UNIVERSITY OF TORONTO

INSTITUTE FOR AEROSPACE STUDIES

4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

Battery Dispensing & Sorting Machine

prepared by

Team 01 – Monday

Mohamad Khalil (1002412659)
Victoria (Jin) Cheng (1002547808)
Bryan de Bourbon (999686843)

prepared for

Prof. M.R. Emami

A technical report submitted for
AER201 – Engineering Design

TA: Allen Chee

April 13, 2017

Victoria (Jin) Cheng - 1002547808 | Mohamed Khalil - 1002412659 | Bryan de Bourbon - 999686843

AER201: Engineering Design

Battery Dispensing and Sorting Machine

Team 1 | PRA0101 | Teaching Assistant: Allen Chee | Prepared for: M. Reza Emami

i. Acknowledgements

This project would not have been possible without the support of various teaching staff, students, and mentors involved.

Our team would like to extend their unbounded gratitude to Professor Reza Emami for his patience and assistance during frustrating machine malfunctions, and for his motivational words during and outside of evaluations.

Thank you to our TA, Allen Chee, for his patience, motivation, and enthusiasm during lab hours and for his compassion and humanity through spontaneous malfunctions of our machine during evaluations. As well, thank you to Tim Teng and Geoff Donoghue, for guiding our team along with Allen and for lending us their circuits expertise.

Finally, thank you to every student in Engineering Science 1T9, with special mention to Anika Meng, Hunter Richards and Logan Rooks, for their extraordinary support and encouragement throughout term. It was great sharing a laboratory with all of you during the past 3 months, and your friendship and company provided us with the motivation to carry on during hard times.

Thank you all for your encouragement, support, and company over the past 4 months, and good luck with all you future endeavors.

— Team 1, AER201 Winter 2017

ii. Abstract

This document outlines the design details of a Battery Dispensing and Sorting Machine. This machine costs approximately $224 Canadian Dollars, and is intended to be used for the autonomous sorting of AA, C, and 9-Volt batteries, as well as the drained versions of each of these battery types. Up to 15 batteries can be sorted by this machine into their appropriate bins in under 3 minutes. This machine will reduce battery waste by encouraging the reuse of adequately charged batteries, impacting the environment in a positive way.

The machine will be composed of three subsystems: electromechanical, circuit, and microcontroller. The electromechanical portion is responsible for the structure and frame of the machine, as well as the mounting of all of the actuators and sensors required. The circuits portion is responsible for creating connections between the actuators and sensors and the microcontroller, as well as supplying adequate power to all of the circuits components. The microcontroller portion is responsible for designing the system logic and coordinating the actuators and sensors in order to receive the desired response from the machine. The microcontroller is also responsible for storing and displaying data about the machine's performances.

This machine sorts batteries based on their behaviour in a custom-designed voltage tester. Batteries are to be loaded in a funnel by the machine operator. From there, servo-driven Ethylene Vinyl Acetate wheels will feed the batteries from the funnel into a vertical column where they will be stacked. Each battery will then be sequentially placed in a voltage testing chamber that is capable of simultaneously determining the type of battery and whether that battery is drained or charged. The chamber then rotates and places batteries into their respective bins at the base of the machine using servo motors.

Future improvements can be made to the machine through slight shape adjustments in the neck of the funnel in the machine to prevent jamming, and through the development of a custom printed circuit board to remove the use of the DevBugger Board and reduce the cost of the machine.

iii. Table of Contents

iv. Table of Figures, Equations, and Tables

v. Team Members and Expertise

Victoria Cheng - Circuits and Sensors

Through innovation and design, I aspire to provide individuals with opportunities and improve their life experiences. I believe in using engineering to not only create interesting new technologies, but also to create user-friendly solutions where they are needed most. As an aspiring engineer, I plan to launch ideas focused on user-centered design, accessibility, and adaptability.

Bryan de Bourbon - Microcontroller and Programming

Background: Computer Science, Biochemical Sciences, Business, Engineering

Motivation: Inspired to simplify and solve the increasingly complex problems faced by humanity in the context of computer science (focus: artificial intelligence and information security), neurophysiology (focus: neural engineering), design, and pedagogy.

Mohamed Khalil - Electromechanical Design

My innovative, creative and logical problem solving skills allow me to thoroughly investigate and exhaust multiple solutions until an optimal one is converged upon. I am to create solutions that are environmentally friendly, and that is through the usage of biodegradable materials, and highly optimized mechanisms that require low power consumption. My proven expertise in mechanical design working on projects like remote-controlled aircraft enable me to create a machine that is not only efficient and compact, but also inherently intelligent, modular, and accessible.

vi. Symbols and Abbreviations

Charged: A battery is assumed charged if the voltage difference across its leads is at least 85% of its nominal rating; otherwise, it is drained.

Drained: See “Charged”

Operator: Individual who loads, starts, and stops the machine

Machine: Abbreviation for the Battery Sorting Machine

CAD: Computer Aided Design

FEA: Finite Element Analysis

SMA: Second Moment of Area

MOA: Moment Of Inertia

COM: Center Of Mass

1. Introduction

Figure 1.1 *Overall View of Final Machine*

1.1. Problem Statement

The Request for Proposal states that a supply company needs to sort a continuous stream of used and unused batteries based on their type and whether they are charged or drained[1]. Since the repetitive movements of the slow process of manual battery sorting can lead to fatigue-related chronic illness, and being in proximity to potential battery acid and other harsh chemicals can also damage employee health, the entire sorting process should be automated to maximize efficiency and employee safety[2], [3].

The batteries to be sorted will be of heavy duty ZnC or ZnCl or Alkaline batteries[3]. Once filled to a maximum of 15 randomly mixed batteries (of ZnC/ZnCl or non-rechargeable Alkaline) and instructed to start, the machine is expected to separate batteries into categories of charged AA batteries, charged C batteries, charged 9-volt batteries, and drained batteries. The batteries need to be sorted in 3 minutes with and the machine must display statistics about the batteries sorted. This allows for a record to be kept about the machine's operations, and the data can be used for analysis purposes at a later time. Sorting batteries allows them to be possibly reused in another product, lowering the number of batteries that are diverted to landfills and reducing the amount of toxic chemicals that are released into the surrounding ecosystems[3].

Especially in our growing technological world, responsible battery usage is important. The responsible re-use of manufactured batteries can be used as a method of reducing environmental footprints, which can reduce the environmental impact associated with battery production. [2]

1.2. Goal

The goal of this design was to construct the proof-of-concept prototype for a machine that can separate a continuous mix of used and unused batteries based on their type and charging condition, as outlined in the project proposal. [1]

1.3. Stakeholders

The following stakeholders will be impacted by the creation and development of the Battery-Dispensing & Sorting Machine:

1. **Battery Recycling Companies:** Companies that receive a large amount of unsorted batteries desire a way to quickly and safely sort the batteries.
2. **Employees of the Battery Recycling Companies:** The employees desire to have a safe work environment and to be free from contact with toxic chemicals.
3. **Design Team:** The team desires to satisfy the client requirements for the machine.
4. **Client (Professor M. Reza Emami & Allen Chee):** The course instructor and TA desire for the team so demonstrate understanding of the course material, as well as excellence teamwork and coordination during a thorough design process. [1]

1.4. Design Theory

The design team’s determined design values for this machine have been listed in order of decreasing important below, with Safety being the most important design value. These design values have been incorporated into the final prototype design, which will be further discussed in later sections.

1.4.1. Safety

Definition: Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. (MIL-STD-882E for systems safety) [8]

Justification: In order to prevent employee exposure to toxic chemicals and electrical shock, safety must be the utmost priority for this machine. Incorrect circuitry connection during the battery sorting process can cause sparks or flames that would release hazardous chemicals, damage the machine, and possibly injure the operator.

1.4.2. User-Friendly ("Operability")

Definition: Little time/effort is needed to set up and calibrate the machine, "the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human."[8]

Metrics: "quality of the interaction in terms of parameters such as time taken to perform tasks, number of errors made, and the time to become a competent user " [1]

Justification: In order to allow employees of a variety of skill levels to operate the machine, the machine interface must be easy to understand and use, and the loading procedure must be simple.

1.4.3. Compactness and Portability

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

Justification: In order to allow the client and the engineering team to test the machine is a variety of setting around campus during the design process, the machine must be portable enough to be carried easily. The supply company will also likely have multiple sorting machines, and the machine not occupy excessive space. The machine must also be less than 5 kg, as specified in the RFP.

1.4.4. Durability (“Robustness”)

Definition: The machine can withstand wear over several uses and mechanical stress. functions consistently in a wide range of operating environments with a low failure rate. [1], [8]

1.4.5. Time-Efficiency

Definition: The completion of the sorting task using the minimal amount of time possible [1]

Justification: The machine must be able to complete the sorting task within the required 3-minute time frame. As well, since the client receives a constant stream of batteries to be sorted, the faster the machine can sort the batteries, the more time the client can save.

1.4.6. Modularity (Maintainability and Easy-to-Assemble)

Definition: Easy to assemble, parts can be replaced or repaired easily. [1]

Justification: In case of required repairs, the machine should be able to be repaired and maintained easily. Modularity will allowed for the machine to be easily disassembled and fixed, without the need to replace the entire machine. This also allows for the machine to be easily assembled during prototyping.

1.4.7. Affordability

Definition: The price to make and assemble the machine.

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

1.4.8. Elegance

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

1.5. Market History: Background and Survey

Market, Literature, and Patent surveys were done in preparation for the prototype design. Summaries of the findings of these surveys have been included below, along with an analysis of the findings in sections below.

1.5.1 Market Survey

The following two machines, the Optical Battery Sorter 600 and the Optical Battery Sorter 200 are currently used as the industry standard for battery sorting. Their sorting mechanisms and machine details have been outlined below.

1.5.1.1 Optical Battery Sorter (OPS) 600

Table 1.5.1.1: *Specifications of the battery sorting mechanism for OPS 600 [4]*

Overview	
	Sorting Rate: 8 batt/sec Size: 3.5m x 8.5m x 2.3m Weight: 250kg
Sorting Steps	
	Step 1: Batteries are dumped into a conveyor belt with guard rails brings the batteries to the top of the machine.
	Step 2: Batteries are separated into groups my opening ad and closing a trap door system at the end of the conveyer belt
	Step 3: Grouped batteries travel along a conveyer belt that tapers to the size of one battery, but still has wide guard rails. Conveyer belt is oriented downward
	Step 4: a small vertical spring loaded arm ensures the batteries are lined up single file as they pass through the conveyer belt
	Step 5: A picture is taken of each battery and computer vision is used to determine their charge chemistry and type.
	Step 6: A pressurized air gun shoots each battery off the conveyer belt and down a tube to the appropriate bin

1.5.1.2 Optical Battery Sorter 200

Table 1.5.1.2: *Specifications of the battery sorting mechanism for OPS 200 [5]*

Overview	
	Sorting Rate: 160 batt/min Size: 2.5m x 6.6m x 2.0m Weight: 200kg
Sorting Steps	
	Step 1: up to 200kg of batteries are loaded into a bin at the base of the device.
	Step 2: A conveyor belt with grooves and a guard rails brings the batteries to the top of the machine in groups of 5-10
	Step 3: The groups of batteries are dropped into a rotary bowl, which lines up the batteries and feeds them into the next part via centrifugal forces
	Step 4: a small lateral spring loaded arm ensures the batteries are lined up single file as they pass through the conveyer belt
	Step 5: A picture is taken of each battery and computer vision is used to determine their charge chemistry and type.
	Step 6: A pressurized air gun shoots each battery off the conveyer belt and down a tube to the appropriate bin

1.5.2 Literature Survey

The Robotic Arm Battery Sorter was found in the literature as a proposed method of battery sorting. The mechanisms that it uses have been analyzed below.

1.5.2.1 Robotic Arm Battery Sorter [6]

Figure 1.5.2.1: *The CAD for the robotic arm battery sorter*

The reseach paper describes a machine that uses a robotic arm with a clamp at the end which picks up an individual battery from the input bin and places it in between probes oriented vertically. Next depending on the voltage test results, the arm takes the battery to its appropriate bin depending on shape and charge.

1.5.3 Patent Survey

The following patent outlines a method of load testing the battery.

1.5.3.1 Canadian Patent Document 1196379 [6], [7]

Figure 1.5.3.1: *A Drawing from the patent describing methods of load testing the battery*

The battery voltage without is a load is not indicative of how the battery will actually perform in applications where it will be powering a load. Loading a battery may actually cause its measured voltage to drop significantly. Therefore simply measuring the voltage difference across the battery leads is not a reliable way to determine if a battery is dead or charged. In order to obtain a better measurement of the battery voltage, the battery must be first loaded (a simple resistor can serve this purpose), and then the voltage difference across the battery leads can be taken. This will allow for a better determination of whether or not a battery can supply adequate voltage for useful purposes (and therefore whether it is dead or charged).

1.5.4 Extensions and Applications to our machine

Table 1.5.4.1: *A comparison of various designs from the market, literature and patents*

Device	Advantages	Disadvantages	Inspiring Steps
OPS200	-Sorting Speed -Accuracy and detail of sorting categories -Streamlined process	-Large -Heavy -Reliance on Computer -Vision -Complexity to Manufacture -Many moving parts	**Step 2:** Storing groups of batteries before release as a dispensing mechanism **Step 3:** Narrowing channel to get stream of single batteries
OPS300			**Step 3:** Use centrifugal force to align the batteries into a central channel **Step 4:** Use light-spring loaded doors to get batteries in single stream
Robotic Arm Sorter	-Small -Versatile Motion	-Heavy -Complicated design	-Voltage Tester detects multiple batteries based on shape

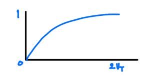
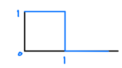
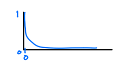
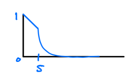
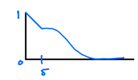
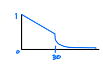
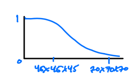
Various designs from the market, literature and patents were compared in Table 3.5.1. Some notable mechanisms that will be chosen for further extension in prototyping and brainstorming are noted in the "Inspiring Steps" column of Table 3.4.1. Some notable errors to avoid include over complication of the design as seen in the robot arm and the complex reliance on computer vision and several conveyer belts as seen in the OPS series solutions. ***There is clearly a gap in the market that can be addressed by our design.*** To be unique, our team intends to develop a solution that sorts the batteries quickly within a small and light volume, which has not been seen in the industry survey.

1.6. Objectives, Constraints, and Criteria

1.6.1. Quantifiable Objectives

The following objectives and constraints were chosen for the machine.

Table 1.6.1.1: *Objectives, Parameters, Scales, Units, Constraints, and Utility Functions*

Objective	Parameter	Scale	Unit	Constraints	Utility Function
Safety					
Electrical components do not burn out	Amount of voltage tolerance of circuit	interval	voltage	Must have at least a safety factor of 2 for expected voltage values in the circuit.	
Batteries are not damaged during sorting run	Amount of batteries damaged	ratio	n/a	No batteries must be damaged	
Stop button stops all motion immediately	amount of seconds to stop machine after button is pushed	ratio	seconds	machine must stop all motion immediately	
User-Friendliness					
The navigation menu should be self-explanatory.	Amount of time to transfer between states of the machine UI	ratio	seconds	Must not be more 5s	
Sorted battery containers should be easily removable and clearly identified	Amount of time remove all containers and return them to correct place	ratio	seconds	Must not be more 5s	
The machine should be easy to load batteries into	Amount of time required to load batteries	ratio	seconds	Must not be more than 30 seconds	
Compactness and Portability					
The machine should be small in size.	The volume occupied by the machine	ratio	cubic meters	Machine dimensions should be 0.45x0.45x0.45m^3	
The machine should be light in weight.	The weight of the entire machine	ratio	kilograms	Machine should not weigh more than 5kg	
Time-Efficiency					
The machine should sort batteries quickly	Time spent sorting one battery	ratio	seconds	Must sort at least 1 battery every 12 seconds	
Durability					
All joints maintain intended contact and orientation of all members	Number of joints that stay intact after operation or applied force	ratio	n/a	All joints must remain intact	

Machine materials do not buckle or yield easily	Noticeable structural displacement	ratio	mm	No more than 5mm displacement in any member	
Modularity					
The machine should have removable parts for repair	number of parts that can be removed from the frame vs number of parts in the machine	ratio	n/a	n/a	
Affordability					
The machine should be cost-affordable.	Cost of one machine	ratio	Canadian Dollars	Must not be more than $230 CAD before shipping and taxes	
Elegance					
The machine should operate smoothly.	Maximum amplitude of oscillation during battery sorting	ratio	millimeters	Must not exceed 2mm	
The machine should operate quiety.	Maximum level of noise generated during battery sorting.	ratio	decibels	Must not exceed 30Db.M	

1.6.2. Prototyping Process

1.6.2.1. Prototype 1 (A)

The first iteration of the prototype was made using foam board, wood, and plastic containers. A photograph of the prototype has been shown in Figure 1.6.2.1.1 and a sketch of its intended function has been provided in Figure 1.6.2.1.2. It was comprised of a spinning tube that fed batteries into a more narrow tube. The batteries were then pushed down into a voltage testing chamber where the battery charge and type was determined. The battery was then sent down a ramp that filtered the battery based on its shape and dropped it into the appropriate bin.

Figure 1.6.2.1.1. *First Prototype*

Figure 1.6.2.1.2. *Sketch of First Prototype*

This prototype presented issue with space constraints, and did not sort the batteries at a fast enough pace to meet the requirements of the RFP. Therefore, future iterations of the prototype explored ways to reduce the size and speed of sorting of the machine.

1.6.2.2. Prototype 2 (B)

Figure 1.6.2.2.1. *Multi-Path Servo Separation Mechanism*

Prototype 2 was similar to Prototype 1, with the ramp filtering mechanism replaced by the multi-path ramp with 2 servo motors shown in Figure 1.6.2.2.1. This was in attempt to reduce the amount of time required to sort the batteries. It was able to reduce the the time slightly, but introduce the issue of the batteries becoming stuck while traveling through the mechanism, due to the small gaps between the servos and the wall, which were found to be very difficult to resolve geometrically. This design also had negligible effect on the size of the machine, so other options had to be explored in Prototype 3.

