The control subsystem (PCB) houses the state machine which uses signals from the parts of the can-crushing system described above and the collection subsystem's IR sensor to determine what state the machine is at every instance in time.

The collection subsystem contains the IR senor which detects when a crushed can has fallen through the collection chute and sends a signal to the PCB to internally increase the number of cans crushed displays this value on the Arduino serial monitor. The IR sensor also serves as a means to tell the system when maintenance is required by sending signals to the PCB indicating if it has been blocked too long by a stuck can or overflowing collection bin.

2.2 Hardware Design

The overall hardware of the system is based off a wooden frame which provides a stable base for the entire assembly. The bulk of the individual hardware components of our system can be located in the can-crushing platform. The linear motor is at the heart of the machine and drives the piston with just enough force to crush an empty-can against the can-crushing surface. The potentiometer is integrated with the linear motor to adjust the piston's travel distance ensuring that the cans are crushed to optimal thickness. The various sensors which individually send corresponding signals to the PCB to be used by the state machine.

2.2.1 Design Description and Justification

1. Linear motor with potentiometer feedback : In figure 8 we see the linear motor which was made specifically upon request to be able to supply the 150 lbs of force needed to drive the piston with the can-crushing platform to crush the cans to adequately and to our desired thickness of 2 inches with the help of the potentiometer it comes with which at any point lets the system know the location of the piston which makes it possible for use this information to tell the system when we want our motor to begin retracting the piston.

2. Wasp H-bridge: The wasp H-bridge, figure 9, was added to send motor control signals to the the linear and essentially we chose this model H-bridge because it is able to handle the 12 V, 7 A that our linear motor could potentially draw up to.

3. Force sensor: The force sensor, figure 10, had a sensitivity which was perfect with regards to what we were trying to achieve as it was sensitive enough with errors of less than a gram to distinguish between empty and non-empty cans unlike our initial option of the load cell which read in up to hundreds of grams which was not ideal for our goal.

4. IR sensor: The IR sensors were able to detect when objects emit and reflect infrared radiation in milliseconds which was placed on either side right above the collection chute to detect when a crushed can fell through the chute and sent the signal to the PCB to internally increase the number of cans crushed.

5. AT-Mega328P: The AT-Mega328P micro-controller chip is an ideal choice for any design due to its low power consumption, extensive range of built-in peripherals and pins and ease of programming through the Arduino development platform.

The PCB's microchip (AT-Mega328P) also keeps track of Vm to make sure the machine is only crushing empty cans which also serves as a safety measure. The motor current is given as:

$$I = Vm/0.001\Omega \tag{1}$$

At certain times the current could get over the desired value for the motor if not controlled properly. To ensure that the system correctly rejects any object that can or is not be placed on the platform when this is the case, we experimentally determined the stalling current for the motor. We needed to keep the current sensing resistor's value low to minimize power loss caused by it:

$$P = I^2R \tag{2}$$

The PCB microchip's software indirectly measures the motor current (I). When it stalls, it has reached an evaluative value. This is because Vb (the back EMF) is directly proportional to the motor's angular velocity:

$$V b \alpha \omega \tag{3}$$

Additionally, to help minimize power loss further, we added a power saving state or rest state at the end of each crushed can iteration which can be broken out of via the go button.

2.2.2 Diagrams and Schematics of Subsystems

Contained in this section are all of our relevant subsystem schematics and diagrams. This includes schematics and board layouts for all of our PCBs, along with pictures of all subsystem components which are not readily visible from a larger scale view of our project. The first six figures outline our PCB for the control subsystem in its various iterations, followed by three outlining the subsystem components. The final two figures are of our machine and the completed project in use, respectively.