1.6.2.3. Prototype 3 (C)

Figure 1.6.2.3.1. *Full Machine of Prototype 3*

The third iteration of the prototype was the basis for the final design, which is shown in FIgure 1.6.2.3.1. The internal mechanisms of the machine in this prototype have been discussed in greater detail in Section 5. The third iteration of prototyping reduced the volume occupancy of the machine, but required a mechanism for the voltage testing chamber to allow for the battery inside the voltage testing chamber to access all the battery containers. A rotation mechanism was determined to be suitable for this purpose. The initial design rotated the battery containers around the voltage testing chamber, similar to a carousel. However, due to the fact that rotating containers may injure the user if the user attempted to remove the battery containers during the sorting operation,this was determined to be less safe and less user-friendly than rotating the voltage testing chamber instead.

1.6.3. Analytical Hierarchy Method for Choosing Final Prototype

Table 1.6.3.1: *The inputs of the different prototypes to for the AHP, where Prototype 1 is A, Prototype 2 is B, Prototype 3 is C.*

	A	B	C
Safety (CI =0.009 , CR =0.02)			
A	1	1/2	1/4
B	2	1	1/3
C	4	3	1
User-Friendliness (CI = , CR =)			
A	1	1/2	1/2
B	2	1	1
C	2	1	1
Compactness and Portability (CI =0.0001 , CR =0.000001)			
A	1	1/3	1/5

B	3	1	1/4
C	5	4	1
Time-Efficiency (CI =0.042 , CR =0.01)			
A	1	4	2
B	1/4	1	2
C	1/2	1/2	1
Durability (CI =0.1 , CR =0.04)			
A	1	1/2	1/8
B	2	1	1/3
C	8	3	1
Modularity (CI =0.004 , CR =0.00001)			
A	1	1/2	1/2
B	2	1	1
C	2	1	1
Affordability (CI =0.000001 , CR =0.00001)			
A	1	1/2	1/2
B	2	1	1
C	2	1	1
Elegance (CI = 0.0018, CR =0.00001)			
A	1	1/2	1/5
B	2	1	1/3
C	5	3	1

Table 1.6.3.2: *The outputs of the eigen analysis for the AHP*

Prototype	Total Preference Score	Normalized Preference
A (Prototype 1)	5.11	0.21
B (Prototype 2)	6.44	0.27
C (Prototype 3)	12.45	0.52

Therefore, it is clear that Prototype 3 should be chosen to prototype further. This was the prototype that was chosen for our final design.

1.7 Budget

Below is the budget for the final machine, split by the different machine components. Please note that in cases where a full package of items was purchased but not completely used in the final machine, the “Number of Items” portion contains a fraction amount, proportional to the amount of the package used in the final machine.

Links for the stores where the items were purchased have been included in the appendices.

Table 1.7.1: *Budget for Machine, split by subsystem and module*

Item Description	Item Code	Shop	Number of Items	Cost per Item	Total Cost
Development Board					
PIC DevBugger Development Board	N/A	AER201	1	$50.00	$50.00
Keyboard	N/A	AER201	1	$3.00	$3.00
RTC Chip and Coin Battery	N/A	AER201	1	$5.00	$5.00
20x4 LCD	N/A	Amazon	1	$6.19	$6.19
SODIAL(R) Red Mushroom Cap 1NO 1NC DPST Emergency Stop Push Button Switch AC 660V 10A	N/A	Amazon	1	$3.20	$3.20
PIC Pin Board					
10 Pcs 2x20 Pin 2.0mm Pitch Double Row IDC Pin Headers Connectors	N/A	Amazon	0.1	$6.97	$0.70
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
Arduino Nano Board					
XCSOURCE 5pcs Mini USB Nano V3.0 ATmega328P 5V 16M Micro Controller Board F Arduino TE359	N/A	Amazon	0.2	$25.99	$5.20
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
SODIAL(R) 50 Pcs 103 10K ohm 3296W Trim Pot Trimmer Potentiometer 25 Turn	N/A	Amazon	0.02	$5.53	$0.11
Power Board					
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
20pcs 5V 0.3 A Mini Size Black SPDT Slide Switch for Small DIY Power Electronic Projects	N/A	Amazon	0.3	$1.96	$0.59
Mr.Geeker 10 Pcs Male Power Adapter DC Barrel to Screw Plug Jack Connector 2.1 x 5.5MM	N/A	Amazon	0.1	$9.99	$1.00
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.01	$17.98	$0.18
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.02	$17.98	$0.36
1/4W 5% RESISTOR (10 PACK) 220 Ohms	RESIS-500025	Creatron Inc.	0.3	$0.25	$0.08
1/4W 5% RESISTOR (10 PACK) 10k Ohm Ohms	RESIS-500025	Creatron Inc.	0.7	$0.25	$0.18
Rectifier Board					

Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
1N914 - 100V 0.2A RECTIFIER DIODE (5 PACK)	DIODE-000914	Creatron Inc.	0.8	$0.35	$0.28
IC OPAMP GP 50KHZ RRO 8DIP	LMC6462BIN	DigiKey Inc.	4	$4.24	$16.96
1/4W 5% RESISTOR (10 PACK) 200k Ohm Ohms	RESIS-500025	Creatron Inc.	0.8	$0.25	$0.20
Voltage Divider Board					
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
1/4W 5% RESISTOR (10 PACK) 1k Ohms	RESIS-500025	Creatron Inc.	0.8	$0.25	$0.20
Limit Switch Board					
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
1/4W 5% RESISTOR (10 PACK) 220 Ohms	RESIS-500025	Creatron Inc.	0.1	$0.25	$0.03
1/4W 5% RESISTOR (10 PACK) 10k Ohms	RESIS-500025	Creatron Inc.	0.1	$0.25	$0.03
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.006666666667	$17.98	$0.12
LED Control Board					
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80
TRANS NPN DARL 60V 5A TO220AB	TIP120GOS-ND	Digikey Inc.	3	$0.82	$2.46
1/4W 5% RESISTOR (10 PACK) 10k Ohms	RESIS-500025	Creatron Inc.	0.3	$0.25	$0.08
10 Pcs DC 5V Coil 7A 240VAC 10A 125VAC/28VDC 5 Pins SPST Power Relay JQC-3F	N/A	Amazon	0.3	$4.94	$1.48
Mr.Geeker 10 Pcs Male Power Adapter DC Barrel to Screw Plug Jack Connector 2.1 x 5.5MM	N/A	Amazon	0.1	$9.99	$1.00
12 inch RGB LED Strip	N/A	Creatron	2	$5.99	$11.98
Actuators					
9g Continuous Rotation Micro Servo	RB-Fit-02	RobotShop	4	$6.65	$26.60
RioRand® 5PCS x SG90 Micro 9g Servo For RC Airplane Car Boat Genuine	N/A	Amazon	0.6	$16.99	$10.19
SODIAL(R) Long Straight Hinge Lever 3 Pins Basic NO NC Momentary Micro Switch 2 Pcs	N/A	Amazon	0.5	$2.10	$1.05
VS-19 Pico Linear Servo	RB-Sbo-109	RobotShop	1	$12.95	$12.95
Stuctural Material					
8X24X.025 Aluminum Sheet Metal	142-550	Home Depot	1	$13.28	$13.28
1/4 inch x 2 Feet x 2 Feet Birch Plywood Handy Panel	$621,615.00	Home Depot	1	$5.68	$5.68
Flexible Wood (from Home Hardware)	N/A	Home Hardware	1	$3.68	$3.68
Small Acrylic Sheet	N/A	Home Depot	2	$2.00	$4.00
2-1/2 Inch Zinc Broad Hinge	859-110	Home Depot	2	$1.98	$3.96
4-40 x 3/8 Inch Phillips Truss Head Machine Screws Fasteners 50 Pcs	N/A	Amazon	0.8	$5.49	$4.39
SODIAL(R) Metric M3x0.5mm Stainless Steel Finished Hex Nut Silver Tone 50pcs	N/A	Amazon	0.8	$2.09	$1.67
SODIAL(R) 20mm x 15mm Metal Corner Brace Joint Right Angle Bracket Silver Tone 20Pcs	N/A	Amazon	1	$2.85	$2.85
PVC Pipe	CPLG-100	Home Depot	1	$0.77	$0.77

Miscellaneous Components					
1x40 Pins Male 2.54 mm Pitch Single Row Pin Header Strip 10 Pcs	N/A	Amazon	0.1	$2.36	$0.24
10 Pcs 1x40 Pin 2.54mm Pitch Straight Single Row PCB Female Pin Headers	N/A	Amazon	0.2	$2.78	$0.56
Jumper Wires Premium 6" F / F Pack of 20	RB-Ada-170	RobotShop	1	$2.60	$2.60
PK-Power AC Adapter for DVE Switching Model DSA-24CA-05 050400 5V 4A Power Supply	N/A	Amazon	1	$7.82	$7.82
HDE US Plug AC100-240V to DC 12V 2A Power Supply for 3528 Flexible LED Light Strips	N/A	Amazon	1	$3.95	$3.95
Extension Cord	N/A	Amazon	1	$2.50	$2.50
Total					$224.92

2. Problem Breakdown

The required functions of the machine were first decomposed using a functional flowchart (Figure 2.1).

Figure 2.1: *Functional Flowchart of the required functions of the machine*

The problem was divided into 3 main subproblems that were each assigned to be solved by one module of the machine. These subproblems where to feed batteries one-by-one into the voltage testing chamber, measure the voltage of the battery, and to place the battery in the appropriate bin, which are handled by the top, center, and bottom modules respectively.

2.1 Top Module: Feed Batteries One-by-One into the Voltage Testing Chamber

The top module addresses the problem of aligning the batteries to be fed into the voltage testing chamber one-by-one. In order to accurately measure the voltage of each battery, the voltage testing chamber must be only fed one battery at a time. When the batteries are loaded into the machine, they are placed by the machine operator in a pile into the machine without any specific orientation. The machine must be able to align the batteries from their random orientation.

The top module of the machine begins when the battery enter the loading funnel, and ends when the battery exits the orientation tube to fall into the voltage testing chamber. The processes that occur in this module have been depicted in Figure 2.1.

Figure 2.1 .1:*Top Module Process Flow Chart*

2.2 Center Module: Battery Voltage Measurement

In order to sort the batteries based on their voltage, the testing chamber must be able to determine what type of battery it is and what the voltage of the battery is.

The center module includes the voltage testing chamber and the rotating column that it sits on top of. The processes that occur in this module have been depicted in Figure 2.2.

Figure 2.2.1: *Center Module Process Flow Chart*

2.3 Bottom Module: Place Battery in Appropriate Bin

Once the type of battery and the battery voltage has been determined, the battery must be placed in the appropriate container.

The bottom module includes the base, structure, and the battery containers. The processes that occur in this module have been depicted in Figure 2.3.

Figure 2.3.1: *Bottom Module Process Flow Chart*

2.4 Division of Problem

A brief summary of the roles of each member of the group is included below. Each member's role is discussed in greater detail in their respective subsections.

Processing and Control (Microcontroller) [1]:

The microcontroller will be in charge of developing all the software for the system. This member of the team is required to use Arduino and the PIC Devbugger Board to coordinate the movements of the actuators in response to sensor signals. They are also responsible for developing the logic of the systems, as well as memory management and user interfacing.

Mechanism and Actuation (Electromechanical) [1]:

The electromechanical member is responsible for constructing the containers, structure and frames and incorporating whatever actuators and mechanisms are required in the system. Major components of the electromechanical subsystem can include: structure and frames, feeding mechanisms, sorting mechanisms, containers, and sensor and microcontroller mounts. This member is also responsible for mounting any required circuits or sensors to the frame and ensuring the durability of the machine.

Instrumentation and Interfacing (Circuit) [1]:

The circuits member is responsible for all the digital and analog interfacing electronics to connect the sensors and actuators to the microcontroller board. This member is responsible for communication between the microcontroller and the actuators, and for sending signals between the sensors and the microcontroller. In addition, this member shall complete wiring the machine and acquire suitable power supplies for the actuators, circuits, sensors and microcontrollers.

3. Individual Subsystems

3.1. Electromechanical Design Subsystem

This section describes the details of the robot's electromechanical design. The structure, drive systems/actuators, materials, and manufacturing process will be presented and analysed, with CAD models, simulations and calculations when applicable.

3.1.1. Assessment of the Problem

The RFP called for the design and development of an autonomous robot that can sort 9 volt, AA and C type batteries, based on shape and charge, for recycling purposes. From a Mechanical point of view, the main challenge is battery flow control and jam-prevention. These problems were used as the basis of the design, and the development process kept them in consideration at every step. Due to the small size of batteries, and the small size constraints presented in the RFP, manufacturing must be kept in the front lines of the design process, because a design that cannot be easily built is a not a feasible design.

3.1.1.1. Defining points of this subsystem:

1. Overall dimensions must comply with constraints presented by the client in the project's RFP.
2. Support the weight of the functional components, circuits, wiring, actuators and acrylic covers.
3. Absorb external impact loads, if any
4. Provide portability to make transportation and storage easier
5. Provide space to mount functional components
6. Provide easy access to all components for maintenance and update procedures
7. Provide a safe path for the batteries to flow, and ample storage space for post-sort dispensing.
8. Ensure the machine is safe to use, and minimize the chance of operator injury
9. Ensure the device does not present challenging manufacturing scenarios and procedures

3.1.1.2. Overall design philosophy

The robot is designed with a vertically oriented structure to maximize the use of gravity in the separation and sorting process. More specifically, the funnel-feeder system is the part where the batteries get ordered from a random pile to an orderly stack, awaiting sorting. The use of gravity will be crucial here, as the batteries must build up sufficient velocity in order to fall through the spring door loaded feeder that will ensure a vertical orientation. The structure is of a cylindrical shape, and falls within the reduced physical constraints as per the RFP, where the base diameter is 30cm, and the height is 45cm. The tall cylindrical structure gives the robot a very unique shape, and allows the machine to occupy a very small volume, while presenting ample space to mount all circuits, functional components, aesthetic components, power and signal wires, etc. The way this device works can be deduced simply by looking at and spending less than a minute analyzing the workflow inside, which is a very desirable characteristic, since it implies that the design is simple yet effective.

3.1.2. Relevant theory

3.1.2.1. Center of Mass

The COM of an object is the point of balance of the object, where the mass of the object is equally distributed around it. Knowledge of the COM is crucial for the balance of the robot, especially if internal agitators are present. In this case, since the structure is static, dynamic stabilization using center of mass is not applicable, however, a well balanced robot minimizes the chance of it tipping over. For a 1D discrete system of particles, the COM is mathematically defined as:

Equation 3.1.2.1.1: *Center of Mass for a 1D System[9]*

$$\overline{x} = \frac{1}{M} \sum_{i=1}^{n} m_i x_i$$

Where the system has n particles. For a continuous volume in 3D space, the COM is defined as:

Equation 3.1.2.1.2: *Center of Mass for a 3D system[9]*

3.1.2.2. Second Moment of Area

The SMA is one one of the governing parameters of flexural rigidity. It controls how much an object will bend under load, depending on the orientation of the object. It is defined as:

Equation 3.1.2.2.1: *Second Moment of Area[9]*

3.1.2.3. Stress and Strain

Stress is a measure of how much load is applied on a certain amount of material, and it is a criteria for material failure. It is defined as force per area, albeit that is a very naive definition. The definition of stress that is used by FEA software to perform simulations is the Cauchy Stress Tensor, defined as:

Equation 3.1.2.3.1: *Cauchy Stress Tensor[9]*

The software uses various numerical methods to discretize the material space into a suitable mesh, and calculates the stress tensor for every mesh element, which is then summed to produce the global stress results shown in the simulations. The stress representation is done using the von-mises stress method. However, the technical and mathematical details of this process are beyond the scope of this report.

Strain is a measure of deformation, and can be used a criteria of material failure. Deformations in mechanical systems, especially structures, are not desired, and the robot's structure must be designed in a way to minimize deformations, to prevent mechanical failures.

3.1.2.4. Torque and moment of inertia

Torque is the quantity that is used to evaluate motor performance and systems power consumption. It is a measure of a motor's power output, and it is very important to ensure that a motor can generate enough torque to drive the load, which has a specific moment of inertia. Moment of inertia is a measure of mechanical stress that a load puts on a motor. Moment of inertia of various components will be measured numerically using solidworks, as it is much easier and faster, and provides very high accuracy and reliability.

3.1.2.5. Buckling

Buckling is a mathematical instability that leads to a failure mode. When a structure is subjected to compressive stress, buckling will occur under certain conditions, and it is characterized by a sudden sideways deflection of a structural member. This may occur even though the stresses that develop in the structure are well below the materials critical failure stress values, making buckling a tricky mode of failure to deal with. As an applied load is increased on a member, such as a column, it will ultimately become large enough to cause the member to become unstable and therefore buckle. Further loading will cause significant and somewhat unpredictable deformations, possibly leading to complete loss of the member's load-carrying capacity. The force required to buckle a structural member can be calculated using the euler critical load formula, however, under certain slenderness conditions, the equation is inaccurate, and Johnson's Parabola must be used. Johnson's Parabola is an empirically derived relation that governs the buckling of a relatively low slenderness member.

Equation 3.1.2.5.1: *Euler Buckling Load Equation[9]*

Equation 3.1.2.5.2: *Johnson's Parabola[9]*

Equation 3.1.2.5.3: *Radius of Gyration*

The radius of gyration is defined as[9]:

The critical slenderness ratio is [9]:

If the member's slenderness ratio is less than the critical slenderness ratio, then the johnson parabola must be used.

Figure 3.1.2.5.1: *Column Buckling Conditions [10]*

3.1.3. Technical specifications of Solution

3.1.3.1. Materials selection analysis

3.1.3.1.1 Optimizing the Stiffness-Mass characteristics

The material parameters of the mass function must be optimized in order to provide the best balance between stiffness and mass. The mass function can be written as m = ρV = ρ(L)b^2. The results of calculations of these characteristics is included below:

It is clear that in order to optimize the Stiffness-Mass characteristics of the beam, we are required to minimize [$\rho/\sqrt{E}$]

3.1.3.1.2 Analysing materials based on Stiffness-Mass-Cost optimization

To induce cost into the equation, mass is multiplied by a certain cost index that is specific to the material.