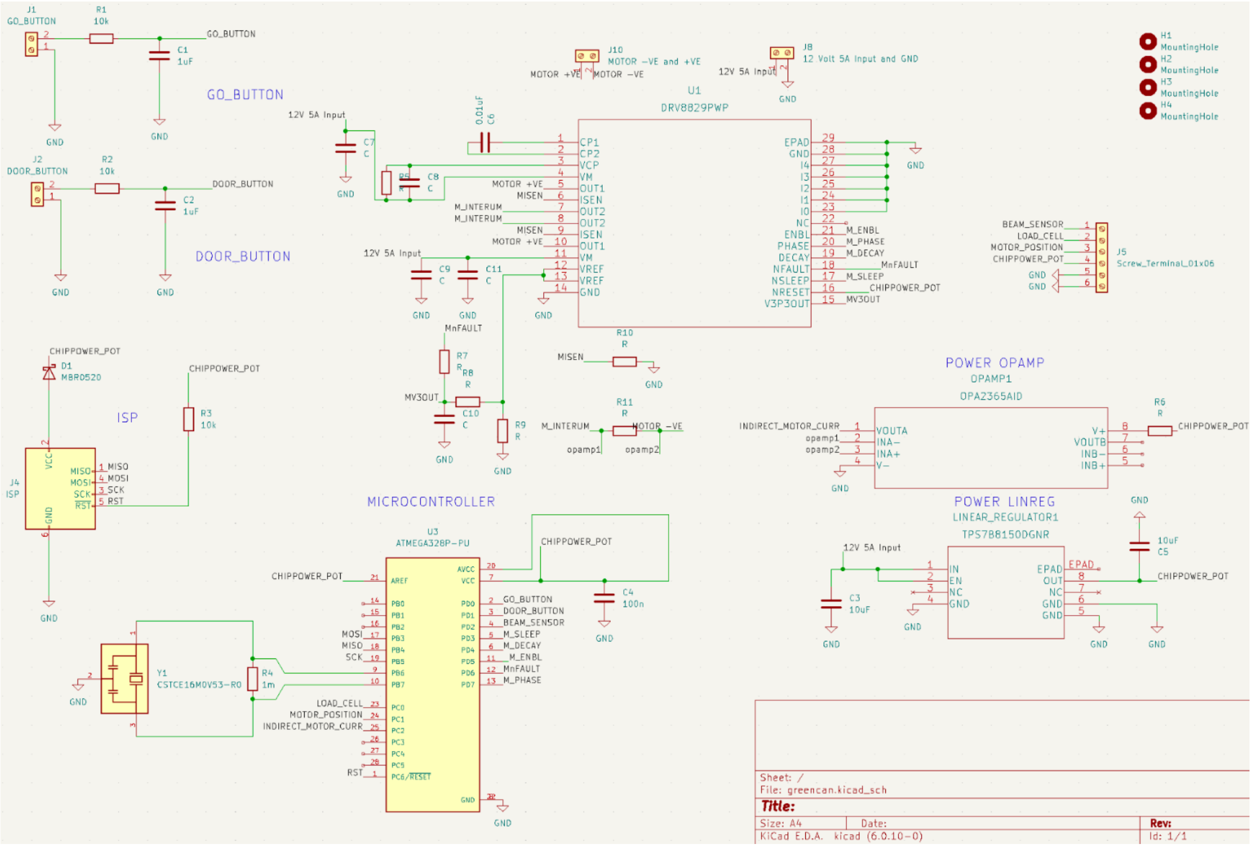


Figure 2: Schematic of Original PCB Design

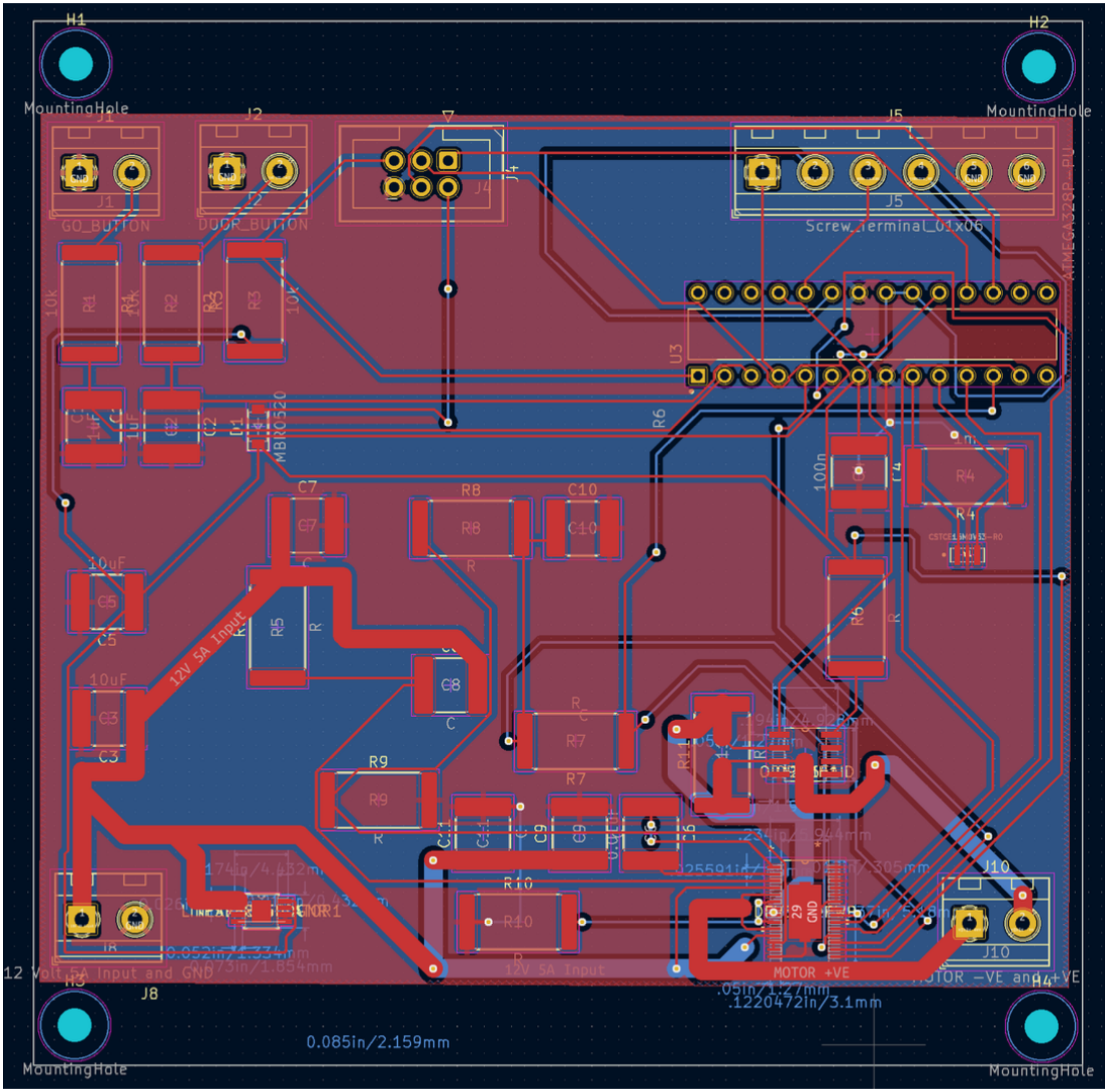


Figure 3: Layout of Original PCB

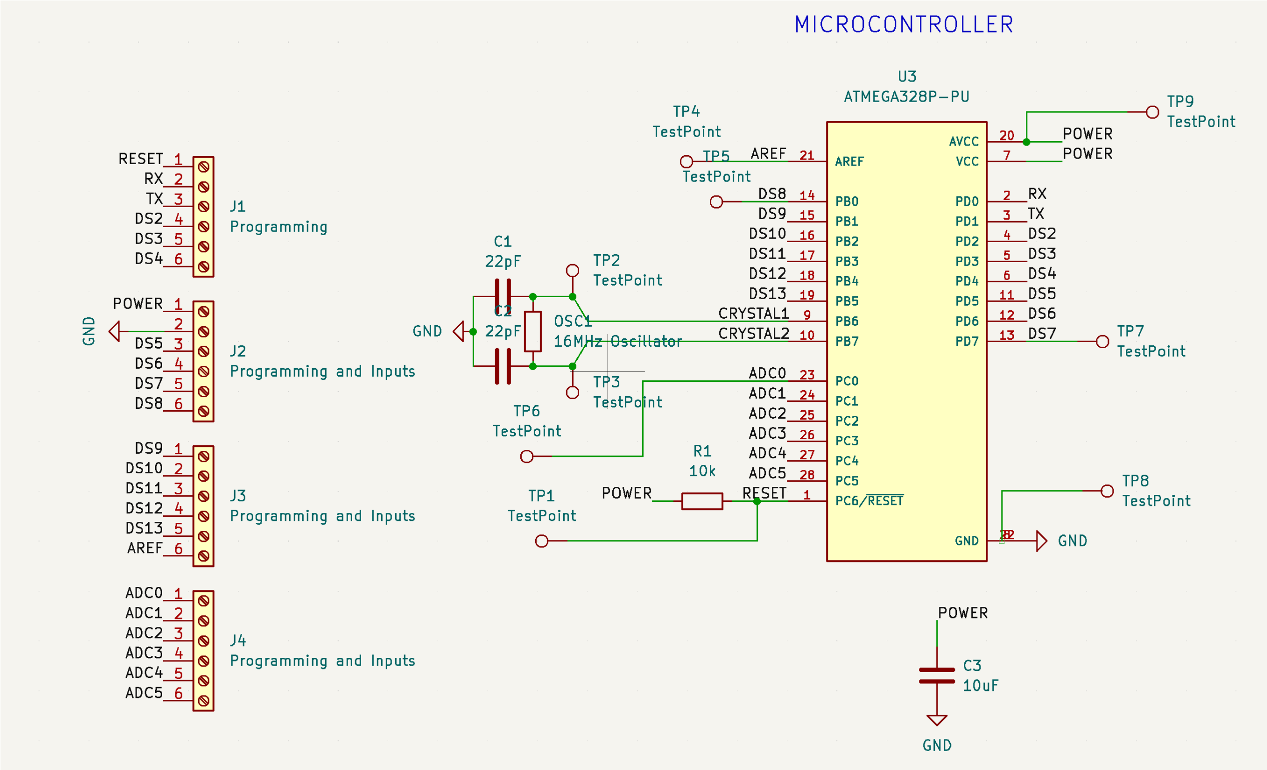


Figure 4: Schematic of Simplified PCB Design

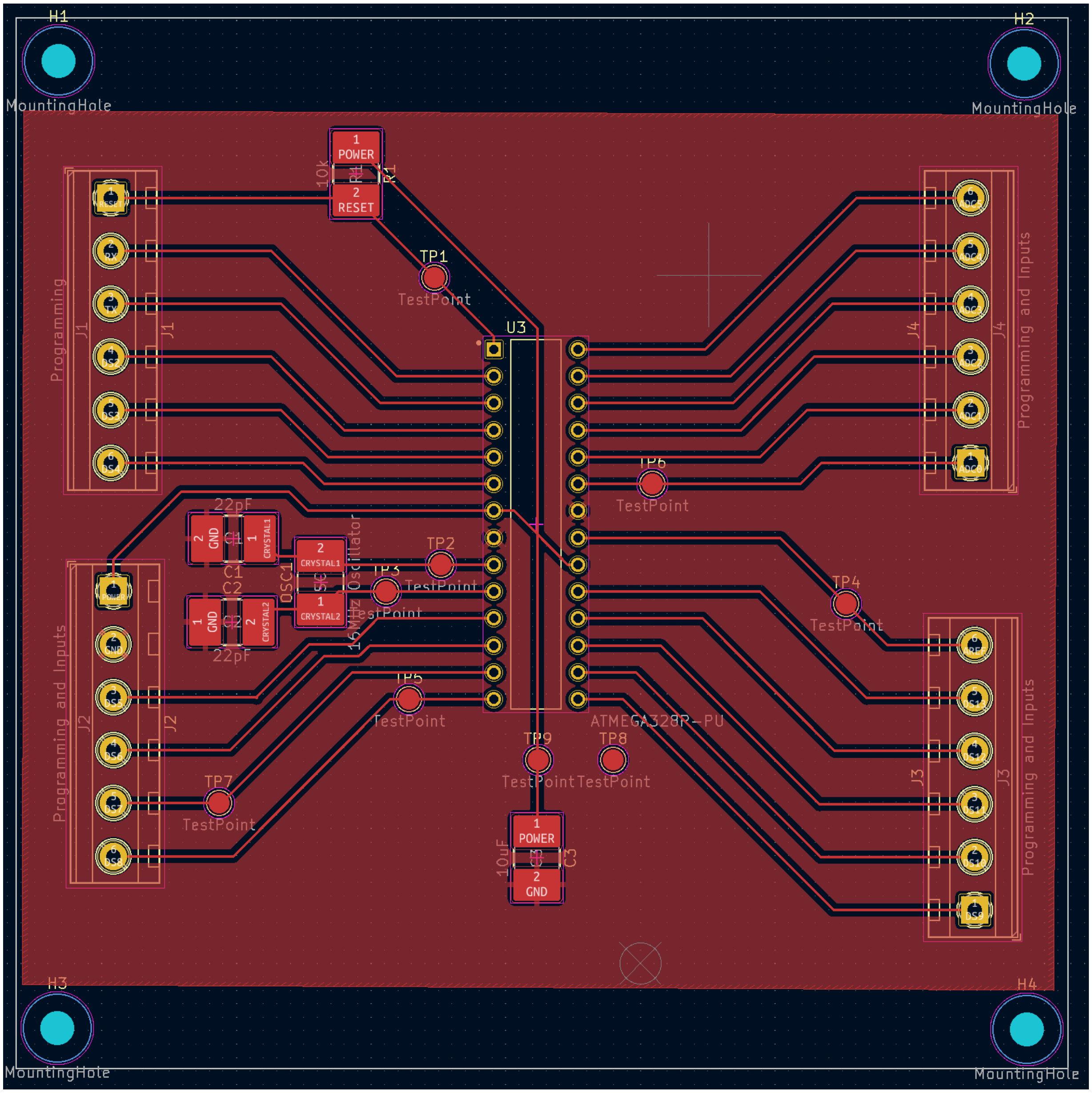


Figure 5: Layout of Simplified PCB Design

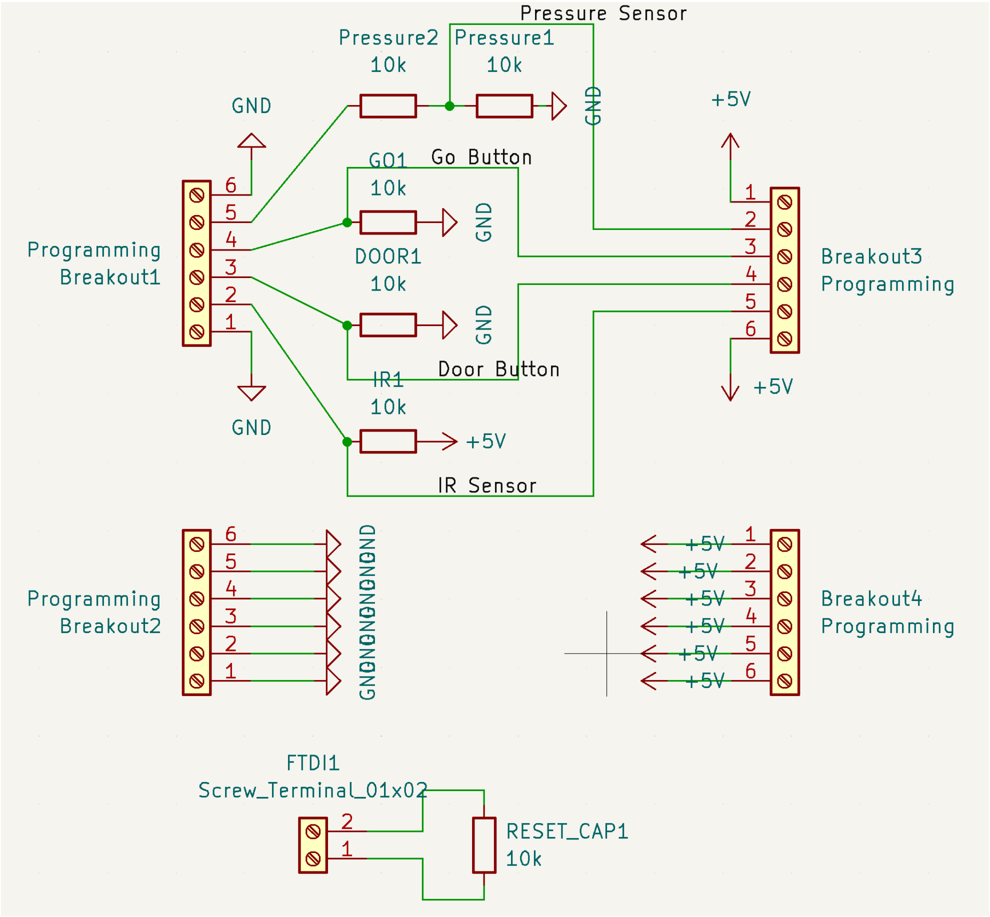


Figure 6: Schematic of PCB for Buttons and Programming

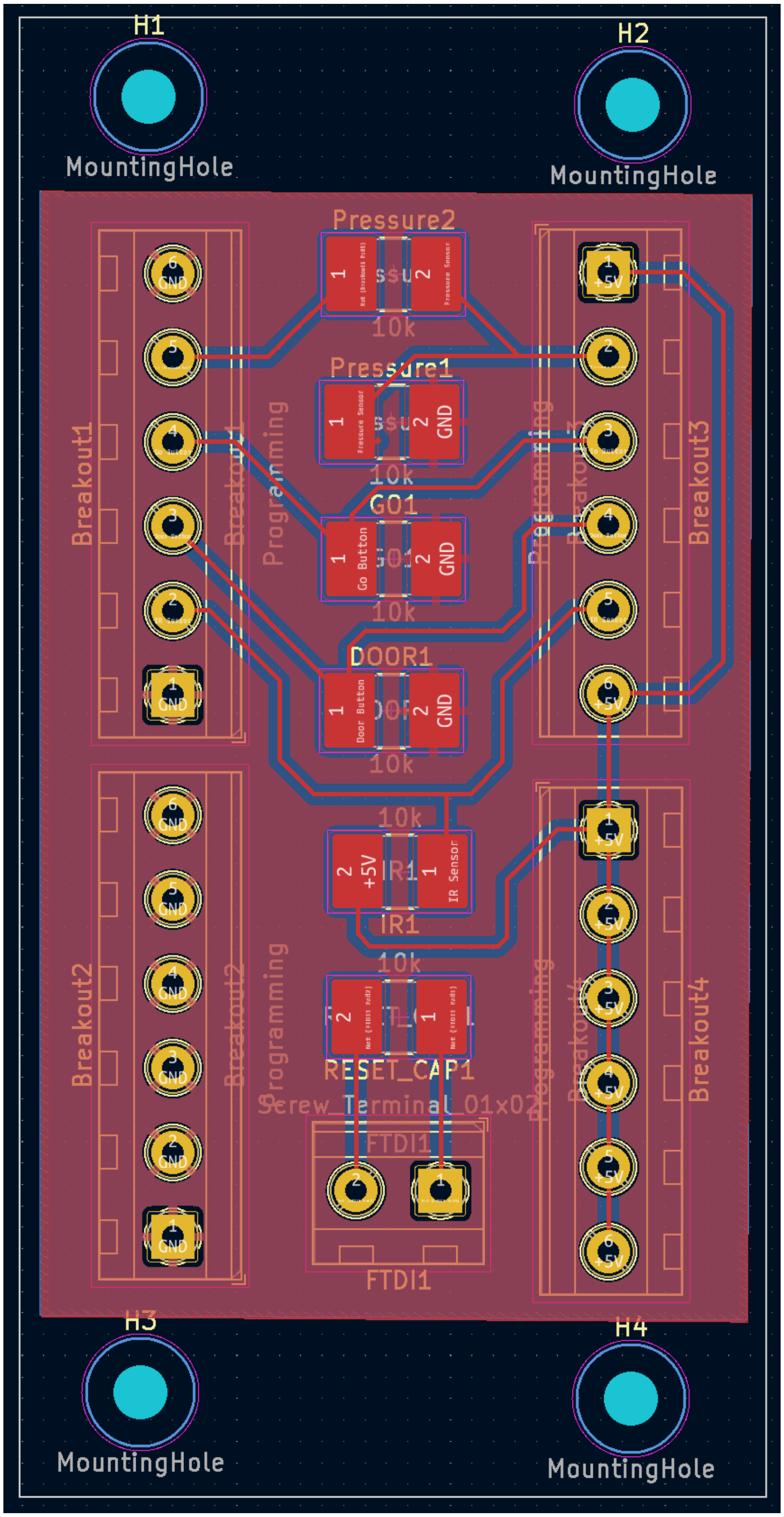


Figure 7: Layout of PCB for Buttons and Programming

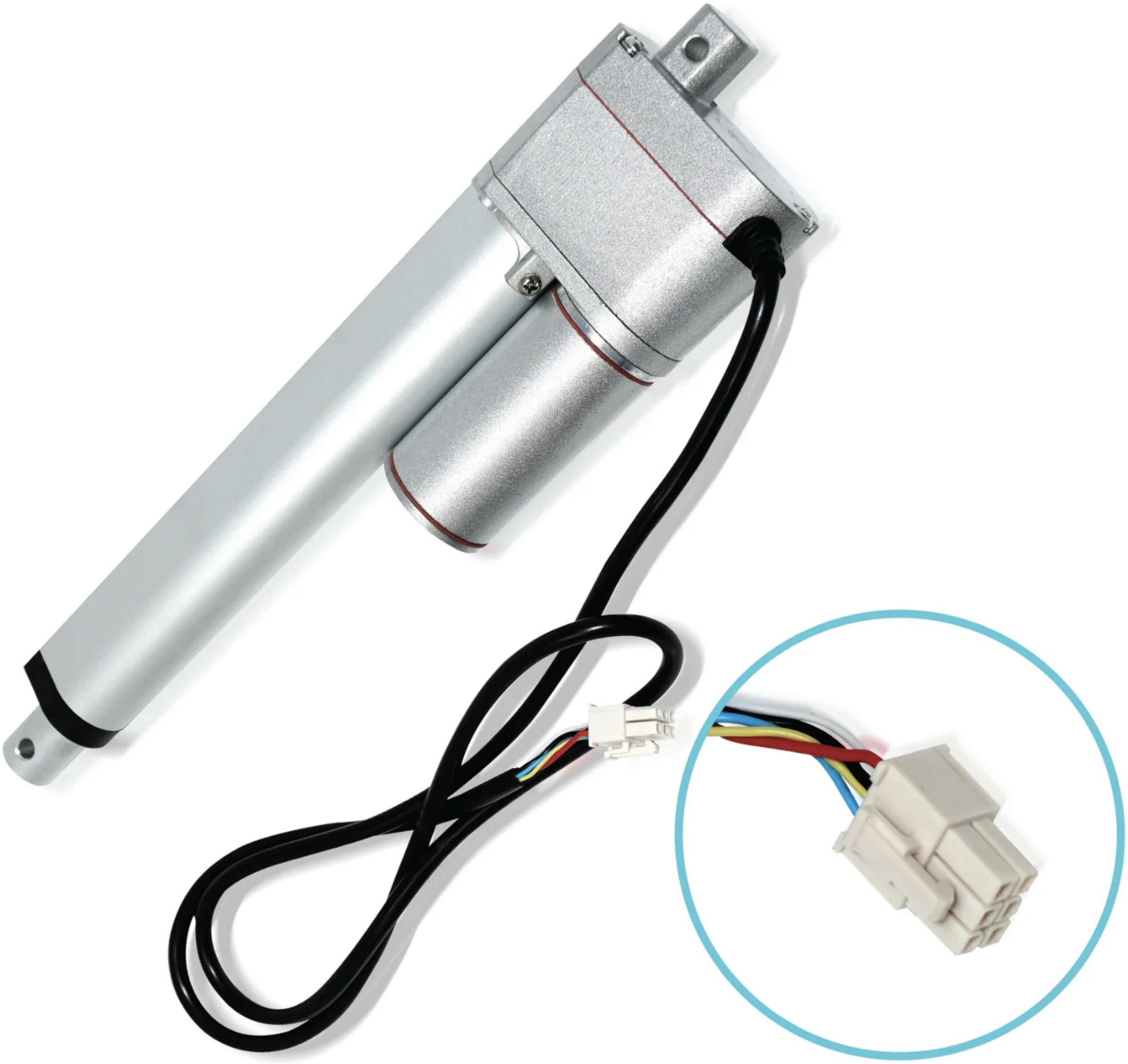


Figure 8: Picture of the Linear Actuator

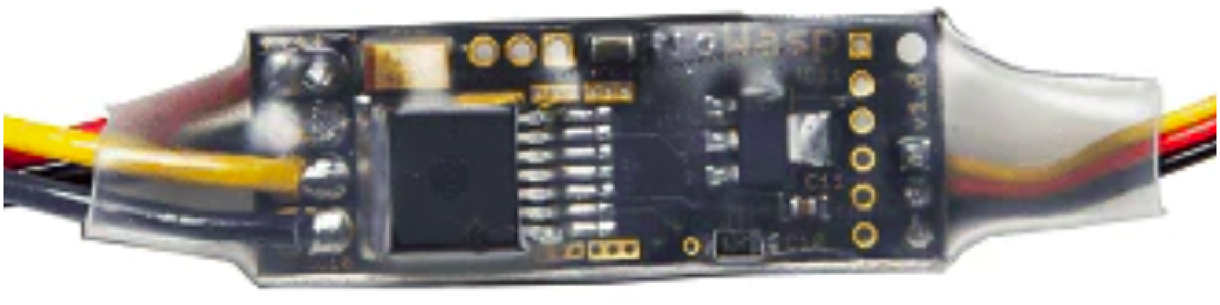


Figure 9: Picture of the ”WASP” Motor Controller

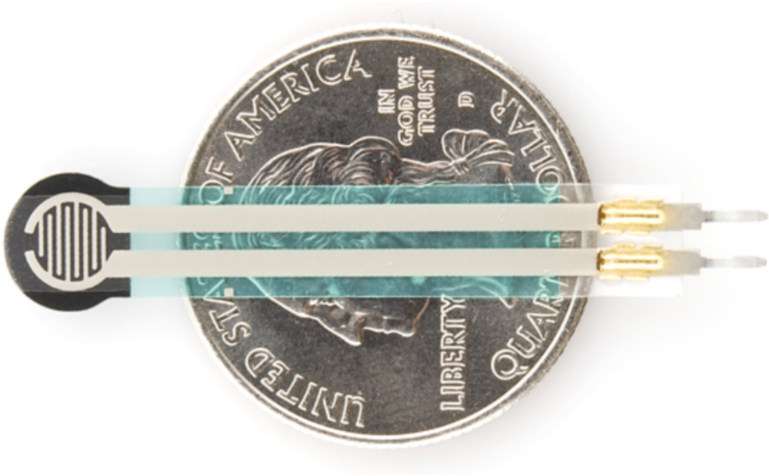


Figure 10: Force sensor

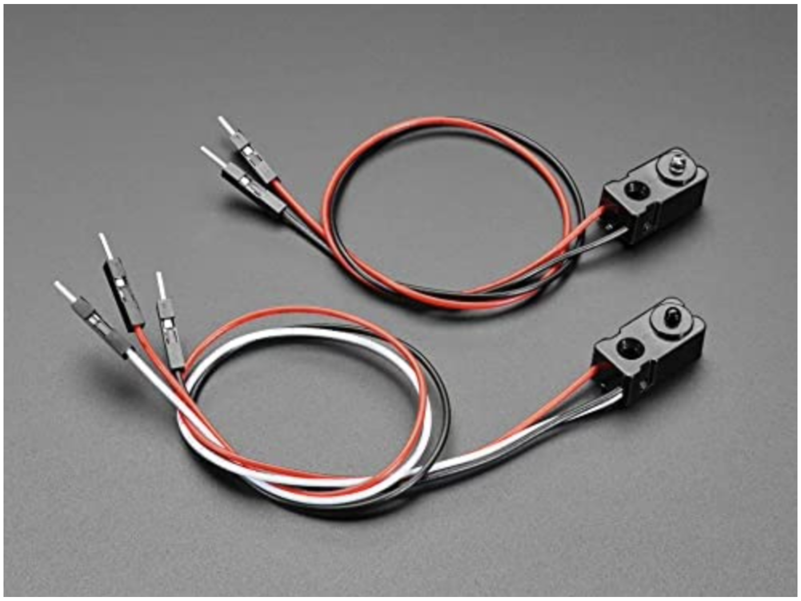


Figure 11: IR Beam Sensor

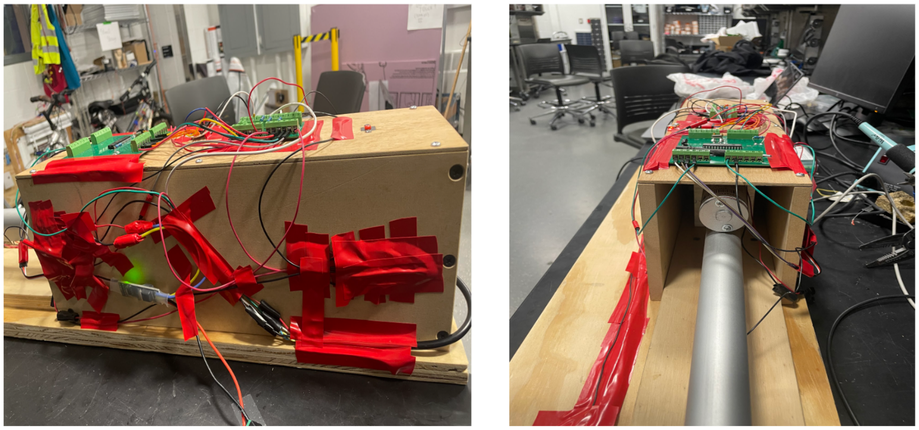


Figure 12: Physical machine consisting of main hardware components

Figure 13: TA Sainath next to the completed project

2.3 Software Design

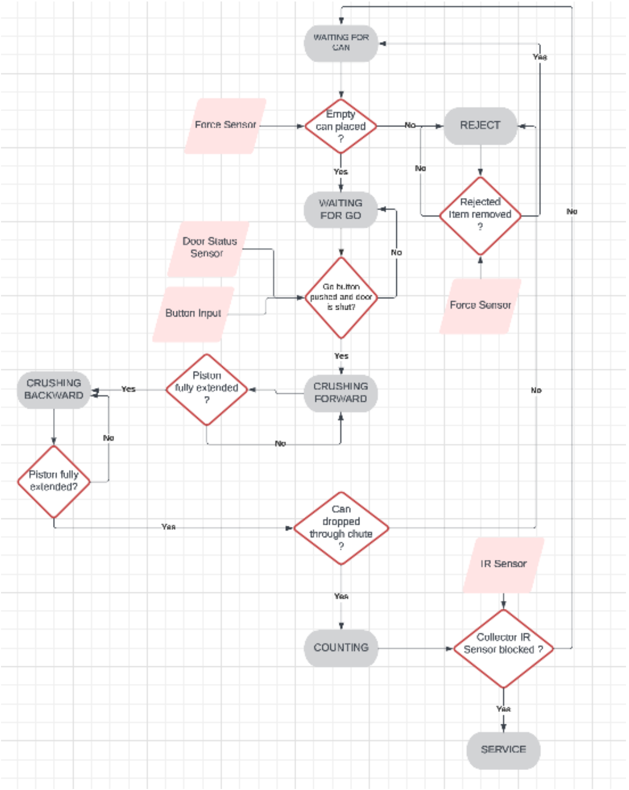