3.1.3.1.3 Analyzing materials based on Structural failure in bending

Analyzing structural failure required the use of a force function that defines the relationship between geometric/material parameters to structural failure loads. The failure load for a square cross section cantilever beam with a single point-load applied to its tip is:

Table 3.1.3.1.3.1: *Table of calculation results for stiffness and mass[11]*

3.1.3.2 Main Load Bearing Structure

The support frame is the central load bearing structure of the robot. It is responsible for robustness and durability, and allows for the integration of all the other components, including circuitry and microcontroller units, into one operational unit. It is made up of four vertical members that will absorb the mechanical loads of the entire robot, and bracing members to stiffen the structure.

The vertical members are 45 x 5.08 cm and 3.175mm thick, and are constructed out of 3-layer birch plywood. They span the entire height of the robot, and they are the main load bearing members. Note that one of the members is a two-part component, composed of a vertical and a diagonal member, designed to optimize the mounting of the PIC microcontroller and the support of the main loading funnel.

The four-vertical member configuration was selected because it enables the robot to inherently allow gravity to aid the battery separation process, offers a great deal of flexibility in terms of mounting space for other components, great cable management, and is structurally strong. It also allows for very easy mounting of the cylindrical plexiglass case. Because they are constructed out of **3.175 mm** thick birch plywood, the vertical members will be very stiff, having young's modulus between **4.5 and 6.5 Gpa**[11], and second moment of area equal to **38,576 cm^4** along the lateral axis) and resistant to buckling. Birch plywood is also very light, with a density of about **0.7 g/cc**[11], which is significantly lower than the two other alternatives explored for use (aluminum and plexiglass). Since the weight of the entire robot is bound by 5 kg, the maximum loads that the members will experience will be no where close to those that risk yielding and structural failure. This hypothesis is confirmed through a simple static simulation in solidworks, where a part of the full structure is put under a **50N load** applied directly from above, and assuming that the bases of the members are fixed and that the material is **100% isotropic**, where the maximum von-mises stress recorded was **1.65 Mpa** (compared to **4.5-6.5 Gpa** strength)[11], and the maximum displacement was **0.195 mm**. It is worth noting that the loads in the real robot will not be applied in the same manner, as they will be more spread out along the length of the members. However, the results allow for a good estimation on the order of magnitude of the stresses experience, and based on the results, the plywood structure is perfectly capable of handling the maximum loads allowed for this design.

The simulation of the new structure involved correct bolt configurations for joints (most bolt symbols hidden for performance optimization), and the read deflection mode was visible through very high multiplicative exaggeration of the numerical results. The displacements were exaggerated by a factor of 251.22, and the real mode is shown in figure 3.2.3.1.2. It is clear that the braces designed indeed keep the structure from engaging into a dangerous deflection mode, which is not very clear in the older, unbolted simulation.

However, due to the limitations of the linear elastic model solidworks simulation uses, the bending modes of the members are not very realistic. This is because the model assumes perfect linear, homogeneous, isotropic materials, which is the case of many plastics and metals. However, wood is a fibrous material, which introduces its own challenges, because the mechanics of fibrous materials are vastly different from more continuous materials, especially the shear characteristics. However, the simulation provides a good enough order of magnitude estimation for the stress and displacement values, which can be used to validate the design.

Buckling was another design consideration, the calculations showed that approximately 11.5 Kg per member are required to buckle the structure.[9]

Figure 3.1.3.2.1: *Full FEA static load simulation in solidworks, showing the distribution of stresses (von mises, Mpa) with the old structure on the left and the new structure on the right.*

Figure 3.1.3.2.2*: Full FEA static load simulation in solidworks, showing local displacement (exaggerated) (mm) with the old structure on the left and the new structure on the right.*

***Figure 3.1.3.2.3**: Full FEA static load simulation in solidworks, showing true scale local displacement (mm) (new structure)*

The two component vertical member serves the purpose of creating a sturdy mount for the PIC microcontroller case and the user interface panel. It consists of a 20 cm lower vertical member connected to a 25 cm tall diagonal member. The lower vertical part is identical to the lower portion of the regular vertical members, and it contributes to connecting the lower base and the second bracing level (next section) together to the diagonal member. The diagonal component serves as a link from the second brace level to the top structural interconnection, and a mounting basis for the PIC microcontroller and the interface. The top structural interconnection serves as a mounting base for the battery intake funnel.

Although aluminum is much stronger than plywood, it is however about 4 times heavier, and more expensive. Considering the types of loads to be expected in this robot, it is far more reasonable to choose plywood as the construction material for the frame.

3.1.3.3 Brace members

The brace members are 21 x 2.5 cm and 3.175 mm thick, with the sides cut at 45 degree angles. They are responsible for stiffening the main structure by interconnecting the four vertical members. By creating a rigid interconnection between the vertical members at about one half of their length, their effective length is halved. This causes the force required to buckle the vertical members to increase by a factor of four, which greatly increases the strength of the structure. Theoretically, creating an inter-connection between the vertical members simulates fixing the members at their midpoint, which decreases its effective length. The braces also serve to create a strong base for the robot to sit on. Birch plywood was selected as the construction material for reasons mentioned in the previous section. Straight bracing members were chosen due to their ease of manufacturing. Initially, the design called for the use of quarter-circular members as braces, however, it was deemed that the extra complication is unnecessary, and straight members were adopted for the bracing structure. The straight brace members also provided extra space to attach hinged panels for mounting the central power board and the secondary actuator and voltage testing control circuits.

Figure 3.1.3.3.1: *Brace Member Formation*

3.1.3.4 Structural weight analysis

Using Solidworks, the total volume of the structure was precisely calculated to 331.92 cc. Using 0.7 g/cc as the volume of birch plywood, the total weight is 232.34 g. This is a very good weight for a structure of this size because it only contributes to about 4.6% of to maximum weight allowed for the robot under the reduced weight constraints. Note that when constructed, the volume and therefore weight will not be the exact theoretical weights calculated using solidworks, however, the difference will be very minimal (due to sanding, non-perfect sheet thickness, non-perfect cuts, etc).

3.1.3.5 Major Functional Components

3.1.3.5.1 Intake funnel

The intake funnel is a basic converging trapezoidal prism with a feeding throat. The intake measures 16 x 10 cm. The depth of the funnel is 7 cm deep, and converges to a 5 x 4 cm throat.

Figure 3.1.3.5.1.1: *Funnel of Machine*

The design of the funnel allows it to remain within the 30 cm diameter of the robot while presenting sufficient volume to load a maximum of 15 assorted batteries. The funnel must remain within the 30 cm diameter because it allows for the usage of only one standard size sheet of plexiglass for the construction of the outer case of the robot, since no cuts will be required.

From the initial iteration of the prototype, it was found that a large amount of batteries feeding into the funnel at a single time would clog the orientation tube and cause jamming. Therefore, it was determined that a feeding mechanism was necessary to control the number of batteries exiting the funnel throughout the sorting operation. A dual wheel feeding mechanism is built into the funnel to easily and reliably dispense one to two batteries simultaneously. The wheels will be driven using two FEETEC FS90 continuous servo motors, at an angular speed of one rotation per second. This ensures that the servos will be dispensed at a rate fast enough for the feeder (next section) to handle. The wheels will allow for a passage space that is the equal to the diameter of a AA battery. They will be made out of a soft and compressible insulation foam to allow for the larger batteries to push through. This was done to ensure that only one or two batteries will be dispensed at a time, and not overload the vertical feeder.

3.1.3.5.2. Vertical battery feeder

The vertical battery feeder is the device responsible for ensuring that the batteries align vertically and positions them for dispensing into the voltage testing chamber. The feeder is 15 cm long, and the cross section is 3 x 3 cm. These dimensions allow the batteries to comfortable fall down the feeder and minimize the risk of jamming. The intake portion of the feeder is 5 x 4 cm to match the outlet of the funnel.

Figure 3.1.3.5.2.1: *Feeder*

In order to prevent two AA batteries to fall simultaneously, a spring loaded plate is mounted on the inside of the feeder to ensure that only one AA battery can fall down at a time. The spring door acts as a delay mechanism, to delay the drop of the second AA battery, such that the final result is two AA batteries stacked on top of each other. The spring will be selected to allow a 9V and or C battery to open the door using their own weight. Such a spring has a spring constant of about 50 n/m. More experimentation will be performed to determine the optimal spring constant later on.

Figure 3.1.3.5.2.2: *Spring Door*

Another solution of a trap door was considered, however, Experimentally, the Trap Door method did not allow for precise control of the amount of batteries that fell out of the funnel while the doors were open. Reducing the period of time that the doors were open still caused multiple batteries to exit the funnel at one time, and caused batteries to be jammed in the door.

3.1.3.5.3. Voltage testing chamber operation mechanism

The voltage testing chamber has two degrees of freedom, one rotational about its vertical axis, and the other rotational about the servo drive axis. The arms where the axle and the servo are loaded on are constructed using wood, and the main tube is PVC. The entire chamber mechanism is rotated from the bottom with the use of a continuous servo, and the dumping mechanism is activated with the use of a position FS90 servo, connected directly to the rotational shaft of the chamber. The entire spinning mechanism has a moment of inertia of 596.13 g/cm^2 (calculated using solidworks). The drive servo has an output torque of 1.6 kg*cm, which is enough to safely drive the system.

Figure 3.1.3.5.3: *Voltage testing chamber and mount*

3.1.3.6 Actuators and Sensor Selection

3.1.3.6.1. Funnel Feeder Wheel Actuators

Table 3.1.3.6.1.1: *Actuator Considerations for Funnel Feeder Wheels*

Actuator Type	Sample Actuator used for Experimentation	Performance Analysis
Motor	Shenzhen DC Gearhead Motor (Straight)	• Too fast
	Shenzhen DC Gearhead Motor (Straight)	
Continuous Servo	Continuous Servo Motor (Project Kit)	• Met requirements
Position Servo	9g Continuous Rotation Micro Servo	• Similar to a double-trapdoor operation • Difficult to control how many batteries
	Position Servo Motor (Project Kit)	
	9g Micro Servo Motor (4.8V)	
Spring Door	Prototyped	• Difficult to control

From the initial iteration of the prototype, it was found that a large amount of batteries feeding into the funnel at a single time would clog the orientation tube and cause jamming. Therefore, it was determined that a feeding mechanism was necessary to control the number of batteries exiting the funnel throughout the sorting operation.

The desire to control the amount of batteries entering that Orientation Tube at one time directed the selection between the possible types of actuators for this action (listed in Table 3.2.4.1).

Linear servos, spring doors, and continuous servos were experimented with in a "trap door" type of method. Continuous servos and motors were experimented with as "geared waterwheels" that held the batteries between the rotating teeth.

Experimentally, the Trap Door method did not allow for precise control of the amount of batteries that fell out of the funnel while the doors were open. Reducing the period of time that the doors were open still caused multiple batteries to exit the funnel at one time, and caused batteries to be jammed in the door.

The Geared Waterwheel method was determined to be the applicable method in our device. It was tested with motors and with continuous servos. Motors did not provide the desired speed control that was required to control the feeding of the batteries. The continuous servo was able to provide this control, and along with the directional control that was also available with motors, without the need for an H-bridge.

Initially, rigid gears were experimented with on the continuous servo. The rigid gears also occasionally caused jams when batteries were stuck between the teeth. Therefore, it was decided that the teeth would be made out of a more flexible material that would not cause batteries to be caught between the teeth.

Grey insulation foam was selected to fulfill this purpose due to its ready-availabiity to the design team and its ease-of-machining.

Due to the control that the continuous servos offered over the flow of batteries into the orientation tube, the team was able to reduce the height of the orientation tube to accommodate 3 batteries, rather than the maximum of 15, and start and stop the continuous servos according to if the tube was full or empty. This procedure was able to drastically reduced the height of the machine.

The continuous servo used in experimentation was also larger than required, so smaller continuous motors were purchased for use in the machine.

3.1.3.6.2. Orientation Tube Status Sensors

In order to accommodate the sensing of whether the orientation tube is empty or full a sensor need to be included in the orientation tube to communicate this information. The possibilities considered have been listed in Table 3.2.4.2.

Table 3.1.3.6.2.1: *Sensor Considerations for Orientation Tube Status Sensors*

Actuator Type	Performance Analysis
Break Beam Sensor	• Varies under different lighting conditions
IR Sensor	• Varies under different lighting conditions
Ultrasonic Sensor	• Inaccurate around loud sounds
limit switch	• Inaccurate for measuring actual weight, but okay for measuring if something is there
Pushbutton	• Is not always pushed and gets in the way when trying to push the battery out of the tube

The break beam, IR and Ultrasonic sensors would have been placed within the tube in order to communicate the status of full/empty within the tube.

The calibration required for these 3 sensors in differing environmental conditions was deemed not user-friendly. The design team desired the machine to operating in various lighting and sound conditions without the need for recalibration each time. Therefore, pushbuttons and limit switches were considered as alternatives. These sensors would be placed at the bottom of each tube and communicate the full/empty status of the orientation tube based on the weight of the batteries.

Small push buttons were deemed an unreliable communication method during experimentation, as the surface area of the button was not large enough for the batteries to consistently hit. Even if a flat plate was placed on top of the button to increase the surface area, the batteries did not consistently push the button down, and the plate occasionally got stuck within the tube. Larger pushbuttons were also experimented with, but similar issues occurred with the consistency of the weight of the batteries holding down the button. Although the batteries often pushed the button after falling through the tube, once they were resting after the fall, the button was not consistently pushed by the weight of the battery. As well, when the batteries where to exit the orientation tube, they would be inhibited by the bevel of the pushbutton and become stuck in the tube.

limit switches were able to communicate fluctuations in weight in the tube. Although the limit switches did not accurately communicate the actual weight of the batteries, its communication of a binary status of whether or not there was a battery in the tube was reliable. limit switches were also available in sizes that fit the opening of the orientation tube and were able to stay flat against the bottom of the tube without inhibiting battery movement.

Therefore, it was determined that limit switches would be used to fulfill the sensor requirement in the orientation tube.

3.1.3.6.3. Orientation Tube Exit Actuators

Table 3.1.3.6.3.1: *Orientation Tube Exit Actuator Consideration*

Actuator Type	Performance Analysis
Trap Door	• Similar to Section 5.3.1.4.1
Linear Servo	• Not enough extension
Solenoid	• Not enough extension
Continuous Servo	• Applies force in a non-ideal orientation, but performs required purpose

In order for the batteries to exit the orientation tube, there actuators in the Table above were considered. It was desired for the batteries to fall out in a horizontal orientation in order for their voltage to be measured in the voltage testing chamber. Although vertical orientation was experimented with as well, it was determined that this orientation a reliable formation for the voltage testing chamber to catch the batteries in, and that this orientation occupied a greater amount of vertical space, which was extremely undesirable, as the machine was already near the height restriction of 0.45m.

The trap door method of exiting the orientation tube presented similar issues as discussed in above sections, and also introduced an unknown orientation to the fall of the battery. Pushing the battery out of a side panel hole with a lowered height was experimentally proven to produce a reliable horizontal battery orientation. Therefore, a pushing actuator was required. Linear servo motors and solenoids were researched, but it was found that the extension length of these actuators were too short, or the actuators exceeded the budget allotment of this project.

The design team was able to resolve the issue but employing a rotation actuator (a continuous servo) to perform the pushing motion. Although the force would not be applied in the ideal manner, this actuator was experimentally shown to perform the task reliably. The servo also allowed for control of the speed and frequency of battery exit, and had already been employed previously in the funnel wheels, so it reduce the amount of different parts required in the machine and reduced cost.

3.1.3.6.4. Voltage Testing Chamber Plate Actuators

Table 3.1.3.6.4: *Voltage Testing Chamber Plate Actuator Considerations*

Actuator Type	Performance Analysis
Linear Servo	• Adequate actuation distance
Solenoid	• Not enough actuation distance
Position Servo	• Mechanism is too complex and prone to jamming

After exiting the orientation tube, the batteries enter a voltage testing chamber where the voltages of the batteries are measured. In order to allow the batteries to fall out of the orientation tube with some amount of uncertainty, the testing plates of the voltage testing chamber needed to be initially larger than the size of a single battery, and be mounted on a mechanism that is able to move the plates closer once a battery is in the chamber. The actuators in the table above were considered candidates to perform this pushing motion.

Solenoids that were within the budget,weight, and size constraints of the machine were unable cover the distance uncertainty required for the chamber. Position Servos with additional mechanism attached were considered to add unnecessary additional complexity to the machine.

Linear servos were able to provide more simple solution that offered a greater amount of movement for less weight than solenoids.

Initially, the design comprised of 2 linear servos on either side of the voltage testing chamber. After experimentation, it was determined that 1 linear servo would be sufficient for the requirements of the machine. A spring mechanism will be included in the mobile voltage testing plate in order to account for any small deviations in standard battery height between the plates.

3.1.3.6.5. Voltage Testing Chamber Exit Actuator

Table 3.1.3.6.5.1: *Voltage Testing Chamber Exit Actuators*

Actuator Type	Performance Analysis
Trap Door	• Not enough control • Bevel required for hinge blocks batteries
Linear Servo	• Not enough actuation distance to push battery out of chamber
Solenoid	• Not enough actuation distance to push battery out of chamber
Continuous Servo	• Adequate control and fast speed

After testing the battery voltage, the battery must be removed from the voltage testing chamber and placed in the appropriate container. The voltage testing chamber must first align itself with the correct container. it was determined that the voltage testing chamber would rotate to the correct container. In order to rotate the chamber, the following actuators in Table 3.2.4.5 were considered.

In order to reduce the amount that the chamber needed to rotate to align itself, the chamber should allow for the battery to exit from either side. This reduced the maximum amount of rotation require from 135 degrees to 45 degrees (Shown in Figures 3.2.4.5.1 and 3.2.4.5.2). Therefore, the type of actuator selected must allow for the battery to exit either side of the chamber. Solenoids and linear servo motors were considered as pushing mechanisms to push the battery out of the chamber. However, the distance covered by these mechanisms was found to be insufficient for pushing the battery out of the chamber.

The side trap door and chamber rotation were considered in parallel as options, since they required the same type and amount of actuators (2 position servo motors). Experimentation showed that the chamber rotation was a more effective technique due to it existence of a slight bevel for the hinge required for the trap door.

The initial design of the chamber rotation had a symmetrical design with one servo motor on either end of the same, each supporting a different side. This was later changed to have one motor attached to the chamber by the shaft.