Figure 14: Flowchart of finite state machine

The PCB's microchip will have some code uploaded to it via an FTDI adapter which is a USB to serial converter which allows for a simple way to connect interface devices to USB. The code that is uploaded to the microchip contains a finite state machine consisting of six states which direct the system to what tasks to be currently run and parameters to be checked. These states include:

• WAITING FOR CAN: this is the initial state of the system. In this state, the motor is kept off and piston is fully retracted while the analog reading from the force sensor is checked to ensure an empty can is placed on the crushing platform as opposed to a full can. The system also checks that the door which gives access to the crushing space is kept shut for safety reasons before waiting for the user to press the go button at which point the system transitions into the CRUSHING state.

• CRUSHING: while in this state, the system begins it can-crushing process by the motor driving the can-crushing platform forward pushing the empty recyclable can against a surface until the potentiometer has sent the signal that the piston has extended to the desired length needed to crush the can to a thickness of 0.28cm. At this point the system transitions to the REVERSE state. If there are any disruptions during the crushing process especially one that could endanger the user like the door getting opened, the system would go into the PAUSE state till this has been resolved.

• REVERSE: in this state, there are no necessary signals or checks as the piston is just being retracted fully to its initial position after which the system goes into the COUNTING state.

• PAUSE: this state serves as a placeholder for when something disrupts the crushing process, whether the door has been opened during the crushing process which is a safety hazard or there is a stall in the crushing process for some reason. Once ready to continue crushing, the system reverts back to the CRUSHING state.

• COUNTING: in the counting state, the IR sensor has detected a can falling through the chute, so the system internally increases the number of cans crushed. The system also checks if the collector's IR sensor has been blocked for a long period of time which would indicate that there is a blockage that needs to be serviced at which point the system goes into the SERVICE. If that is not the case the system stays in rest mode till the user is ready to crush another empty can.

• SERVICE: this state allows for maintenance of the machine's collector bin in order to unblock the IR sensor. Blockage could result from a crushed can being in a position to block the sensor or the collector's bin overflowing with cans that have been crushed earlier.

2.4 Design Alternatives

The main problem with our original design was the load cell (Figure 16). The load cell we were provided with and planned on taking weight readings off of to distinguish between empty and non-empty cans was not sensitive enough or the weight range we were aiming to distinguish (12grams to 16grams which is about the average weight for an empty can). We decided to replace the load cell with a far more sensitive force sensor which was enough for us to tell empty and non-empty cans apart.

An initial worry for when we switched to the force sensor was where we going to place it in order to detect can weight and not get destroyed by the 150 pounds of force coming through the piston and can-crushing platform. The load cell was placed underneath the can crushing area where it would not obstruct the piston, but the force sensor had to be taped above the surface where it would be exposed to the can-crushing surface and piston. This caused our first force sensor to get damaged, but we figured a way to tape it in place in a way where it would not obstruct the piston.

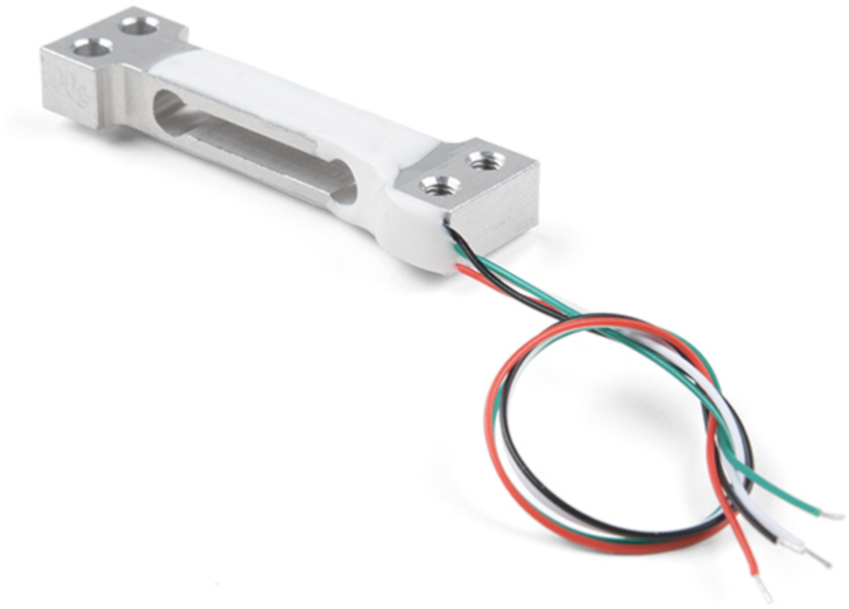


Figure 15: The Original Load Cell

3 Cost and Schedule

The price of the parts, as shown in Table 1 below, when combined with delivery and sales tax, comes to 161.87 dollars. A salary of 41.67 dollars per hour (about the average pay for an electrical engineer in the US) multiplied by 2.5 (overhead) and 84 hours worked results in 8,750.70 dollars for each team member. This sum needs to be multiplied by the number of team members, therefore 8,750.70 x 3 becomes 26,252.10 U.S. dollars in labor costs. We estimate that it will take 48 hours to produce the mechanical component at a cost of 22 per hour for the machine shop that worked on the mechanical portion of the design, or 1,056. This results in an overall expense of 27,469.97 U.S. dollars.

3.1 Cost

Table 1: Cost overview of project

S/N	Part	Manufacturer	Model Number	No. of parts	Total cost (USD)
1	Linear actuator with potentiometer	Progressive Automations	PA-14P-2-35	1	145.00
2	H bridge	Texas Instruments	DRV8829	1	5.00
3	Molex male terminal crimp	Molex	39000040	1	0.04
4	Current-sensing resistor	TE Connectivity	RL73K1ER82JTD	1	0.14
5	Molex mini fit	Molex	39013063	1	0.48
6	Force Sensor	SparkFun Electronics	SEN-09673	1	7.50
7	IR Sensor	Texas Instruments	TMCS1108A4UQDR	1	2.50
8	Buttons	C/K	PTS645TM43-2 LFS	2	0.26
9	Op-amps	Texas Instruments	595-TL972IP	1	0.95
10	Labor (Team)	N/A	N/A	3	26,252.10
11	Labor (Machine Shop)	N/A	N/A	1	1,056.00
12	TOTAL				27,469.97

3.2 Schedule

Table 2: Schedule overview corresponding to weeks and responsibilities.

Week	Objectives	Individual
2/20	Begin PCB design	Everyone
	Oversee hardware and software components	Ifesi
	Along with safe testing components like an Arduino, switches, LEDs, and a protoboard, new components for the machine shop, like a mountable button, are also included. Set up a debugging setup with switches to represent board inputs and LEDs to represent board outputs. At the machine shop, inquire. Examine PCB design with TA.	Matt
	Develop code for setup testing	Michael
2/27	PCB design conclusion	Everyone
	Ensure all parts needed are identified and ordered	Ifesi
	Follow up on machine shop and prototyping of the PCB	Matt
	Plan ordering of PCB as well as PCB schematic	Michael
3/6	PCB ordering	Everyone
	Ensure current PCB design satisfies all requirements	Ifesi
	Continue to follow up with machine shop on physical frame of machine	Matt
	Create gerber files for PCB and ensure audit is passed	Michael
3/13	Focus on control subsystem	Everyone
	Look over code and ensure it covers every possible state and case	Ifesi
	Plan system integration of PCB and physical component of machine	Matt
	Test finite state machine code on current PCB and prototype PCB	Michael
3/20	Test PCB setup via breadboard to avoid damage to parts	Everyone
	Make sure the PCB setup for outputs depending on debugging stimuli operates as the Arduino did with them, with any additional components needed for the desired outcome	Ifesi
	Search for possible improvements to current PCB design	Matt

Continued on next page

Table 2 – *Continued from previous page*

Week	Objectives	Individual
	Document all progress in functionality currently via testing	Michael
3/27	Door button, motor potentiometer sensor, motor current readings, IR sensor for can detection, and other inputs from the control subsystem are interfaced with sensors from the can-crushing subsystem.	Everyone
	Set threshold data for each component via testing	Ifesi
	Record performance statistics for machine	Matt
	Verify machine matches expected data and outcomes	Michael
4/10	Start on final report for project	Everyone
	Oversees technical descriptions	Ifesi
	Oversees diagrams and citations	Matt
	Oversees clerical work like statistics, orders	Michael
4/17	Final testing phase	Everyone
	In charge of final testing protocols	Ifesi
	Verify final tests	Matt
	Document test results in video	Michael
4/24	Final demo	Everyone
	Prepare for final demo and last minute changes to system	Ifesi
	Transport and handle final machine	Matt
	Assist in transportation and create presentation slides	Michael
5/1	Final Presentation and Report	Everyone
	Edit final presentation slides and report	Ifesi
	Edit final presentation slides and report	Matt
	Edit final presentation slides and report	Michael

4 Requirements and Verification

Outlined are modular tests to ensure the subsystems work as needed. These were individually verified before integrating the subsystems and again during the demo by the successful operation of Greencan.

4.1 Subsystem Verificatons

4.1.1 Can-crushing subsystem

Once in the can-crushing state, the opening through which one places the can must remain shut unless the motor stops moving the can-crushing platform (there is a button the door should hold down to tell if the door remains shut). The motor will only move the piston to crush the inserted can if the weight sensor beneath the can does not sense that the can weigh outside the acceptable 12 to 16 g range. Once crushed, gravity pulls the crushed can through an open chute which leads to the collector subsystem.
The motor's potentiometer reports the piston's location to the PCB, which, in turn, sends back control signals to the motor.
For safety reasons, the motor's current is also reported to the PCB. If the motor stalls for any reason (for example, squishing a small animal that crawled into the can-crushing space, crushing a human body part if the door was opened and the button's feedback to the control system is too late, or the motor is trying to crush an empty to pressurized can which could explode under compression, etc.) this creates a spike in the current drawn by a motor (stall current). This spike is noticed by the control system, which reverses the direction of the motor. Please note that the system states mentioned are displayed under the Software Design section.
The verification table is table 1.