Figure 3.1.3.6.5.1: *Rotation degrees required for chamber where the battery can exit on either side*

Figure 3.1.3.6.5.2: *Additional rotational degrees required for a chamber where the battery can only exit on one side*

3.1.3.6.6. Voltage Testing Chamber Rotation Actuators

As discussed in Section 5.3.1.4.4. The voltage testing chamber will rotate in order to orient itself with the correct battery container. The mechanisms in the table below were considered as possible actuators for the required rotation motion.

Table 3.1.3.6.6.1*: Voltage Testing Chamber Rotation Actuators*

Actuator Type	Performance Analysis
Position Servo	Fast and accurate distance travel
Motors	Difficult to use for accurate distance travel
Continuous Servo	Not ideal to get chamber to specific degree

Experimentation with the actuators in the table above demonstrated that motors and continuous servos are unable to consistently provide an exact rotational degree. Position servo motors were selected because they are able to consistently turn to a predetermined degree and return to their original position.

This position servo motor is required to bear more weight than the servo motors used to tilt the voltage testing chamber, so a larger servo motor will be used.

3.1.3.7 Specific Actuators to be used

Datasheets for these actuators can be found in the Appendices.

3.1.3.7.1 Position FS90 servo[12]

This servo is extremely lightweight, powerful and durable. It weighs 9 grams, costs about 6$, and outputs 1.6 kg*cm of torque. It is very easy to work with because of its size and weight, and can perform a variety of different tasks. This servo was chosen for the voltage testing chamber because it will be very easy to mount and connect to the mechanical moving parts.

3.1.3.7.2 Continuous FS90 servo[13]

This is the continuous rotation version of the previous servo. It was chose to spin the wheels of the funnel because of its great speed control and steady torque output, combined with the aforementioned weight and price reasons.

3.1.3.7.3 VS-19 Pico linear servo[14]

This 13$ linear servo is small and lightweight (4 grams), and offers great actuation length (2 cm). This was chosen over solenoids because it is cheaper and much smaller and lighter.

3.1.4 Changes from Initial Proposal

3.1.4.1 Removal of Holes in Structural Supports

Initially, the main load bearing structural members were designed with holes that ran along their entire span. The main purpose of the holes was weight reduction, since the reduced physical constraints placed a 5 Kg upper bound on weight. Initially, the wood selected for use was 6.25mm thick, but due to availability and cost, 3.175mm thick wood was used instead. This halved the weight of the structure, and there was no longer a need for the member holes. A benefit of this decision was a huge time saving in the manufacturing phase, since no holes had to be machined. This change also allowed stress to be more uniformly distributed over the members, which improves structural stability and decreases the chance of mechanical failure. The changes in stress distributions were shown in the new static FEA simulations.

3.1.4.2 Additional Agitation Servo in Funnel

The agitation servo was a last minute addition to the vertical feeder in order to prevent jamming. The servo rotates an arm at a 20 degree angle from vertical inside the joint between the feeder and the funnel, which, in case of jamming, yanks the battery out of place, thus, unjamming it.

3.1.5 Suggestions and Improvements

3.1.5.1 Plastic Brackets instead of steel brackets

The usage of steel brackets ensures extremely sturdy connections between all members, however, it added significant weight to the machine. Plastic brackets can be found that are strong enough to perform as required yet contribute less to the weight.

3.1.5.2 More spacious feeder

A more spacious battery feeder would have reduced the chances of batteries jamming, and eliminated the need for an extra agitation servo. This would have cut down both cost and runtime, and increased the efficiency of the overall machine.

3.1.5.3 High friction wheels

The wheels used to suck the batteries into the feeder have just enough friction to be able to drive the batteries. However, if the batteries are somehow stuck in a certain way where they interlock, the wheels will not be able to break the batteries up, since the wheels will simply slip beneath the batteries. An improvement could the use of a thin rubber layer over the wheel to increase their grip, and thus ability to break the lock between batteries.

3.1.5.4 More powerful wheel drive servos

The servos used to drive the battery suction wheels were fitec FS90 continuous micro servos, with output torque of 1.6 kg*cm. Under certain circumstances, the servos will stall simply due to the mechanical resistance presented by the larger C type and 9 volt batteries, and they placed significant stress on the drive axles as they were driven down the funnel into the feeder. Alternative servos exist that can output about 3.7 kg*cm, which seems like a much safer drive system, since it would be much more difficult to stall.

3.2 Circuits Subsystem

3.2.1 Assessment of the Problem

The circuits subsystem is the interfacing system between the microcontroller subsystem and the electromechanical subsystem. The responsibilities of the circuits systems can be separated into 4 sections, as outlined below:

1. **Provide a constant and appropriate power supply to actuators, sensors, and microcontroller**
   The circuit subsystem must be able to supply constant voltage according to the demands of the various components in the circuits. The actuators and sensors in the circuit require 5V, and the microcontroller board and the LEDs required 12V. The circuitry must also meet the current demands of the components, calculations for which are shown in Section XX. The 5V power supply needed to supply a minimum of 2.86 A, and the 12V power supply needed to supply a minimum of 1.76 A.

2. **Allow for the emergency stop of the system that stop the supply of power to all actuators and sensors.**
   The systems needed an emergency stop button that could stop all sensors and actuators in the machine, as well as indicate to the microcontroller that the machine was now in an emergency state.

3. **Communicate and Transfer signals between different sensors, microcontrollers, and actuators.**
   The circuitry must carry the signals from the different components of the machine between the various circuit modules with minimal interference in order to the ensure the coordinated communication between machine components.
   The actuators must receive signals from the microcontroller and the sensors must output signals to the microcontroller. The sensors must be able to detect the location of a battery during the sorting and communicate that location to the microcontroller.
   The two microcontrollers used in this project were the Arduino Nano (Figure 3.2.1.1) and custom AER01 PIC DevBugger Board (Figure 3.2.1.1).

Figure 3.2.1.1: *PIC DevBugger Board and Arduino Nano Microcontroller[15]*

The actuators used in this project were continuous servo motors (Figure 3.2.1.2), positional servo motors (Figure 3.2.1.2), and linear servo motors (Figure 3.2.1.2).

Figure 3.2.1.2: *Continuous servo motors (left)[13], Positional servo motors (center)[12], [13], and linear servo motor (right)[14]*

The sensors used in this project were voltage testing pins (Figure 3.2.1.3) and a limit switch (Figure 3.2.1.3).

Figure 3.2.1.3: *Side view of voltage testing pins (left) and limit switch (right)*[14], [16]

4. **Measure the voltage of the batteries, scale the voltage to between 0-5V, and rectify any negative voltage.**

   The orientation of the batteries is undetermined, so the input the the voltage measurement circuit might be a negative voltage. The microcontroller is unable to measure negative voltage, so this voltage must be rectified before it can be sent to the microcontroller for analysis.

   The microcontroller can also only measure voltages between 0V and 5V, so the 9V battery measurements must be scaled down in order to accommodate that restriction.

3.2.2 Solution and Supporting Calculations

The main connections required from the circuit are depicted below. Detailed circuit diagrams are located in sections below. Datasheet for all circuit components can be found in the Appendices.

Figure 3.2.2.1: *Diagram of main circuit connections*

Figure 3.2.2.1 shows the major circuit component connections, which will each be discussed in further detail in the following sections. The microcontroller serves as the interface and signal receiver between these circuit components, and sends out instructions accordingly. The exact part numbers, specifications, and prices of each of the selected circuits components have been included in the Budget Section of this proposal.

The final circuit modules can be seen in Figure 3.2.2.2.

Figure 3.2.2.2: *Diagram of main circuit connections*

The final circuits provide 5V power, ground, and PWM signals from the microcontroller to 4 continuous servo motors, 3 positional servo motors, and 1 linear servo motor. It also provides power to the machine LEDs and rectifier Op Amps.

The communication circuitry sends and receives signals from a limit switch and a voltage testing chamber.

3.2.2.1 Arduino Nano Board

Figure 3.2.2.1.1: *Arduino Nano Board circuit (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.*

The Arduino Nano Board contains mounting pins for the Arduino Nano, communication connection pins and a potentiometer. A photograph of the board is shown in Figure 3.2.2.1.1, along with a diagram depicting detailed connections of the circuitry.

The Arduino Nano has an operating voltage of 5V and a power consumption of 19 mA, plus 40 mA for each additional I/O pin used. This project used 16 pins from the Arduino.

3.2.2.1.1 Arduino Nano Power Requirement Calculations

$$Voltage = 5V$$

$$Current = 19\ mA + (40\ mA \times 16\ pins) = 659\ mA$$

$$Power = Current \times Voltage = 5V \times 659\ mA = 3.295\ W$$

The calculated power requirement of the Arduino Nano is 3.295 W.

3.2.2.1.2 10k Potentiometer Power Circuit and Power Calculations

The 10k potentiometer is connected in the circuit as shown in Figure 3.2.2.1.2.

Figure 3.2.2.1.2: *10k potentiometer voltage divider circuit diagram, with Vin=5.17 V and Vout=5V. Vout goes to the REF pin of the Arduino Nano.* [17]

$$Voltage = 5V$$

$$Resistance = 10k\Omega$$

$$Current = \frac{Voltage}{Resistance} = \frac{5V}{10k\Omega} = 0.0005A$$

$$Power = \frac{(Voltage)^2}{Resistance} = \frac{(5V)^2}{10\ k\Omega} = 0.0025\ W$$

The calculated power requirement of the potentiometer is 0.0025 W.

3.2.2.2 Power Distribution Board

Figure 3.2.2.2.1: *Power Distribution Board (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.*

The power distribution board provides the required 5V power and ground to all of the servo motors, the Arduino, the limit switch, and the LED control circuit. A detailed diagram of the different components of the Power Distribution board is included in Figure 3.2.2.2.1.

The barrel jack at the bottom right corner of the board comes from a 5V power supply. The current requirements of this power circuit have been calculated in the section below, including the total power requirement.

3.2.2.3 Emergency Stop

The emergency Stop button is connected in the Power Board. Its circuit diagram is as shown in Figure XX. When it is pressed, it stops supplying power to the power rail of the power distribution board and instead supplies power to a signal to the PIC that indicates a state of emergency and an indicator LED.

Figure 3.2.2.3.1: *The emergency stop button circuit*

3.2.2.3.1 Emergency Stop Calculations

According to the data sheet, the green indicator LED has requirements of:

$$Voltage = 2.2V$$

$$Current = 8\ mA$$

$$LED\ Resistance = 275\ \Omega$$

$$Total\ Circuit\ Resistance = 275\ \Omega + 220\ \Omega = 495\Omega$$

$$Power = \frac{(Voltage)^2}{Resistance} = \frac{(5V)^2}{495\Omega} = 0.05\ W$$

The calculated power requirement of the LEDS is 0.05W. However, it should be noted that when this circuit is powered, the rest of the machine is not, so this power requirement is not in addition to the other power requirements of the machine.

3.2.2.4 Single Servo Motor Circuit

A single servo motor actuator is connected in the circuit as shown in Figure 3.2.2.4.1.

Figure 3.2.2.4.1: *The single servo motor circuit, with the PWM signal coming from the Arduino*

3.2.2.4.1 Positional Servo Motor Calculations

According to the data sheet, the positional servos used have operational requirements of[18]:

$$Voltage \ = \ 5V$$
$$Stall\ Current \ = \ 250mA$$
$$Total\ Current \ = \ 250mA \times 3 = 750mA$$
$$Power = \ (Current \ \times \ Voltage) \ \times 3\ motors \ = (5V \ \times 250\ mA) \times 3 = 3.75\ W$$

The calculated power requirement of the positional servo motor is 3.75W.

3.2.2.4.2 Linear Servo Motor Calculations

The linear servo used have operational requirements of[19] :

$$Voltage \ = \ 5V$$
$$Approximate\ Current\ Consumption = \ 250mA$$
$$Power = \ (Current \ \times \ Voltage) \ = (5V \ \times 250\ mA) = 1.25\ W$$

The calculated power requirement of the linear servo motor is 1.25W.

3.2.2.5 Double Servo Motor Circuit

The machine uses 6 servo motors, 2 pairs of which are dependent on the same PWM signal. These motors at the ones used in funnel and the orientation tube, which will be connected in opposite physical orientations on the funnel so that the rotation of the wheels force batteries down the funnel and unjam the batteries. These wheels must be able to turn in the same direction at the same speed, hence the decision of using the same PWM signal for both.

This circuit also allows 2 servo motors to be driven simultaneously with the same PWM signal, allowing the motions to spin at the same speed and same direction at the same time.

The power requirement calculations are the same as the requirements for a single servo motor, since power and ground are still supplied separately.

__Figure 3.2.2.5.1:__ The double servo motor circuit, with the PWM signal coming from the Arduino

3.2.2.5.1 Continuous Servo Motor Calculations

According to the data sheet, the positional servos used have operational requirements of[13]:

$$Voltage = 5V$$

$$Running\ Current = 250mA$$

$$Total\ Current = 250mA \times 4 = 1000mA$$

$$Power = (Current \times Voltage) \times 4\ motors = (5V \times 250\ mA) \times 4 = 5\ W$$

The calculated power requirement of the continuous servo motor is 5W.

3.2.2.6 Voltage Testing Chamber Circuit

As shown in Figure 3.2.2.6.1, the voltage testing chamber is a triangular prism with 2 triangular voltage testing plates at each end. Figure 3.2.2.6.1 shows the locations of the 4 voltage testing pins. A C battery will only touch the two C pins on either side of the chamber and an AA battery will only touch the two AA pins on either side of the chamber. A 9V battery will touch one of the 9V pins on one side of the chamber and the AA pin on the same side.

Figure 3.2.2.6.1: *One side of the voltage testing chamber, showing 4 of the input pins, and their corresponding battery geometries. The other side of the voltage testing chamber also has the same pin configuration.*

The voltage testing circuit takes inputs voltage measurements from the 8 pins of voltage testing chamber (Figure 3.2.2.6.1).

In order to determine which type of battery is in the chamber and the voltage of this battery, the following circuit configuration in Figure 3.2.2.6.2 has been designed. The inputs then go through the voltage testing circuit, which divides the voltage of each battery in two, in order to bring the 9V battery measurements within the 0-5V range of the microcontroller, while maintaining the proportionality of the battery voltages.

Figure 3.2.2.6.2 shows a photo of the voltage testing circuit, along with a labelled diagram of the input and output pins.

Figure 3.2.2.6.2: *Voltage Testing Circuit Board (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.*

Due to the configuration of the pins in the voltage testing chamber, which input pins are trigger by the battery will output a voltage signal to a specific output pin from this circuit. These output signals can be used to determine the type of battery. The output signal from the blue wire will indicate a 9V battery, an output signal from the purple wire will indicate a C battery, and an output signal from the orange wire will indicate a AA battery.

A charged 9V battery signal will be above 3.825V from one of the blue wires, since the voltage of the battery is divided in half. A charged C or AA battery will have a signal of above 0.6375 V.

The voltage divider circuit for each battery is shown in Figure 3.2.2.6.3. There are a total of 4 of these circuits on the voltage testing circuit board.

Figure 3.2.2.6.3: *Voltage Division Circuit*

3.2.2.6.1 Voltage Testing Circuit Power Calculations

This circuit receives no input power from the power supply, but power requirement calculations were done in order to ensure that the power ratings of the capacitors was sufficient.

9V Battery:

A fully charged 9V battery will sometimes have a voltage of around 10V for a short period of time. Therefore, this calculation will be done with 10V.

$$Voltage = 10V$$

$$Resistance = 2 \times 10k\Omega = 20k\Omega$$

$$Power = \frac{(Voltage)^2}{Resistance} = \frac{(10V)^2}{20\ k\Omega} = 0.005\ W$$

0.005W is less than 0.125W (¼ W), so the resistors selected for this circuit were appropriate.

AA and C batteries:

In order to ensure symmetric behavior, 10kΩ resistors were also used for the AA and C battery voltage divisions. Fully charged AA or C batteries may sometimes have 2V for a short period of time, so the calculations will be done with 2V.

$$Voltage = 2V$$

$$Resistance = 2 \times 10k\Omega = 20k\Omega$$

$$Power = \frac{(Voltage)^2}{Resistance} = \frac{(2V)^2}{20\ k\Omega} = 0.0001\ W$$

0.0001W is less than 0.125W (¼ W), so the resistors selected for this circuit were appropriate.

3.2.2.7 Rectifier Circuit

The rectifier circuit used op amps to rectifier voltage with minimal loss. The circuit used is depicted in FIgure 3.2.2.7.1, along with a diagram of all of the input and output pins.

Figure 3.2.2.7.1: *Rectifier Circuit Board (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.*

Each of the 4 rectifiers uses an LMC6462 Operational Amplifier Chip to rectify the voltage (Figure 3.2.2.7.2).

Figure 3.2.2.7.2: *Internal Diagram of an LM6462 [20]*

3.2.2.7.1 Rectifier Circuit Calculations

The rectifier circuits were connected as shown in Figure 3.2.2.7.1.1.

Figure 3.2.2.7.1.1: *Rectifier Circuit Diagram [21]*

The maximum allowed current at the power supply pin of the rectifier is 40mA.It was used at a voltage of 12V.

$$Voltage = 12V$$

$$Total\ Current = 40mA \times 4\ rectifiers = 160mA$$

$$Power = (Current \times Voltage) \times 4\ rectifiers = (12V \times 40\ mA) \times 4 = 1.92\ W$$

The maximum power consumption of the rectifier circuit from the 12V power source is 1.92 W.

3.2.2.8 LED Control Circuit

The RGB indicator LEDs on the machine (shown in Figure 3.2.2.8.1), change colour to indicate the state of the machine or the battery type being sorted. These LED strips must be powered with 12V, but the PIC and Arduino can only provide 5V. Therefore, the LED control circuit uses transistors and relays in order to use 5V signals from the PIC to control 12V signals of the LEDs.

Figure 3.2.2.8.1 shows a photograph of the LED control circuit and a diagram depicting the input and output pins.

Figure 3.2.2.8.1: *LED Control Circuit Board (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.[22]*

3.2.2.8.1 Transistor and Relay Calculations

Figure 3.2.2.8.1.1: *Transistor-Relay Circuit Diagram[22]*

Circuit Diagram for the transistor-relay connections, where Ein is the 5V signal from the PIC and Vcc is 5V. The NPN transistor used was the TIP 120. According to the data sheet, the current requirements for one relay is 66.7mA, and that each relay has a resistance of 75 Ω, a voltage rating of 5V, and a power consumption of 0.36W.[23]

The TIP120 has a Collector Current rating of 5A, which is much less than the required 66.7mA, so it is appropriate for usage in this circuit. 3 transistors and 3 relay were used in the circuit. The total current and Power requirements are included below.

$$Voltage\ Rating\ =\ 5V$$

$$Total\ Current = 66.7mA \times 3\ = 200.1\ mA$$

$$Total\ Power =\ \ = 0.36W \times 3\ = 1.08\ W$$

The total calculated power for the LED circuit is 1.08W.

3.2.2.8.1 LED Strip Calculations

Figure 3.2.2.8.1.1: *Transistor-Relay Circuit Diagram [24]*

According to the datasheet for the LED strips, the current requirement is 0.6 Amps/meter. Approximately 1 meter was used in the machine. 12V are required to power the strips.

$$Voltage\ =\ 12V$$

$$Current\ =\ 0.6\ A$$

$$Power = \ Current\ \times\ Voltage\ = 12V\ \times 0.6\ A\ = 7.2\ W$$

The calculated power requirement of the LEDS is 12W.

3.2.2.9 Limit Switch Board

The Limit Switch board is depicted in Figure 3.2.2.9.1. It contains a green indicator LED to know when power is being supplied to the circuit and a red indicator LED to know when the switch is being pressed.

Figure 3.2.2.9.1: *Limit Switch Circuit Board (left) with its detailed pin diagram (right). Note that solid lines represent wires and the light green represents solder bridges under the board.*

3.2.2.9.1 Limit Switch Calculations

The limit switch is connected in the circuit as shown in Figure XX.

Figure 3.2.2.9.1.1: *Limit Switch Circuit Diagram[16]*

According to the Data Sheet, the red LED has requirements of [25]:

$LED\ Voltage\ =\ 1.8V$

$Current\ =\ 5\ mA$

$LED\ Resistance\ =\ 360\ \Omega$

$Total\ Circuit\ Resistance\ = 360\ \Omega\ + 220\ \Omega = 580\Omega$

$Power = \frac{(Voltage)^2}{Resistance} = \frac{(5V)^2}{580\Omega} = 0.043\ W$

The calculated power requirement of the LEDS is 0.043W.

3.2.2.10 12V and 5V power requirements

The current and power requirements for the PIC DevBugger Board are unavailable on the User Manual, so the current requirement has been estimated from the given power supply for the board (12V 1A).

Table 3.2.2.10.1: *Total Power Calculations*

Circuit Component	Voltage Requirement	Current Requirement	Power Consumption
Arduino Nano	5V	659 mA	3.295 W
Potentiometer	5V	0.5mA	0.0025 W
LED Control Circuit	5V	200.1 mA	1.08 W
Limit Switch Board	5V	5 mA	0.043 W
Positional Servo Motors	5V	750mA	3.75 W
Continuous Servo Motors	5V	1000mA	5 W
Linear Servo Motors	5V	250 mA	1.25W
Total Power Supply Requirements	5V	2.86 A	14.42 W
Indicator LEDs	12V	0.6 A	7.2 W
Rectifier Circuit	12V	160 mA	1.92 W
PIC DevBugger Board	12V	1A	--
Total Power Supply Requirements	12V	1.760 A	--

From these calculations, it was decided that a 5V 4A power supply and a 12V 2A power supply would be used, in order to meet the requirements of the circuitry of the machine.

The two power supplies used have been shown in Figure 3.2.2.9.10.1.

Figure 3.2.2.9.10.1: *The selected power supplies were a 12V 2A Switching Power Supply (left) and 5V 4A Power Supply (right)*[26]

3.2.3 Problems Encountered

3.2.3.1 Interference of AA and 9V signals

In the voltage division circuit, due to the way in which the batteries are connected and how the signals are measured, the 9V and AA signals will sometimes interfere with each other. This is because the orientation of the battery is unknown, so it will sometimes produce a negative voltage signal for the batteries. When one of the signals for the 9V is negative, it will cause there to be a positive signal of the same magnitude in the AA circuit. The opposite phenomenon occurs where there is a negative signal in the AA circuit.

This issue was avoided by recognizing that the voltage of a 9V battery decays exponentially after it is considered dead, and will most likely not ever reach a voltage of 1.5V. Therefore, the microcontroller as able to distinguish between the different battery types by comparing the magnitudes of the voltages, rather than just checking for the existence of a signal.

3.2.3.2 Degradation of quality of connectors

During the integration, many of the connectors had to be plugged and unplugged many times from the machine. This degraded the quality of many of the connectors that were made, and caused there to be loose connections in the circuitry when the connectors were plugged in. In order to reduce the amount of times that a connector had to be plugged and unplugged, on/off switches were added to each actuator.

3.2.3.3 5V power source for op amps was not enough

When testing the rectifier circuit, although Rail-to-Rail opamps were used, when the circuit was powered with 5V, the output voltage of the opamp saturated at around 3.17V. In order to solve this issue, 12V had to be supplied to the rectifier circuit in order to increase the saturation voltage.

3.2.4 System Improvements Made

3.2.4.1 Addition of Debugging LED

During integration, our team encountered many errors due to loose connections in the circuitry connectors. Each time this happened, all of the connectors going into and out of a board would have to be checked painstakingly with a multimeter. In order to reduce the time and effort required for debugging, green LEDs were added to each circuit. There LEDs would indicate whether or not a connection was receiving power and eliminate the need to check with a multimeter. An example of some of there LED are shown in Figure 3.2.4.1.1.

Figure 3.2.4.1.1: *Debugging LEDs on the Power Board and the limit switch board*

3.2.4.2 Addition of On/Off Switches and LEDS

During the debugging phase, sometimes it was desired to turn on and off each actuator in the machine individually. Unplugging and plugging the connectors repeatedly became very difficult and tiring, so small switches and LEDs were added to each actuator circuit in order to facilitate this process in a better manner. Each individual actuator could then be turned off or on with a simple switch. An example of one of these switches is shown in Figure 3.2.4.2.1, along with its blue indicator LED.

Figure 3.2.4.2.1: *On/Off switch and indicator LED for one of the actuators used in the machine.*

3.2.5 Experiments Performed and Alternatives Considered

3.2.5.1 Piezoelectric Sensor and Force Sensor

Figure 3.2.5.1.1: *Piezoelectric sensor [27]and Force Sensitive Resistor [28]used in the experiments. Datasheets can be found in the Appendix.*

Prior to choosing the Limit Switch as the method of indicating when a battery was in the orientation tube, experiments were performed on piezoelectric sensor and force sensitive resistors as alternatives. However, it was found that the vibrations from a battery falling and the weight of a battery did not produce a significant difference in the sensor readings. Therefore, something with a more significant signal had to be used, so the limit switch was selected.

3.2.5.2 Solenoid

Figure 3.2.5.2.1: *Solenoid used in experiments. Datasheets can be found in the Appendix[29]*

A solenoid was experimented with as an alternative to the linear servo. However, the solenoid repeatedly got stuck and did not respond to a signal that was sent to it until it was agitated in some way. The linear servo did not experience this error, so it was selected over the solenoid.

3.2.5.3 Rectifier IC

Figure 3.2.5.3.1: *Rectifier IC used in experiments. Datasheets can be found in the Appendix*[30].

A diode rectifier chip was initially used as an alternative to the op amp rectifier used previously. During individual testing, the rectifier ICs worked well, with some small drops in voltage. When 4 of the rectifiers were connected in order to rectify all 4 of the signals from the voltage testing circuit, there was a lot of interference between the circuits that caused all of the rectifiers to show the same output voltage signal. The opamp circuit did not have this interference issue, so it was selected over the Rectifier IC circuit.

3.2.6 Improvements and Suggestions

3.2.6.1 Make all power supplies from one plug

Due to the different voltage requirements from the different circuit components, this machine needed a 12V power source and a 5V power source. This resulted in two power supplies being used, which occupied a lot of space. This could be improved by integrating the power supplies into one, so that the machine only need one outlet, which saves space and wiring.

3.2.6.2 Fit all circuits onto one board

During the prototyping phase of the machine, the circuits were made in modular form so that they could be swapped out for different ones at the team's desire. Although this was good for the debugging phase, one of the disadvantages that it caused was that the circuits occupied a large amount of space. Now that the circuitry to be used in the machine has been finalized, it can be improved by fitting all of the circuit components onto one board, which would save a lot of space on the machine.

3.2.6.3 Custom Board without PIC DevBugger to reduce Costs

The PIC Devbugger Board was helpful during the debugging phase of the project, but adds a significant amount of weight, cost, and space occupation to the machine. When the debugging features of the board are no longer required, a custom board should be made that does not include these features, in order to reduce the cost of the machine, as well as the machine weight and space occupied by the circuits.

3.3 Microcontroller Subsystem

This section explains the functional and physical characteristics for the PIC18F4620 microcontroller subsystem in this design. The purpose of this subsystem is to use information from sensors to instruct actuators that move batteries into their appropriate containers, while recording the result. This functionality will be executed using the algorithms employed. ***Changes to the algorithms used from the original proposal will be mentioned throughout this section.***

3.3.1 Assessment of the Problem

The major problem that this subsystem had to address to develop the machine was splitting sorting mode and standby mode between two microcontrollers: the PIC18F4650 and the Arduino Nano. ***The decision to use two microcontrollers was made because during testing the Arduino Nano outperformed the PIC in terms of reliability when sending PWM signals to servos and reading voltages accurately from batteries.*** Below is a table outlining the deliverables from each microcontroller

Microcontroller	Standby Mode Deliverables	Sorting Mode Deliverables
PIC	• Provide user with current date and time • Persist battery storting information under user accessible history • Instruct the user how to operate the machine via LCD instructions	• Activate Arduino to control the actuators and voltage sensors • Wait for Arduino to send data ready signal, and store battery type in EEPROM • Time the sorting process and shut down Arduino after 3minutes
Arduino	• Wait for trigger signal from PIC to start sorting	• Activate servos in a coordinated manner to get batteries to base of funnel • Instruct servo to knock battery into voltage tester once limit switch is triggered • Test the voltage and type of battery • send information to PIC about battery type

3.3.2 Solution

The solutions to each mode are described in the paragraphs, flowcharts and code below.

3.3.2.1 Standby Mode

Changes from Proposal:

This mode remained virtually unchanged from the proposal with the exception that battery sorting is now started with the * key and emergency stop was moved to a physical button located beside the PIC board.

Solution:

When the machine is not sorting batteries, users can access data of past runs on the screen of the machine. This data is the operation information (ex. sorting logs with date and time) from up to 4 previous runs on the device and it is accessed by pressing the information buttons [A or B keys]. To return to the main screen while scrolling through data, the Home Button [0 key] can be pushed. Pushing the start button [* key] at any time will bring the machine into Sorting Mode. To stop the device from moving at any point during the Standby Mode or Operation Mode the Emergency Stop button (located beside the development board) can be pressed. This procedure is shown in Figure 3.3.2.1.1.

Figure 3.3.2.1.1: *A flowchart depicting the high level logic of the machine when not sorting batteries.*

3.3.2.2 Sorting Mode (Operational Mode)

Changes from Proposal:

This mode was changed significantly from the proposal. The Arduino now runs the same logic that the PIC had originally, but the additional logic for interfacing between the microcontrollers was added. For instance, the PIC now sends signals to the Arduino to start and stop the sorting process, while the Arduino sends battery type information to the PIC.

Solution:

After loading the batteries into the top bin and pressing the start button, the sorting mode autonomously sorts these batteries into four categories: i) charged AA batteries, ii) charged C batteries iii) Charged 9-Volt batteries, and iv) drained batteries. The detailed mechanism to do this is depicted in the flowchart of Figure 3.3.2.2.1. After system data configuration, the PIC microcontroller starts a three minute timer and triggers the Arduino to start controlling the actuators and sensors that process the batteries.

First, the Arduino starts counting iterations to time how long the sorting process is taking. Next it moves the rollers until either the limit switch sends a signal indicating a battery is on top of it or one of the timing thresholds have been reached. The first timing threshold is a fail prevention mechanism to respond to batteries that are jammed, in which case the rollers rotate forward, in reverse and in sporadic directions to reorient the clogged battery. The second timing threshold is used determine when all the batteries have been sorted. If the Arduino goes through 3 cycles of fail prevention without sensing a battery then the sorting is considered finished and a signal is sent to the PIC to indicate this state change. The last timing threshold is a termination timer from the PIC that counts to three minutes (the maximum allowed time to sort).

When the limit switch at the base of the funnel sends a signal, a positional micro servo knocks the batteries into the voltage tester and testing is initiated when the linear servo pushes the 8 leads of the voltage testing circuit against the battery (four leads per face). Next, the Arduino reads voltages from the four circuits as the average across 10 samples and determines the type and charge of the battery based on the appropriate voltage and circuit that the battery completed. Finally, the Arduino rotates and dumps the batteries into the appropriate bins after first sending the battery type to the PIC via binary encoding numbers zero to three (which represent the four states of the batteries). The Arduino will continue doing this until it receives a termination signal from one of the timers mentioned earlier.

Once the information is passed to the pins on the PIC, the Arduino sends a data ready signal to the PIC, and the PIC stores the values of pins into registers and then stores the register values into EEPROM to persist run statistics. When the run is finished, the complete information is stored in memory and displayed on screen before returning to standby mode.

Figure 3.3.2.2.1: *A flowchart depicting the high level logic of the machine when sorting batteries autonomously. (Note: at any time in sorting mode, the stop button [* key] can be pushed to cause the emergency stop interrupt, which immediately halts the machine.)*

3.3.2.3 EEPROM Structure

Memory Address	Data (in Words – 16 bits)	Example
`0xn1`	Year of Run	`2017: 0b11111100001`
`0xn2`	Month of Run	`February: 0b10`
`0xn3`	Day of Run	`First: 0b1`
`0xn4`	Total Number of Batteries	`Fifteen:0b1111`
`0xn5`	Duration of Sorting Run	`180 seconds: 0b10110100`
`0xn6`	Number of AA Batteries	`Three: 0b011`
`0xn7`	Number of C Batteries	`Three: 0b011`
`0xn8`	Number of 9 Volt Batteries	`Three: 0b011`
`0xn9`	Number of Drained Batteries	`Six: 0b0110`
...		

Table 3.3.2.3.1: *Showing how the sorting run data will be stored in the EEPROM memory, where n is the nth run for n between 1 and 5 inclusive.*

Figure 3.3.2.3.2: *LCD Displays for User*

3.3.3 Pin Assignments

Table 3.3.3.1: *Pin Assignments for the PIC18F4620 Microcontroller*

Pin	Assignment	Description	Pin	Assignment	Description
GND		Common Ground	GND		Common Ground
KPD	Digital Input	Releases RB4 for KB10	RC7	Digital Output	Arduino Start Sort Trigger
RC6	Digital Output	Turn off KPD on Emergency Stop	RC5		
RC4	SDA	RTC	RC3	SCL	RTC
RC2			RC1		
RC0			GND		Common Ground
RE2			VCC		
RE1	Digital Input	Battery Number High Bit	RE0		
RA7	OSC	Main Oscillator	RA6	OSC	Main Oscillator
RA5	Digital Input	Battery Number Low Bit	RA4	Digital Output	LED Blue
RA3			RA2	Digital Output	LED Green
RA1	Digital Input	Arduino Data Ready	RA0	Digital Output	LED Red
RD7	Digital Output	LCD Data Bit 4	RD6	Digital Output	LCD Data Bit 3
RD5	Digital Output	LCD Data Bit 2	RD4	Digital Output	LCD Data Bit 1
RD3	Digital Output	LCD Enable Bit	RD2	Digital Output	LCD Read/Write Bit
RD1			RD0		
RB7	Digital Input	Keypad Data Bit 4	RB6	Digital Input	Keypad Data Bit 3
RB5	Digital Input	Keypad Data Bit 2	RB4	Digital Input	Keypad Data Bit 1
RB3	Digital Input	LCD Enable Bit	RB2	INT0	Emergency Stop
RB1	INT1	Keypad (Data Available Interrupt)	RB0	Digital Input	Arduino Stop Sort Trigger

Table 3.3.3.2: *Pin Assignments for the Arduino Nano*

Pin	Assignment	Description	Pin	Assignment	Description
D13			D12	Digital Out	Battery Reading Ready
3V3			D11	PWM	Left Roller
VREF	Reference Voltage	Analog Voltage Calibration	D10	PWM	RIght Roller
A0	Analog Input	C Battery Voltage	D9	PWM	Battery Dispenser
A1	Analog Input	AA Battery Voltage	D8	Digital In	Limit Switch
A2	Analog Input	9V Left Side Voltage	D7	Digital Out	Sort Stop Trigger
A3	Analog Input	9V Right Side Voltage	D6	PWM	Battery Pusher
A4	Digital Out	Battery Number Low Bit	D5	PWM	Battery Rotator
A5	Digital Out	Battery Number High Bit	D4		
A6			D3	PWM	Battery Dumper
A7	(Digital In)	(Break Beam Sensor)	D2	Digital In	PIC Start Sort Trigger
5V			GND		Common Ground
RST			RST		
GND		Common Ground	RX1		
VIN	Power	Supply 5V	TX1		

3.3.4 Supporting Calculations [1]

The following formulas were used to calculate the scaling factor to convert runtime timer interrupts into seconds:

$$f_{out} = \frac{1}{T_{out}} = \frac{f_{clk}}{4*Prescaler*(256-TMR0)*Count} \text{ and } \frac{1}{(\frac{fclk}{4})*prescaler*bitsize} = overflows$$

from this equation a prescaler of 2 was calculated to be used with an XTAL frequency of 10 MHz.