4.1.2 Can-counting subsystem

The collector chute has an IR sensor that monitors when crushed cans fall into the collector below. If, for some reason, the chute becomes blocked (this could happen when the crushed cans pile too high or a crushed can do not fit down to the chute properly), a signal is sent to the control subsystem, which puts the system in a state where no more cans will be accepted until the system is serviced. Additionally, the IR sensor acts as a counter for the can-counting done by the PCB in the control subsystem.
The verification table is table 2.

4.1.3 Control subsystem

This comprises the PCB and the Arduino serial output terminal. Using inputs from the can-crushing subsystem (load cell weight, motor potentiometer, and the motor current readings), the PCB outputs control signals to control the direction of the motor. The button under the door of the can-crushing area also sends a signal to the PCB to stop the motos if the door is opening during can-crushing. The PCB, upon receiving a signal from the collector chute's IR sensor, increments the can count and displays this using the Arduino's serial output terminal. The verification table is table 3.

4.2 Quantiative Results

Overall, the operation of Greencan was a success. We verified this by testing our three main high-level requirements. The figures below show the test logs.

4.2.1 Safety Test Results

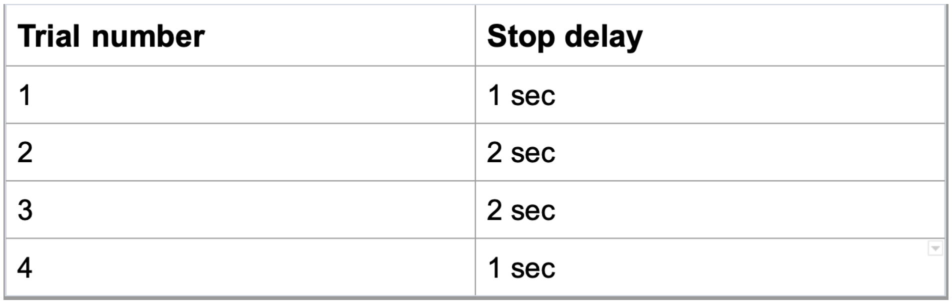
To ensure metal shards cannot reach immediate users while can crushing is in progress, we ensure the can crushing is paused within 5 seconds of opening the protective door of the can crushing area. In testing, opening the door paused can crush in the required time frame.

4.2.2 Can-crushing Test Results

The motor must drive the can-crushing platform to crush can be as thin as 2 inches (with an allowed error of ± 5%). In testing, this requirement is met as repetitive runs leave cans crushed to 2 inches of thickness.

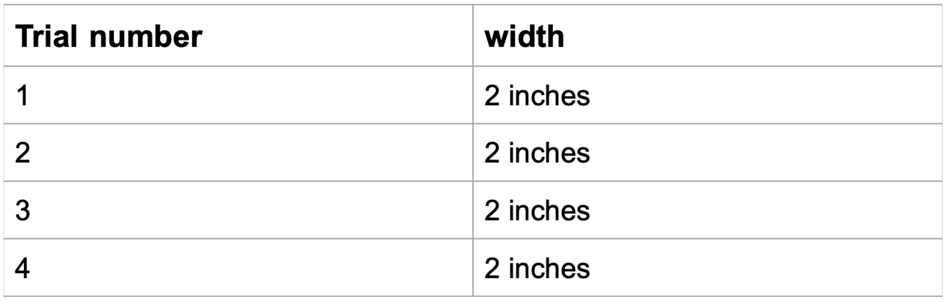
4.2.3 Can-counting Results

The system must accurately report the number of crushed cans. In testing, This requirement is also met because repeated runs show that the shown crushed can count correctly corresponds to the number of cans crushed.



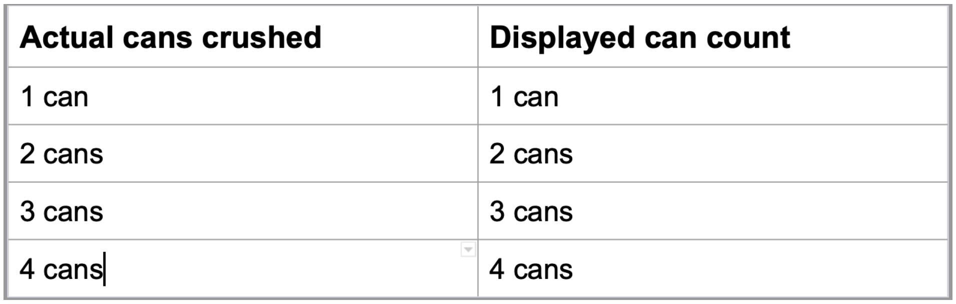
Trial number	Stop delay
1	1 sec
2	2 sec
3	2 sec
4	1 sec

Figure 16: Safety Test Results



Trial number	width
1	2 inches
2	2 inches
3	2 inches
4	2 inches

Figure 17: Can-crushing Test Results



Actual cans crushed	Displayed can count
1 can	1 can
2 cans	2 cans
3 cans	3 cans
4 cans	4 cans

Figure 18: Can-counting Results

Requirement	Verification test
Motor drives can-crushing platform to crush can be as thin as 2 inches (with an allowed error on ± 5%)	Any empty can fed into the can-crusher must be crushed thin enough for the crushed can to slide through the collection chute (which is 0.28 cm in width). Two empty cans will be fed to the machine. After the cans are crushed, they must fall through the chute to show they have been crushed to satisfaction.
Only empty cans are crushed with an accuracy of at least 90% percent– with our trial of 10 cans, at least 9 cans should set the machine into the expected state as outlined on the right.	Ten Aluminum cans (five completely empty, five full) will be individually fed into the machine. The crushing door remains closed once they are fed into the machine. Only the empty cans will be accepted and crushed; the other items will sit in the machine, not crushed, with the system staying in the REJECT state until they are removed.
Opening the door stops the motor from its crushing motion within 5 seconds.	Two empty cans will be fed into the machine: one can's crushing will be interrupted before the motor starts crushing the can and the other's crushing will be interrupted while the can is being crushed. Both experiments should put the machine in REJECT state until the cans are removed. Both cans will then be placed back into the can-crushing unit and left to be crushed uninterrupted. Now, the cans should be completely crushed and removed.

Table 3: Can-crushing Requirements and Verifications

Requirement	**Verification test**
MBlockages in the collector chute are reported to the control system as a flag to stop accepting cans (within 5 seconds) till the blockage is removed. After the blockage is removed, the system must be able to accept cans within 5 seconds.	Blockages in the collector chute put the machine in SERVICE state until they are removed. A crushed can or small rock will then be placed into the collected chute to simulate an obstruction the chute might encounter when launched on organizations' campuses. The system must accept no more incoming cans and stay in the SERVICE state until the obstruction is removed.
Only empty cans are crushed with an accuracy of at least 90% percent– with our trial of 10 cans, at least 9 cans should set the machine into the expected state as outlined on the right.	Ten Aluminum cans (five completely empty, five full) will be individually fed into the machine. The crushing door remains closed once they are fed into the machine. Only the empty cans will be accepted and crushed; the other items will sit in the machine, not crushed, with the system staying in the REJECT state until they are removed.
Sends can-count increment signal to PCB every time a can is crushed and sent down the chute. This count increment must be made within 5 seconds.	Two empty cans, after being crushed and collected in the collection bin, must trigger the can counter to increment by exactly two units. The time between either can be dropping and they can count increment must be 30 seconds (we will check this with a timer).