3.3.5 Computer Programs Results

Significant functions from the Code in the Appendix are described below (Note: common functions like LCD keypad etc. from sample code are not described since it would be redundant to do so):

3.3.5.1 Arduino

Note: Arduino's standard library already has the functionality to read analog signals and output PWM signals without necessary calculations or register configurations

Battery Categorization Function

As mentioned in the circuits section, the Voltage Testing chamber is composed of four circuits, one of which a given battery completes. Once a battery lands in the chamber the function delays 1 second to allow the linear servo to extend fully and make contact with the battery. Next, 10 voltage measurements are taken and averaged from each of the four circuits. The results are then compared to conditions that are required to categorize each battery as charged (have a voltage that is at least 85% of the nominal voltage). Due to non-ideal cases that occurred during testing, it is required that the batteries are assessed considering 9V first, then C batteries then AA batteries. This is because in rare cases batteries can make contact with not only their own leads, but other leads, thus completing more than one circuit.

```
float volt0 = 0;
float volt1 = 0;
float volt2 = 0;
float volt3 = 0;

int i = 0;
delay(1000);
while (i < 10){

    volt0 += analogRead(A0) * 5.0 / 1023.0;
    volt1 += analogRead(A1) * 5.0 / 1023.0;
    volt2 += analogRead(A2) * 5.0 / 1023.0;
    volt3 += analogRead(A3) * 5.0 / 1023.0;
```

```
            i += 1;
        }

         volt0 = volt0/i;
         volt1 = volt1/i;
         volt2 = volt2/i;
         volt3 = volt3/i;

         if ( volt0 > 7.65/2 || volt1 >  7.65/2 || volt2 >  7.65/2  ){    //
9V
           return 3;
         }
           if (volt3 > 1.27/2 ){ // C
           return 1; // C
          }

         if ( (volt0 > 1.27/2 && volt0 < 2/2) || (volt1 > 1.27/2 && volt1 <
              2/2)){ // AA
           return 2;

        }  else { // 0V
           return 4;
         }
```

Battery Unjamming Function

As explained in the sorting mode section above, when the Arduino suspects that a battery is jammed, it executes the following function, which makes the wheels move in several different ways in an attempt to reorient the jammed battery in the proper way. Experimentation determined that this was the most successful pattern of wheel movements to unjam the batteries.

```
 if (roller_count > 0){
      rollers_fwd(); // pull batteries in
} else if (roller_count >= 50 && roller_count < 100 ){
      roller1.write(0);  // full backwards
      roller2.write(90); //stopped
} else if (roller_count >= 100 && roller_count < 200){
      roller1.write(90); //
      roller2.write(180);
} else if (roller_count >= 200 && roller_count < 250){
      rollers_bck() ; // push batteries out
}
```

Voltage Testing Chamber Jiggle Function

During testing, batteries were often knocked out of the funnel such that they landed in a vertical orientation instead of the necessary horizontal orientation. To solve this issue, the testing chamber was coded to rapidly rotate back and forth in an attempt to swing the battery into the chamber. This also helped to ensure the battery settled in their appropriate configurations.

```
for (int j = 0; j > 3; j++){
     delay(500);
     dumper.write(100);
     rotator.write(10)
     batt_pusher.write(90);
     delay(200);
     dumper.write(80);
     rotator.write(90);
 }
```

Send Information to PIC Function

As mentioned in the integration section, the PIC development board does not have pull down resistors at its pins, which makes the use of HIGH I/O necessary for the Arduino to communicate properly with the PIC board. The Arduino sets the pins to the appropriate voltage using decimal to binary encodings for the battery states, which range from 0 to 3 (see encoding table). Where Pin A5 is the most significant bit and Pin A4 is the least significant bit. Finally, the Arduino sends a data ready signal to the pic through the trigger pin 12.

```
//uses high I/O

  switch (bin){

    case 1:
      digitalWrite(A4, LOW);
      digitalWrite(A5, HIGH);
      break;

    case 2:
      digitalWrite(A4, HIGH);
      digitalWrite(A5, LOW);

      break;

    case 3:
      digitalWrite(A4, LOW);
      digitalWrite(A5, LOW);
```

```
    break;

  case 4:
    digitalWrite(A4, HIGH);
    digitalWrite(A5, HIGH);
    break;
}

delay(500);
digitalWrite(12, LOW);

delay(500);
digitalWrite(12, HIGH);
digitalWrite(A4, HIGH);
digitalWrite(A5, HIGH);
```

Table 3.3.5.1.1: *Encoding for Arduino Communication*

Battery Type	Decimal Encoding	Binary Encoding	High I/O
C	1	01	10
AA	2	10	01
9V	3	11	00
Dead	0	00	11

3.3.5.2 PIC

Battery Sorting Function

When start sort is pressed on the Keypad, the global date-time and battery count arrays on the PIC are reconfigured, a message is displayed on the screen and the timer is set to start. Next the PIC sends the Arduino a signal to start battery sorting and waits until the timer interrupts have set the duration to 3 minutes or the Arduino sends a signal back to the PIC indicating that sorting has finished faster than expected. While waiting the pic polls on the data ready pin until the Arduino sends a signal indicating battery information is ready. Based on HIGH I/O readings the PIC stores the battery result and changes the indicator LEDs to the appropriate color. When one of the termination signals is reached, the PIC will display a finished message on screen, store the sorting data in EEPROM and display the results on the screen.

```
date_time[0] = 0;
 ...
battery_count[0] = 0;
```

```
    ...
      total = 0;
    print_lcd("TIME  ELAPSED:  ","SORTING  BATTERIES...",  "IN  CASE  OF
EMERGENCY", "PRESS EMERGENCY STOP");

    read_time(date_time);
    init_TMR0();
    TMR0ON = 1;
    LATCbits.LATC7 = 1;// start Arduino

    while ( duration <= 180 && PORTBbits.RB0 == 1){ // the max operation
time is 3min = 180s

        set_colour(0,1,0);

          while (PORTAbits.RA1 == 1); // wait till Arduino is ready to be
read

        total = total + 1; // increment battery count

        if (PORTEbits.RE1 == 0 && PORTAbits.RA5 == 0 ){ // C

            battery_count[0] = battery_count[0] + 1 ;
            set_colour(1,1,0); //yellow

        } else if (PORTEbits.RE1 == 1 && PORTAbits.RA5 == 1 ){// 9V

            battery_count[1] = battery_count[1] + 1 ;
            set_colour(0,0,1); // blue

        } else if (PORTEbits.RE1 == 1 && PORTAbits.RA5 == 0 ){ // AA

            battery_count[2] = battery_count[2] + 1 ;
            set_colour(1,0,1); // purple

        } else if ( PORTEbits.RE1 == 0 && PORTAbits.RA5 == 1 ){ // Dead

            battery_count[3] = battery_count[3] + 1 ;
            set_colour(0,1,1); // teal
        }
    }


    finished_bsm();


    //store run information in EEPROM
    save_and_print_sort_data(date_time, total, duration, battery_count);
```

3.3.6 Suggestions for Improvement of Subsystem

i. Communication between two Microcontrollers:

Relaying information between the PIC and Arduino requires additional logic and polling that makes this subsystem more complicated than necessary. Furthermore, using two microcontrollers adds additional points of failure to the algorithm in terms of loose connections and interference that would not be present if one microcontroller were used. Therefore this subsystem can be improved if one microcontroller were used. Normally, the PIC18 series would be considered to have more features than necessary to complete this task since it has a surplus of memory and pins to stay general while prototyping and designing. However for this particular design, more PWM pins than on the PIC18 are necessary, and a microcontroller with a more reliable history of PWM consistency is needed since PWM is vital to process of this machine. Thus in future iterations a custom microcontroller should be ordered with the appropriate number of pins with more reliable PWM features should be used. If this is unavailable, using only the Arduino nano or Arduino mega should be considered for the project. A microcontroller that uses C as a base language is necessary since compiled programs can be more memory optimized, and allows for more readable code [1] .

ii. Common grounding pins of Arduino and PIC on the development board:

While debugging the development board it was noticed that the Arduino's ground pins did not have common ground with reference to the PIC. In future iterations of the development board should have an implicit connection between all grounds.

4. Integration

Figure 4.1: *Venn Diagram of problems encountered in the different subsystems during integration.*

The individual problems from each subsystem have already been discussed in their respective sections of the report. This section will elaborate on issues that arose during integration between subsystems.

4.1 Electromechanical and Circuits Integration Problems

4.1.1 Circuit mounting

Problem

Mounting the circuits within the physical boundaries of the robot, ensuring adequate space for any additional circuits/modifications/replacements. This was difficult due to the small space constraint of the machine, which had limited volume available for the wiring and connectors needed for the circuitry.

Solution

In order to mount all of the required circuits, extra space had to be created inside the robot to accommodate both power distribution and control circuits. Initially, two wooden plates were constructed and statically mounted on the upper support braces, between the two back vertical members. The wooden panels were very durable, and presented ample space to mount all the required circuits. However, the static mounting was very cumbersome, because gaining access to the inner components of the robot required full disassembly of the panels, along with any attached wiring. Our solution was to mount the panels using hinges instead of the traditional static brackets, and that allowed very easy access to the inside of the robot, since the panels can be easily opened/closed, requiring no disassembly of any kind.

Figure 4.1.1.1: *Mounted Circuit Boards, showing hinges (blue), and screws (red)*

The material was later changed to acrylic, because it provides the ability to look past the panel into the inner portion of the robot, which can help in troubleshooting and performance assessment. The acrylic is a little flexible, so operating the circuits must be done with care in order to preserve the structural integrity of the panels.

The circuits were mounted using two screws and two nuts each. ¾ inch 4-40 machine screws were used, and the circuit boards were manually drilled in order for the screws to fit in.

Only two screws were used because they adequately held the circuit boards to the acrylic panels, cut the number of required screws in half, which reduced weight (albeit minor) and cost.

4.1.2 Cable Management

Problem

Cable management was a much more challenging task than first anticipated in the early design stages of the robot, mainly due to the added circuits and increased dependency on Arduino. The small nature of the robot required accurate and reliable cable management, where no stray wiring is allowed. The wires connecting actuators to control circuits, especially servos on the voltage testing chamber, must also be able to accommodate the full range of motion the actuator must go through, without presenting the risk of added mechanical resistance and accidental entanglement with other components. The wiring technique must also allow for the quick removal and repositioning of wires, to accommodate any design/placement changes.

Solution

The cables were held to the structure of the robot using industrial strength velcro, which can be easily removed and repositioned as required. The velcro also allows the wires to be pulled for a small distance, which prevents any out-of-range actuation from damaging components.

Figure 4.1.2.1: *One of the wires mounted with velcro.*

4.1.3 Difficulties with rigid wiring

Problem

The type of wiring used was too rigid, and cable was managment was difficult because it required a lot of wire bending, and the wires had to be wrapped around/through some components in order to be well restrained/reach their final connection point.

Solution

The wires were changed to a more flexible type, making the cable management job easier. However, the softer wires are multi-threaded, which makes electrical connections more difficult, and as a result, more solder was required to ensure robust and durable connections.

4.1.4 Laser Break-Beam Sensor as a stop condition

Problem

The device requires a way of determining if no more batteries are still in the funnel, and thus halt operation. This must be done without impeding the normal operation of the robot.

Solution

Our proposed solution was to use a laser break beam sensor in order to detect the presence of batteries. The laser would have been mounted beneath the vertical feeder, and the laser would emit through a hole in the bottom plate. Then, the signal would travel up the vertical feeder and trigger a photoresistor mounted on the upper surface of the loading funnel, which would signal the end of operation. However, this concept was discarded due to its low reliability and frequent malfunctioning. Because the bottom plate of the feeder was subject to vibrations and impact loading due to batteries falling, the laser would eventually become misaligned with the photoresistor, and the system would not function at all. Another problem was that the laser light scattered significantly inside the feeder due to the ample presence of many metallic objects, which significantly weakened the signal. Another problem was the frequent fluctuation in the light intensity arriving at the photoresistor, which caused incorrect halt signals, and ultimately the end of operation even though not all batteries were sorted.

4.1.5 Voltage Testing Contact Issues

Problem

The original copper leads of the voltage testing chamber did not properly come in contact with the leads of the battery, leading to incorrect voltage measurements, or no measurements at all.

Solution

Our solution was to use a more flexible spring material rather than hard copper leads for the voltage measurement pins in the machine. These leads were able to have more contact with the battery leads and we experience better measurements as a result. The metal spring were taken from a solder sponge, and soldered on top of the original copper leads.

Figure 4.1.5.1: *The copper leads of the voltage testing chamber with metal springs soldered on.*

4.2 Electro-Mechanical and Microcontroller Integration Problems

4.2.1 Micro Servo Coordination and Control

Problem

When servos were moving in full range of motion, they sometimes got caught on components of the machine. Also, the servos were not timed well enough to allow smooth flow of batteries through the machine. For example, the front rollers spun fast enough to cause batteries spill out of the base of the machine funnel without detection of the limit switch, leaving batteries stuck in the voltage chamber. Furthermore the voltage measurements were originally taken without waiting for the linear servo to make full contact with the battery, causing lower voltage averages and inaccurate readings. Finally, moving the battery to its final location originally caused the servos to move so fast that they flung batterie against collection bin walls, which risked damaging the batteries.

Solution

To stop servos from getting caught on the machine, servo mounts had to be tweaked, tightened and translated as well as servo motion had to be reduced from its full range to the appropriate bounds that were available. For example, since the positional servo that knocked batteries into the voltage tester kept getting caught on the metal of the funnel (and damaging itself), the servo was remounted several millimeters lower to not graze the metal of the funnel during motion and the the servo was set to move from 0 to 150 degrees instead of the original 180 to not have the servo slam into the side of the funnel.

To fix speed and timing issues, experiments had to be run to determine the best speed for the rolers, the optimal time to wait before measuring voltage and appropriate speed and delay necessary to move the batteries gently into the bin.

4.3 Microcontroller and Circuits Integration Problems

4.3.1 Servo and Connections and Interference

Problem

After all the wiring for the robot was complete, several servos on the machine started to behave chaotically. This unsolicited behaviour was composed of shivering or sharp movements when the servos were connected to power but not given any PWM signals. The servo that exhibited this behaviour most was the regular positional servo at the bottom, which was found to be drawing more current to operate than the rest of the actuators.

Solution

In terms of connections, it was discovered that the rigidness of the original wires used in cabling caused the pins of servos to be loosely connected and eventually become disconnected if the machine was moved around too much. To solve this issue limp wires were used for recabling, longer pin adapters were used for interfacing wires and actuators, and these interfaces were taped together.

In terms of interference, testing revealed that having PWM signal wires too close to each other caused signal interference. To solve this issue, cabling was braided tighter and the cables were mounted farther apart.

Finally, the bottom servo was replaced with a positional micro servo that had the same current and voltage ratings as the other servos being used in the machine. After these changes were made, significant stability in the actuators was achieved.

4.3.2 Inadequate Current from PIC to control Relay Switches

Problem

The output pins of the PIC were unable to supply sufficient current in order to activate the 5V relay switches used to control the LED lighting on the machine. They could only supply 25mA. Transistors could also not be used in isolation because the LED strip only had one ground pin, so they could not control each LED colour individually.

Solution

TIP120 Transistors were added to the relay circuit, so that the PIC pins could send signals to transistors with a direct 5V supply instead. The transistors then sent a signal to the relays, which then turned on specific LED colours. This circuit has be further discussed in the circuits subsystem section.

This solution was able to overcome the current limitation of the PIC pins and the single-ground-pin limitations of the LED strip through the combination of transistors and relays.

4.3.3 Inaccurate Voltage Measurements from Arduino

Problem

The Arduino Microcontroller did not measure voltages in its analog input pins correctly. This was because its reference was suppose to be 5V, but when the REF pin on the Arduino was measured, it was actually outputting around 4.1V. This may have been due to the fact that the Arduino being used was made by a third party, and the quality may not have been very good.

Solution

In order to fix the reference voltage, a potentiometer was used in a voltage division circuit in order to calibrate an output voltage of 5V, which was then fed into the REF pin of the Arduino. This corrected the voltage readings by providing a correct reference voltage. This circuit is discussed in more detail in circuits subsection.

4.3.4 Linear Servo Power Supply

Problem

In the datasheet for the Linear Servo Motor used in this project, it is rated to be 3.7 V. During testing, voltage regulator circuits were created for the linear servo in order to make sure that it was not being set 5V. At this voltage, many of the motors did not work or moved very slowly, which was not sufficient for its purpose in our machine.

Solution

In order to test the limitations of the linear servo motor, our team sent it 5V instead. This was found to increase the speed of the servo to a desired pace, but our team was worried that this higher voltage may damage the servo. We proceeded to repeatedly test several linear servos with 5V in order to ensure that it would not break during an evaluation. It was observed that the servo repeatedly performed better at this voltage and showed no signs of degradation, so it was operated at 5V instead of 3.7 from then onwards.

4.3.5 Development Board Diode

Problem

Two weeks before public demonstration, the heatsink on the development board started to get dangerously overheated and behave inconsistently when loaded with the same software.

Solution

The current and voltage were analyzed through modules on the development board with the PIC development board designer, Michael Ding. It was discovered that a zener diode (the 1SMB5919BT3G[31]) placed by the input of the voltage source for power spike protection was drawing a significant amount of current to cause the overheating, even though it was only rated to behave this way for 5.3 volts or higher and the PIC was operating with 4.99 volts. After the zener diode was removed from the board the PIC stopped overheating and behaved regularly.

4.3.6 I/O Communication between PIC and Arduino (High/Low IO)

Problem

The PIC development board was able to send information to the Arduino to start the sorting process but the Arduino was not able to communicate its sorting information back to the PIC reliably.

Solution

This problem occurred because the Arduino was communicating with the PIC using LOW I/O and the development board did not have any pull down resistors. Thus it was not guaranteed that the voltages of the pins would remain low without a constant low signal going to them. To fix this HIGH I/O was used to pass information to the PIC instead, which was found to be consistently more reliable.

4.4 Overall System Improvement Suggestions

In addition the the individual subsystem improvements mentioned in above sections, the following are improvements that could be made to the entire system:

4.4.1 Increase Machine Size to allow for better cable management

The robot was built with a 45 cm height and a 30 cm base radius, even though the minimum size restrictions allowed for a base that was 45 cm long. This restriction made it more difficult to tweak the design during debugging and remount servos where necessary during the integration process. However, it most notably made it difficult to mount circuits and connect them with cables. This is because once circuits were installed, there was little space left to install cables where they would not have a chance of interfering with the rollers or the voltage chamber motion. As a result cables had to be mounted down significantly to the frame, which reduced the range of motion for the voltage tester.

4.4.2 Use of ribbon connectors rather than individual jumper wires

Throughout the project, cabling was done via soldering on connectors, inserting stripped wires into housings and using manufactured ribbon connectors. The soldered on connectors were found to be the least reliable due to the potential for the soldered interface to break under the flexible conditions needed for the wire. Housings seemed like a better option but also had issues since the plugs of the housings would slip, causing the illusion of a plugged in connection when the leads were not in contact. However, when using manufactured jumper cables between boards no significant issues came up. Thus replacing all the connecting cables with jumper would increase the reliability and flexibility of the robot.

4.4.3 Integration of Circuits into a Single Board

During the testing process of the machine, circuits were made modularly so that components that were malfunctioning or not permanent could be swapped around until a final prototype was made. Now that the prototype is complete, it would be ideal for all the circuits to be placed on one board. Since space on the robot was so limited, having a single circuit board to handle all signal connections would be preferred since it allows for more room for wiring.

4.4.4 Addition of Vertical Conveyer belt to assist in battery movement

Originally the machine was designed to restrict the flow of batteries to a single lane by the use of trap doors that pushed the batteries to the back corner of the funnel and dampened batteries that were falling together so that one would fall faster. Although this is theoretically functional under ideal conditions, after several tests the spring dampening system became worn out as the metal plates bent from impacts, or the springs k values changed from warping or the spring became loose from its glue. To resolve these issues, having additional rollers at the sides of the funnel in place of the spring door, or a conveyer belt mounted vertically down the funnel should act to more reliably transport the batteries to the funnel base without being wearing down quickly.

4.4.5 Working Laser Indication

The original machine design did not include a sensor to test whether the machine was occupied by batteries, but instead relied on software timers. Installing and testing a break-beam sensing system to determine when the funnel is completely empty would allow the machine to more accurately make the decision to stop or continue to draw batteries into the funnel, thus saving time. As mentioned in one of the prior sections, an attempt to install the break beam sensor was made, but the mounting and orientation was found to be to unreliable to use for demonstration.

5. Initial and Accomplished Schedule (Gantt Charts)

The initial Gantt Chart from the proposal is included below.

The following updated Gantt Chart shows the actual team progress (in dark Blue) overlaid on the original Gantt Chart (in faded red).

Please note that it is sometimes difficult to see the faded red bars of the past Gantt Chart behind the dark blue bars. The Gantt chart is better viewed in the digital version of this report, where the reader can zoom in to view the chart details.

As shown in this updated Gantt Chart, most of the team's individual tasks proceeded on schedule, or before schedule. However, integration took a much longer time than the team had initially predicted. This was mostly due to the team members' beliefs that if their respective subsystems worked individually, they would be able to work together, with minimal changes. However, this was not the case in reality. In future projects, more time should be allocated for integration and debugging.

6. Conclusion

This design report outlines the design process and features of a Battery Dispensing and Sorting Machine. This machine is able to sort up to 15 batteries at a time, and it is able to separate 9 volt, AA and C type batteries. The machine is also able to separate charged batteries from uncharged ones. The machine will cost 224 CAD.

This machine will be deployed in the recycling industry, in order to aid in the correct and safe dispensing of batteries after use. This is a very important task because battery waste is a very toxic agent to the environment, and the chemicals released by batteries can harm many forms of living organisms.

The design process will be composed of three subsystems: electromechanical, circuit, and microcontroller. The electromechanical subsystem is responsible for the machine's strategic and structural design, analysis, and construction, as well as actuator selection and mounting. Mechanical power transmission and any required mechanisms are to be designed by the electromechanical member. The circuits and sensors portion is responsible for designing all sensory systems and all the functional control and power distribution circuits that will control the battery flow and sorting processes, as well as provide sufficient power to microcontrollers to ensure safe operation. The microcontroller portion is responsible for designing the system logic and coordinating the actuators and sensors in order to receive the desired response from the machine. The microcontroller is also responsible for storing and displaying data regarding the machine's performance.

This machine sorts batteries based on their shape and charge in a custom-designed voltage testing chamber. The machine operator loads the batteries via the top loading funnel. From there, servo-driven Ethylene Vinyl Acetate wheels will feed the batteries from the funnel into a vertical feeding column where they will be stacked and aligned vertically, ready for dispensing. Each battery will then be sequentially placed in a voltage testing chamber that is capable of simultaneously determining the type of battery and whether that battery is drained or charged. The chamber then rotates and places batteries into their respective bins at the base of the machine using servo motors.

This design is considered very efficient because it occupies very limited space, is highly portable, lightweight, and completes the sorting process in under 3 minutes. It is capable of being relocated into more remote areas, which is a very important feature, since it can be used in countries without very well established recycling industries and infrastructures. The simplicity of the operation procedure is also a striking feature, since the general consumer will not require any form of training, or an elongated period of time to learn how to operate the machine. The design is also highly modular, which makes it very easy to build and repair.

Future improvements can be made to the machine through slight shape adjustments in the neck of the funnel to prevent jamming, and through the development of a custom printed circuit board to eliminate the use of the DevBugger Board and reduce the cost of the machine. During the public demonstration of the machine, one of the runs experienced jamming that caused failure of full operation. Although jamming has been greatly reduced since the first prototype development, it occasionally still happens. Efforts into research as to how jamming can be permanently prevented would be very beneficial to the future of the machine, as that would ensure successful sorting every time. More research is also needed for better circuit and microcontroller optimization. This was an issue that sunk up a lot of the development time, because the machine requires many servo motors and accurate timing is difficult to attain without the use of a secondary micro controller. The best approach is to design one microcontroller with robust PWM capability, and optimize the circuits for minimal space and power consumption.

7. Description of Overall Machine

7.1. Overall Machine Size and Views

Figure 5.1.1.1 *Top View of Machine*

Figure 5.1.2.1 *Front View of Machine*

Figure 5.1.3.1 Side View of Machine

7.2 Operating Conditions

7.2.1 Power Supply

The machine needs two power supplies, one 5V and one 12V.The 5V power supply needed to supply a minimum of 2.86 A, and the 12V power supply needed to supply a minimum of 1.76 A.

7.2.2 Loading Amount

The machine can handle a maximum of 15 batteries in the loading funnel when required, but worked optimally with smaller amounts.

7.2.3 Operational Temperature

Due to the circuitry of the machine, it is intended to be operated at approximately room temperature indoors, in a dry area. Please keep small children and moisture away from this machine.

7.3 Standard Operating Procedure

1. Plug the white power cord of the machine into an AC 110V-60Hz outlet.

2. Turn ON the. PIC Devbugger Board and ensure that “programming mode” is turned off.

3. Turn ON the Power Distribution Board, the machine will enter standby mode and glow white.

4. Ensure that the battery containers at the bottom of the machine are in place and held in by the velcro.

5. Load the batteries into the funnel of the machine.

6. When ready, Press * on the white keypad to start sorting the batteries, and the machine will glow green.

7. In case of an emergency, please activate the Emergency Stop button, resolve the emergency and then reset the emergency stop button. The machine will glow red in case of an emergency.

8. When the sorting process is complete, the machine will re-enter standby mode and glow glow white

9. Retrieve the sorted batteries by removing the containers.

10. Look at the display to retrieve storing information recorded by the machine.

11. For sorting information about previous runs, please follow the instructions displayed on the screen.

Appendices

Appendix A: References and Bibliography

References

[1] R. Emami, *Multidisciplinary Engineering Design*. McGraw-Hill Ryerson Limited, 2017.

[2] "The Environmental Impacts of Battery Production," *Black & Veatch*, 13-Dec-2016. [Online]. Available: https://www.bv.com/Home/news/solutions/environmental/the-environmental-impacts-of-battery-production. [Accessed: 13-Apr-2017]

[3] "Ontario Battery Recycling Program: Why Call2Recycle? – Call2Recycle® | Canada." [Online]. Available: http://www.call2recycle.ca/ontario-battery-why-call2recycle/. [Accessed: 13-Apr-2017]

[4] "Optical Battery Sorter 600," *Refind Technologies*. [Online]. Available: http://www.refind.se/theobs/. [Accessed: 13-Apr-2017]

[5] "Optical Battery Sorter 200," *Refind Technologies*. [Online]. Available: http://www.refind.se/optical-battery-sorter-200/. [Accessed: 13-Apr-2017]

[6] G. Q. Zhang, L. M. Li, and S. Choi, "Robotic automated battery sorting system," *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2012.

[7] Canadian Intellectual Property Office, "Patent 1196379 Summary - Canadian Patents Database." [Online]. Available: http://www.ic.gc.ca/opic-cipo/cpd/eng/patent/1196379/summary.html?query=((battery)%20%3CAND%3E%20(voltage)%20%3CAND%3E%20(test%20OR%20sort))&start=1&num=50&type=boolean_search. [Accessed: 13-Apr-2017]

[8] T Issa And, *Usability and Human Computer Interaction (HCI)*. Sustainable Design, 2015.

[9] "Data and Formulae for Mechanical Engineering Students." [Online]. Available: https://workspace.imperial.ac.uk/mechanicalengineering/Public/files/databook_previous.pdf. [Accessed: 13-Apr-2017]

[10] G. W. Herbert, *Steel Construction Manual*. American Institute of Steel Construction, 1987.

[11] M. F. Ashby, *Materials Selection in Mechanical Design*. Butterworth-Heinemann, 2016.

[12] "9g Micro Servo Motor (4.8V)." [Online]. Available: http://www.robotshop.com/ca/en/9g-micro-servo-motor-4-8v.html. [Accessed: 13-Apr-2017]

[13] "9g Continuous Rotation Micro Servo." [Online]. Available: http://www.robotshop.com/ca/en/9g-continuous-rotation-micro-servo.html. [Accessed: 13-Apr-2017]

[14] "VS-19 Pico Linear Servo." [Online]. Available: http://www.robotshop.com/ca/en/vs-19-pico-linear-servo.html. [Accessed: 13-Apr-2017]

[15] "Arduino Nano." [Online]. Available: http://www.electronics-lab.com/wp-content/uploads/2017/02/ARDUINO_NANO.png. [Accessed: 13-Apr-2017]

[16] "SPDT Microswitch with Extended Lever - 55mm," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/spdt-microswitch-with-extended-lever-55mm/. [Accessed: 13-Apr-2017]

[17] "Potentiometer Circuit." [Online]. Available: https://cdn.sparkfun.com/assets/6/3/e/5/e/511ac8f5ce395f5846000000.png. [Accessed: 13-Apr-2017]

[18] "HS-81 SUB MICRO SERVO." [Online]. Available: http://www.robotshop.com/media/files/pdf/hs81.pdf. [Accessed: 13-Apr-2017]

[19] "VS-19 Pico Linear Servo." [Online]. Available: http://www.robotshop.com/media/files/pdf/datasheet-22090.pdf. [Accessed: 13-Apr-2017]
[20] "LMC6462 Dual/LMC6464 Quad Micropower, Rail-to-Rail Input and Output CMOS Operational Amplifier." [Online]. Available: http://www.ti.com/lit/ds/symlink/lmc6462.pdf. [Accessed: 13-Apr-2017]
[21] "LT1078/LT1079 10789fe Micropower, Dual and Quad, Single Supply, Precision Op Amps." [Online]. Available: http://cds.linear.com/docs/en/datasheet/10789fe.pdf. [Accessed: 13-Apr-2017]
[22] "SONGLE RELAY." [Online]. Available: http://www.songle.com/pdf/20085271543431001.pdf. [Accessed: 13-Apr-2017]
[23] "TIP120/TIP121/TIP122 NPN Epitaxial Darlington Transistor." [Online]. Available: https://cdn-shop.adafruit.com/datasheets/TIP120.pdf. [Accessed: 13-Apr-2017]
[24] "RGB LED Strips." [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/rgb-led-strips.pdf. [Accessed: 13-Apr-2017]
[25] "3mm LED - Red (20 Pack)," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/3mm-led-red-20-pack/?search_query=red+led&results=225. [Accessed: 13-Apr-2017]
[26] "12V 2A Switching Power Supply," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/12v-2a-switching-power-supply/?search_query=12V+power+supply&results=19. [Accessed: 13-Apr-2017]
[27] "35mm Piezo Element," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/35mm-piezo-element/?search_query=piezoelectric+sensor&results=5. [Accessed: 13-Apr-2017]
[28] "Force Sensitive Resistor - Square," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/force-sensitive-resistor-square/?search_query=force+sensor&results=17. [Accessed: 13-Apr-2017]
[29] "12-24V Pull Type Solenoid," *Creatron Inc*. [Online]. Available: http://www.creatroninc.com/product/12-24v-pull-type-solenoid/. [Accessed: 13-Apr-2017]
[30] "W02M MULTICOMP, Bridge Rectifier Diode, Single, 200 V, 1.5 A, Through Hole, 1 V, 4 Pins, Newark element14 Canada." [Online]. Available: http://canada.newark.com/multicomp/w02m/bridge-rectifier-single-phase/dp/63W5272. [Accessed: 13-Apr-2017]
[31] "3 Watt Plastic Surface Mount Zener Voltage Regulators." [Online]. Available: http://www.mouser.com/ds/2/308/1SMB5913BT3-D-108682.pdf. [Accessed: 13-Apr-2017]

Bibliography

[1] Primary batteries – Part 2: Physical and electrical specifications (IEC 60086-2), 13th ed., vol. 2. Geneva: International Electronics Commission, 2015.

[2]R. Emami, Multidisciplinary Engineering Design. Canada: McGraw-Hill Ryerson Limited, 2017.

[3]"Blog," Ontario Battery Recycling Program: Why Call2Recycle? – Call2Recycle® | Canada. [Online]. Available: http://www.call2recycle.ca/ontario-battery-why-call2recycle/. [Accessed: 01-Feb-2017].

[4] M. S. Sanders, Human Factors in Engineering and Design, McGraw-Hill, Inc., 1993.

[5] T. Issa and P. Isaias, "Usability and Human Computer Interaction (HCI)," Sustainable Design, pp. 19–36, 2015.

[6] "Optical Battery Sorter 600," Refind Technologies. [Online]. Available: http://www.refind.se/theobs/. [Accessed: 01-Feb-2017].

[7] "Optical Battery Sorter 200," Refind Technologies. [Online]. Available: http://www.refind.se/optical-battery-sorter-200/. [Accessed: 01-Feb-2017].

[8] G. Q. Zhang, L. M. Li, and S. Choi, "Robotic automated battery sorting system," 2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 2012.

[9] R. E. MCAULIFFE, "Patent 1196379 Summary," Canadian Intellectual Property Office, 13-Dec-1985. [Online]. Available: http://www.ic.gc.ca/opic-cipo/cpd/eng/patent/1196379/summary.html?query=((battery)%20%3CAND%3E%20(voltage)%20%3CAND%3E%20(test%20OR%20sort))&start=1&num=50&type=boolean_search. [Accessed: 01-Feb-2017].