Table 4: Can-counting Requirements and Verifications

Requirement	**Verification test**
Stop the motor when it is stalling within 5 seconds.	The control subsystem indirectly monitors the current of the motor. When an experimentally determined threshold is reached before a can is crushed, the system notes that the motor has encountered an object too hard to be an empty, recyclable can. This item must be rejected. A rock, water bottle, and empty can will be put into the crushing enclosure, and only the empty can be crushed: the motor must retract within 30 seconds of trying to crush the rock and water bottle, sending the system into REJECT state.
Keeps count of how many cans have been crushed between service sessions, with an accuracy of 100% (assuming only cans are fed in).	After being crushed and collected in the collection bin, two empty cans must trigger the can counter to increment by exactly two units. This data must remain visible on the display until a SERVICE state is reached.
Does not allow the motor to run if the load cell detects full cans in can-crushing space. The detection of full cans must be 100% accurate	An empty can and full can/rock will be placed in the can-crushing container. Only the empty can be crushed; the rock/full can be rejected by the machine and rejected.

Table 5: Control Requirements and Verifications

5 Conclusion

In conclusion, we were able to accomplish the main objective of our project which was to provide a safe means of crushing cans in order to make recycling easier and more efficient in recycling organizations and the project was a huge success. The reason we can confidently say this is because all three of our high level requirements were met. Our system crushes only non-pressurized empty cans to prevent spillage and damage to the machine if someone was to mistakenly or mischievously try to crush a non-empty can filled with liquid, the system also gathers the cans it crushes in a collector bin by crushing them to a certain thickness that allows them to fall through the collector chute right into the bin, and finally the system keeps record of the number of cans that have been crushed between service cycles.

As for as uncertainties go, the only unsatisfactory result we came across in the course of testing our machine was the sensitivity of the load cell we originally planned to use to differentiate between empty and full cans. The load cell read weights with errors in multiple 100 grams which would not fly when trying to distinguish weights of less than 20 grams. As already mentioned earlier, we switched this out for the force sensor and received satisfactory results with that because it was by far more sensitive with errors of less than 1 gram. The testing of the other parts of the project went smoothly as planned.

The impact of project can be easily overlooked because as we know individuals are not too keen on recycling. Environmentally, our system promotes recycling which is the most important thing and motive behind the whole project, it also reduces waste by helping project the volume of waste sent to landfills and other waste management facilities. Recycling aluminum cans also conserves natural resources which are used in the production of aluminum. Other meaningful impacts include job creation since the project has the potential to create new employment opportunities in the recycling sector, fostering a sense of community an collective responsibility for the environment and raising public awareness.

Further can be done to enhance the performance and capabilities of our project. This would include developing a more advanced sensor technology to improve the detection of the object placed on the can platform since our current work is on the assumption that only cans will be placed on this platform, either empty or non-empty. Additionally, incorporating energy efficient mechanisms and sustainable materials in the design and construction of the machine can help minimize its environmental impact and promote sustenance.

Our project is quite straightforward from an ethics and safety perspective. For one thing, ethically speaking, this is a device designed to make recycling cans easier and safer, and to encourage recycling by showing how many cans have been recycled in it, so it could generally be considered a social good. Meanwhile, from a safety perspective, the main feature of our project, aside from counting cans, is that it takes multiple safety countermeasures, including preventing people from injuring themselves or the machine through intentional or unintentional misuse. During the project, we may be working with reasonably high currents to effectively crush the cans, and also will not have implemented all the safety features before building the basic crushing mechanism, so we will need to be careful that none of us injure ourselves during the testing process. This includes not accidentally crushing our hands or accidentally crushing things that shouldn't be crushed while testing the can rejection system After the project. As far as intentional misuse of our finished project goes, there are some things we won't be able to prevent, like someone putting a mouse in the can-crushing machine, but for the most part, our project protects against most forms of misuse. The IEEE and ACM ethical guidelines, while important, are not particularly relevant to our project.[2] [3] That being said, I am very certain that our group members have upheld, continue to uphold, and will continue to uphold these codes in their academic and professional careers. The only two principles which I think are specifically useful to our project are 1.1 and 1.2 of the ACM ethical guidelines. By making recycling easier, we are contributing to society and human well-being, as recycling is a key part of both making society cleaner and more efficient, and thereby more pleasant for human beings to live in. The best way to avoid ethical breaches is to ensure that people are familiar with why they need to avoid them. There are numerous ways to do this, including making people aware of negative consequences for them practically or from a long-term awareness of what they'll be missing out on through not being ethical. What we believe will, in particular, ensure the cooperation of our ethical standards team is the fact that we are all too hard working and earnest to ever resort to underhanded or dishonest tactics. This is demonstrably true given our effort to complete all assignments as soon as they have come out, and consistently stay ahead of schedule. Simply put, we have no need to be dishonest or unethical. We are also strong proponents of personal accountability and will continue to make sure that we stay on a good track. The United States Department of Labor's Occupational Safety and Health Administration (OSHA), lists moving parts and unexpected machine startup as two of the three leading issues for workers in recycling of metals, the third being lead, which is not relevant to this project, given that we intend to recycle aluminum cans, and a lead can would most likely be so heavy that it would trip our system anyway.[4] Plainly, OSHA considers amputation from errors in machine use paired with unintuitive machine functions to be significant concerns for its recycling process, meaning that if anything our project will be an asset towards popularizing recycling. We could not find any relevant federal regulations for our project. We checked the website of the Environmental Protection Agency (EPA) to find that there were no significant concerns surrounding the disposal of crushed aluminum cans. As long as the aluminum is sent to a scrapyard or collection center after being harvested, there should be no concerns surrounding weighting, crushing, and then counting the cans.[5] As far as industry standards are concerned, we could also not find any which were particularly relevant, even after consulting the website of The Aluminum Association. The main knowledge we gained is that recycling aluminum saves almost all of the energy spent making new aluminum, 95% to be exact, and that if all aluminum soda cans were recycled instead of going to landfills, we could save almost a billion dollars for the US economy. This also helps to support our current Aluminum production, as only 20% of the aluminum already produced isn't recycled aluminum, so if we could just recycle a little more we could have 0 net aluminum waste. [6] For campus policy, we checked the website for the Facilities and Services (F&S) Waste Management Department, to be informed the aluminum cans are highly recyclable by a facility we have on campus, meaning that the addition of production quality GreenCan units, to various school buildings or campus areas would probably be a great idea that would positively impact the school's reputation as a net 0 campus.[7] Given that our project is primarily based around ensuring safety, it is unlikely that our end product will present significant safety concerns. However, the safety concerns regarding the moving motor possibly hurting users is handled by our system since a sensor detects if the door is open during the can-crushing state and stops the motor.

References

[1] National Association of Convenience Stores. “The Value of Can and Bottle Recycling.” (2023), [Online]. Available: https://www.convenience.org/Topics/Sustainability/Can-Bottle-Recycling/Can-Bottle-Recycling (visited on 01/30/2023).

[2] Institute of Electrical and Electronics Engineers. “7.8 IEEE Code of Ethics.” (2008), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/28/2023).

[3] Association of Computing Machinery. “ACM Code of Ethics and Professional Conduct.” (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/28/2023).

[4] Occupational Safety and Health Administration. “Green Job Hazards - Recycling: Waste Management and Recycling — Occupational Safety and Health Administration.” (2008), [Online]. Available: https://www.osha.gov/green-jobs/recycling/waste-management (visited on 03/28/2023).

[5] United States Environmental Protection Agency. “Regulatory Exclusions and Alternative Standards for the Recycling of Materials, Solid Wastes and Hazardous Wastes.” (2023), [Online]. Available: https://www.epa.gov/hw/regulatory-exclusions-and-alternative-standards-recycling-materials-solid-wastes-and-hazardous (visited on 03/28/2023).

[6] The Aluminum Association. “Sustainability – Recycling — Aluminum Association.” (2023), [Online]. Available: https://www.aluminum.org/Recycling (visited on 03/28/2023).

[7] University of Illinois at Urbana Champaign. “Have You Bin Recycling?” (2018), [Online]. Available: https://archive.fs.illinois.edu/services/waste-management-and-recycling (visited on 03/28/2023).