Appendix B: Code

main.c

```
#include <xc.h>
#include <stdio.h>
#include "configBits.h"
#include "constants.h"
#include "lcd.h"
#include "I2C.h"
#include "macros.h"
#include "bsm.h"
#include "bsm_data.h"
#include "1_initialize_bsm.h"
#include "sorting_bsm.h"

char date_time [7] = {0,0,0,0,0,0,0};
int battery_count[4] = {0,0,0,0};
int total = 0;

unsigned char addr_bsm[4] = {0x00, 0x20, 0x40, 0x60};

int past_run_bsm = 0;
int oldest_run = 0;

int buffer = 0;

volatile unsigned char data[12];
int duration;

int display_RTC = 1;
int duration = 0;

int sort_state = 0;
int out_of_batteries = 0;

const char keys[] = "123A456B789C*0#D";

void stop_timer(void);
void emergency_stop(void);
void set_colour (int r, int g, int b);

void main(void) {

    initialize();

    home_bsm();

    while(1){
        di();
        if (display_RTC) {
            display_time();
        }
        ei();

        if (sort_state){
            start_bsm();
            sort_state = 0;
        }
    }

    return;
}


void interrupt isr(void) {
```

```
display_RTC = 0;

if(TMR0IF){ // for counting the seconds of the battery operation
    TMR0IF = 0;

    di();
    buffer++;
    if (buffer > 4){
        buffer = 0;
        duration++;


        __lcd_l1();
        printf("TIME ELAPSED: %dsec", duration);
        if( duration > 15 & duration%20 == 0 ) duration = duration + 1;



        if(duration < 180){
            init_TMR0();
            TMR0ON=1; //start timer 0

        } else {
            TMR0ON = 0;
            LATCbits.LATC7 = 0; // stop Arduino


            finished_bsm();
             save_and_print_sort_data(date_time, total, duration, battery_count);

        }
        ei();
    }
}

if(INT2IF){ // when the Emergency Stop Button is pushed, the system will halt
    INT2IF = 0; //reset for next interrupt
    TMR0ON = 0;

    emergency_stop(); //polls
    home_bsm();

}

if(INT1IF){// when a key is pressed, the system will change state appropriately
    INT1IF = 0;

    unsigned char keypress = (PORTB & 0xF0) >> 4;
    while(PORTBbits.RB1 == 1);

    unsigned char pressed_key = keys[keypress];
    switch(pressed_key){

        case '0':  home_bsm();break;
        case '*': sort_state = 1;break;

        case 'A': info_up_bsm();break;
        case 'B': info_down_bsm();break;

        case '1': set_colour(1,0,0); break;
        case '2': set_colour(0,1,0); break;
        case '3': set_colour(0,0,1); break;
        case '4': set_colour(1,1,0); break;
        case '5': set_colour(1,0,1); break;
        case '6': set_colour(0,1,1); break;
```

```
                case '7': set_colour(1,1,1); break;
                case '8': set_colour(0,0,0); break;
                case '9': set_colour(1,0,0); break;
            };
        }
    }


    void set_colour (int r, int g, int b){
        LATAbits.LATA0 = r;
        LATAbits.LATA2 = g;
        LATAbits.LATA4 = b;
    }


    void finished_bsm(void) {

        print_lcd("BATTERY SORTING "," COMPLETE!", "LOADING RESULTS...", "");

        return;
    }

    void emergency_stop(void) {

        print_lcd("  EMERGENCY  STOP!","ALL PROCESSES HALTED", "RESET EMERGENCY STOP"," BUTTON
TO CONTINUE");

        while (PORTBbits.RB2 == 1){

            LATCbits.LATC6 = 1; // suppress keypad
            LATCbits.LATC7 = 0; // stop Arduino

            set_colour(1,0,0);
            _delay(20000000);
            set_colour(0,0,0);

        }
            LATCbits.LATC6 = 0; // re-activate keypad

        return;
    }

    void home_bsm(void) {

        display_RTC = 1;
        set_colour(1,1,1);
        print_lcd("      TIME", "LOAD BATTERIES &","PRESS '*' TO START","[ A/B for DATA ]");

    }

    void start_bsm(void) {

        date_time[0] = 0;
        date_time[1] = 0;
        date_time[2] = 0;
        date_time[3] = 0;
        date_time[4] = 0;
        date_time[5] = 0;
        date_time[6] = 0;

        battery_count[0] = 0;
        battery_count[1] = 0;
        battery_count[2] = 0;
        battery_count[3] = 0;
```

```
            total = 0;


       //    set_colour(0,1,0);
             print_lcd("TIME ELAPSED: ","SORTING BATTERIES...", "IN CASE OF EMERGENCY", "PRESS EMERGENCY STOP");

           read_time(date_time); // information for past run

           init_TMR0(); TMR0ON = 1; // time how fast the run took in seconds

           LATCbits.LATC7 = 1;// start Arduino

            while ( duration <= 180){ //&& PORTBbits.RB0 == 1){ // the max operation time is 3min = 180s
               _delay(4000000);
               set_colour(0,1,0);

               while (PORTAbits.RA1 == 1); // wait till Arduino is ready to be read

               total = total + 1; // increment battery count

               if (PORTEbits.RE1 == 0 && PORTAbits.RA5 == 0 ){ // C

                   battery_count[0] = battery_count[0] + 1 ;
                   set_colour(1,1,0); //yellow

               } else if (PORTEbits.RE1 == 1 && PORTAbits.RA5 == 1 ){// 9V

                   battery_count[1] = battery_count[1] + 1 ;
                   set_colour(0,0,1); // blue

               } else if (PORTEbits.RE1 == 1 && PORTAbits.RA5 == 0 ){ // AA

                   battery_count[2] = battery_count[2] + 1 ;
                   set_colour(1,0,1); // purple

               } else if ( PORTEbits.RE1 == 0 && PORTAbits.RA5 == 1 ){ // Dead

                   battery_count[3] = battery_count[3] + 1 ;
                   set_colour(0,1,1); // teal
               }
           }


           finished_bsm();

         LATCbits.LATC7 = 0; // stop Arduino
           TMR0ON = 0;

           duration = 0;
           //store run information in EEPROM
           save_and_print_sort_data(date_time, total, duration, battery_count);
       }
    void save_and_print_sort_data (char* date_time, int total, int duration, int* battery_count ){
           // determine which data row to overwrite

           unsigned char data[12];

           memcpy(data, date_time, 6);
           data[6] = total;
            data[7] = duration; // needs to be global variable set by timer interrupt when timer is done
```

```
            data[8] = battery_count[0]; //AA
            data[9] = battery_count[1]; //C
            data[10] = battery_count[2]; //9V
            data[11] = battery_count[3]; //Dead

            print_info(data);
            write_data_bsm(addr_bsm[oldest_run], data);

            oldest_run = (oldest_run + 1);
            if (oldest_run > 3) oldest_run = 0; // CANNOT MOD IN MICROCHIP C
        }

        void info_up_bsm(void) {

            if (past_run_bsm < 3) past_run_bsm++;
            read_data_bsm(addr_bsm[past_run_bsm], data);
            print_info(data);

        }

        void info_down_bsm(void) {

            if (past_run_bsm > 0) past_run_bsm--;
            read_data_bsm(addr_bsm[past_run_bsm], data);
            print_info(data);

        }

        void print_info(volatile unsigned char data[12]){

              print_lcd("[A-UP,B-DOWN,0-HOME]",
                    "2017/01/29  15:10:23" ,
                    "TOTAL:## DURATION:##" ,
                    "AA:## C:## 9:## D:##");
              __lcd_l2();
                        printf("%d/%02x/%02x           %02x:%02x:%02x",  17,  data[5],  data[4],
data[2],data[1],data[0] );
              __lcd_l3();
              printf("TOTAL:%02xDURATION:%d",  data[6], 180 );
               __lcd_l4();
              printf("AA:%02x C:%02x 9:%02x D:%02x",  data[10], data[11], data[8], data[9]);
          }
        void display_time(void){
            unsigned char time[7];
            //Reset RTC memory pointer
            I2C_Master_Start(); //Start condition
            I2C_Master_Write(0b11010000); //7 bit RTC address + Write
            I2C_Master_Write(0x00); //Set memory pointer to seconds
            I2C_Master_Stop(); //Stop condition

            //Read Current Time
            I2C_Master_Start();
            I2C_Master_Write(0b11010001); //7 bit RTC address + Read
            for(unsigned char i=0;i<0x06;i++){
                time[i] = I2C_Master_Read(1);
            }
            time[6] = I2C_Master_Read(0);       //Final Read without ack
            I2C_Master_Stop();
            __lcd_l1();
            printf("%02x/%02x/%02x    %02x:%02x:%02x",
                time[6],time[5],time[4],time[2],time[1],time[0]);
    }
        void read_time(unsigned char* time){
            //Reset RTC memory pointer
            I2C_Master_Start(); //Start condition
            I2C_Master_Write(0b11010000); //7 bit RTC address + Write
```

```
    I2C_Master_Write(0x00); //Set memory pointer to seconds
    I2C_Master_Stop(); //Stop condition

    //Read Current Time
    I2C_Master_Start();
    I2C_Master_Write(0b11010001); //7 bit RTC address + Read
    for(unsigned char i=0;i<0x06;i++){
        time[i] = I2C_Master_Read(1);
    }
    time[6] = I2C_Master_Read(0);       //Final Read without ack
    I2C_Master_Stop();
}
```

Arduino_main.ino

```
 /*

 */
#include "Arduino.h"

#include <Servo.h>


Servo roller1;
Servo roller2;
Servo batt_dispenser;
Servo batt_pusher;
Servo rotator;
Servo dumper;

int dispense_flag = 0;
int tester_occupied = false;

int batt_type =0;
int jam_loop_count = 0;

int old_light = 0;
int new_light = 0;

int roller_count = 0;


void setup() {

 old_light =  analogRead(A7);

 Serial.begin(9600);
//  pinMode(0, INPUT);//instr

//  pinMode(1, INPUT);//instr

pinMode(2, INPUT);//instr

//  pinMode(3, OUTPUT);//PWM

//  pinMode(5, OUTPUT);//PWM
//  pinMode(6, OUTPUT);//PWM

 pinMode(7, INPUT); //LEDblue


 pinMode(8, INPUT); //switch

//  pinMode(9, OUTPUT);//PWM
//  pinMode(10, OUTPUT);//PWM
//  pinMode(11, OUTPUT);//PWM
```

```
 pinMode(12, OUTPUT);
// pinMode(13, INPUT);

// pinMode(4, OUTPUT);
 pinMode(A4, OUTPUT);
 pinMode(A5, OUTPUT);



  roller1.attach(11);
  roller2.attach(10);

  batt_dispenser.attach(9);
  batt_pusher.attach(6);

  rotator.attach(5);
  dumper.attach(3);

   Serial.begin(9600);

  base_state();
}

void base_state(){

  roller1.write(90);
  roller2.write(90);
  batt_dispenser.write(0);
  batt_pusher.write(20);
   delay(2000);
  rotator.write(45);
  dumper.write(90);

}

int get_state(){

//  int pin0 = digitalRead(0);
//  int pin1 = digitalRead(1);

  int pin2 = digitalRead(2);
  return pin2; //(pin2 << 2) | (pin1 << 1) | pin0;
}

void loop() {


  digitalWrite(A5, HIGH);
  digitalWrite(A4, HIGH);
  digitalWrite(12, HIGH);

    Serial.print(" sensor7 = ");
  Serial.println(analogRead(A7) );



  if (digitalRead(2)){
     new_light = analogRead(A7);
     roller_count = 0;

if (abs(new_light- old_light)   > 100){
  old_light = new_light;
  delay(2000);
```

```
  base_state();
  // sort complete

} else{



        if (digitalRead(8) == LOW){

           roller_count++;
           if (roller_count > 0){
               rollers_fwd();

           } else if (roller_count >= 50 && roller_count < 100 ){
               roller1.write(0);
               roller2.write(90);

           } else if (roller_count >= 100 && roller_count < 150){
               roller1.write(90);
               roller2.write(180);

           } else if (roller_count >= 150 && roller_count < 200){
               roller1.write(90);
               roller2.write(180);
           } else if (roller_count >= 200 && roller_count < 250){
               rollers_bck() ;
           }


          batt_dispenser.write(0);
          batt_pusher.write(10);
          rotator.write(45);
          dumper.write(90);


          jam_loop_count = jam_loop_count + 1;

          if (jam_loop_count > 500){
                  rollers_bck();
                  delay(5000);
                  rollers_fwd();
                  jam_loop_count = 0;
          }


        } else {


          rollers_stop();


            rotator.write(45);
            dumper.write(90);
             dispense();

            jiggle();


              hold_batt();

              batt_type = read_voltage();

              release_batt();

               send_to_pic(batt_type);
```

```
              dump_container(batt_type);

              load();

          }
        }
  }
}

void rollers_fwd() {
   roller1.write(45);
   roller2.write(135);


}

void rollers_stop(){
    roller1.write(90);
    roller2.write(90);

}


void rollers_bck() {
  roller1.write(180);
  roller2.write(0);

}
void dispense() {
 delay(1000);
    batt_dispenser.write(100);

}

void load(){
     delay(1000);
    batt_dispenser.write(0);
  }

void hold_batt() {
  batt_pusher.write(160);

}

void jiggle(){

    for (int j = 0; j > 3; j++){
      delay(500);
      dumper.write(100);
      rotator.write(10);

      batt_pusher.write(90);
      delay(200);
      dumper.write(80);
      rotator.write(90);
    }
  }

void release_batt(){
  batt_pusher.write(20);
  }


int read_voltage(){
```

```
  float volt0 = 0;
  float volt1 = 0;
  float volt2 = 0;
  float volt3 = 0;


  int i = 0;

  delay(1000); // allow

 while (i < 10){

   volt0 += analogRead(A0) * 5.0 / 1023.0;
  volt1 += analogRead(A1) * 5.0 / 1023.0;
  volt2 += analogRead(A2) * 5.0 / 1023.0;
  volt3 += analogRead(A3) * 5.0 / 1023.0;

  i += 1;

 }

  volt0 = volt0/i;
  volt1 = volt1/i;
  volt2 = volt2/i;
  volt3 = volt3/i;



   // 9V - red orage
   if ( volt0 > 7.65/2 || volt1 >  7.65/2 || volt2 >  7.65/2  ){
    return 3; // 9V
   }
   // C - yellow
   if (volt3 > 1.27/2 ){


    return 1; // C
   }

   // AA - red orage
   if ( (volt0 > 1.27/2 && volt0 < 2/2) || (volt1 > 1.27/2 && volt1 < 2/2)){
    return 2; // AA
   }

   else {
    return 4; // 0V
   }


//  Serial.print(sensorValue);
  delay(2);
}

void dump_container(int bin) {

  switch (bin){
    case 1:
      rotator.write(0);
      delay(500);
      dumper.write(3);
      break;
    case 2:
      rotator.write(0);
      delay(500);
      dumper.write(180);
      break;
```

```
    case 3:
      rotator.write(90);
      delay(500);
      dumper.write(3);
      break;
    case 4:
      rotator.write(90);
      delay(500);
      dumper.write(180);
      break;
  }

  delay(500);
  rotator.write(45);
  dumper.write(90);
}
void send_to_pic(int bin) {

//uses high I/O

  switch (bin){

    case 1:
      digitalWrite(A4, LOW);
      digitalWrite(A5, HIGH);
      break;

    case 2:
      digitalWrite(A4, HIGH);
      digitalWrite(A5, LOW);

      break;

    case 3:
      digitalWrite(A4, LOW);
      digitalWrite(A5, LOW);
      break;

    case 4:
      digitalWrite(A4, HIGH);
      digitalWrite(A5, HIGH);
      break;
  }



      delay(500);
      digitalWrite(12, LOW);


      delay(500);

      digitalWrite(12, HIGH);
      digitalWrite(A4, HIGH);
      digitalWrite(A5, HIGH);
}
```

init_bsm.c

```
#include <xc.h>
#include "1_initialize_bsm.h"
#include "I2C.h"
#include "lcd.h"

void initialize(void){
```

```
    init_TRIS();
    init_LAT();
    initLCD();

    init_RTC();
    init_INT();

    ADCON0 = 0x00;  //Disable ADC to use pins
    ADCON1 = 0xFF;  //Set PORTB to be digital instead of analog default

}

void init_TRIS(void){
/*  Configuring the data direction of pins (as input = 1 or output = 0)
 */

    TRISAbits.RA0 = 0; // LED Red
    TRISAbits.RA1 = 1; // Arduino data ready
    TRISAbits.RA2 = 0; // LED Green
    TRISAbits.RA3 = 0;
    TRISAbits.RA4 = 0; // LED Blue
    TRISAbits.RA5 = 1; // Arduino battery type low bit
    TRISAbits.RA6 = 0; // OSC
    TRISAbits.RA7 = 0; // OSC

    TRISBbits.RB0 = 1; // Arduino stop sort trigger
    TRISBbits.RB1 = 1; // INT1: Keypad Interrupt
    TRISBbits.RB2 = 1; // INT2: Emergency Stop
    TRISBbits.RB3 = 1;
    TRISBbits.RB4 = 1; // Keypad Data Transfer
    TRISBbits.RB5 = 1; // Keypad Data Transfer
    TRISBbits.RB6 = 1; // Keypad Data Transfer
    TRISBbits.RB7 = 1; // Keypad Data Transfer

    TRISCbits.RC0 = 0;
    TRISCbits.RC1 = 0;
    TRISCbits.RC2 = 0;
    TRISCbits.RC3 = 1; // SCL: RTC I2C sync (input)
    TRISCbits.RC4 = 1; // SDI: RTC input
    TRISCbits.RC5 = 0;
    TRISCbits.RC6 = 0;
    TRISCbits.RC7 = 0; // Arduino start sort

    TRISDbits.RD0 = 0;
    TRISDbits.RD1 = 0;
    TRISDbits.RD2 = 0; // LCD Read/Write Trigger
    TRISDbits.RD3 = 0; // LCD Enable Bit
    TRISDbits.RD4 = 0; // LCD Data Transfer
    TRISDbits.RD5 = 0; // LCD Data Transfer
    TRISDbits.RD6 = 0; // LCD Data Transfer
    TRISDbits.RD7 = 0; // LCD Data Transfer

// PortE
    TRISEbits.RE0 = 0;
    TRISEbits.RE1 = 1; //Arduino battery type high bit
    TRISEbits.RE2 = 0;
}

void init_LAT(void){
/* Configure the output status of pins*/

    // Set all pins to not send output voltages
    LATA = 0x00;
    LATB = 0x00;
    LATC = 0x00;
    LATD = 0x00;
```

```
    LATE = 0x00;
}

void init_RTC(void){
    I2C_Master_Init(10000); //Initialize I2C Master with 100KHz clock
}

void init_INT(void){

    INT1IE = 1;
    INT2IE = 1;
    TMR0IE = 1;

    ei();//Enable all interrupts
}

void init_TMR0(void){

    T0CS=0; //Prescaler gets clock from internal oscillator (8Mhz)
    T08BIT=0; //16 BIT MODE
    PSA=0;  //Timer Clock Source is from Prescaler

    //Prescaler is divide by 256
    T0PS0=0;
    T0PS1=1;
    T0PS2=0;

    //setting timer1 : high bit first
    TMR0H=0x0;//0b10000101;;
    TMR0L=0x0;//0b11101110;
  }
```

bsm_data.c (adapted from sample code)

```
#include <xc.h>
#include <stdio.h>

#include "macros.h"

#include "bsm_data.h"
#include "lcd.h"

void write_data_bsm( unsigned char start, unsigned char data[12]){

    int i = 0;
    unsigned char address = start;
    unsigned char value;

    while (i < 12){
        Eeprom_WriteByte(address, data[i]);
        address = address + 1;
        i++;
    }

}

void read_data_bsm( unsigned char start, unsigned char* data){

    int i = 0;
    unsigned char address = start;
    unsigned char value;
    while (i < 12){
        data[i] = Eeprom_ReadByte(address);
```

```
            address = address + 1;
            i++;
        }

}
// A fix to the EEPROM C code from microchip forums:
// http://www.microchip.com/forums/m38720.aspx
//=============================================================================
unsigned char Eeprom_ReadByte(unsigned short address)
{

    // Set address registers
    EEADRH = (unsigned char)(address >> 8);
    EEADR = (unsigned char)address;

    EECON1bits.EEPGD = 0;       // Select EEPROM Data Memory
    EECON1bits.CFGS = 0;        // Access flash/EEPROM NOT config. registers
    EECON1bits.RD = 1;          // Start a read cycle

    // A read should only take one cycle, and then the hardware will clear
    // the RD bit
    while(EECON1bits.RD == 1);

    return EEDATA;              // Return data

}

void Eeprom_WriteByte(unsigned short address, unsigned char data)
{
    // Set address registers
    EEADRH = (unsigned char)(address >> 8);
    EEADR = (unsigned char)address;

    EEDATA = data;          // Write data we want to write to SFR
    EECON1bits.EEPGD = 0;   // Select EEPROM data memory
    EECON1bits.CFGS = 0;    // Access flash/EEPROM NOT config. registers
     EECON1bits.WREN = 1;    // Enable writing of EEPROM (this is disabled again after the
write completes)

    // The next three lines of code perform the required operations to
    // initiate a EEPROM write
    EECON2 = 0x55;          // Part of required sequence for write to internal EEPROM
    EECON2 = 0xAA;          // Part of required sequence for write to internal EEPROM
    EECON1bits.WR = 1;      // Part of required sequence for write to internal EEPROM

    // Loop until write operation is complete
    while(PIR2bits.EEIF == 0)
    {
        continue;   // Do nothing, are just waiting
    }

     PIR2bits.EEIF = 0;       //Clearing EEIF bit (this MUST be cleared in software after
each write)
      EECON1bits.WREN = 0;     // Disable write (for safety, it is re-enabled next time a
EEPROM write is performed)
}
// ============================================================================
```

I2C.c (adapted from sample code)

```
#include <xc.h>
#include "I2C.h"
#include "configBits.h"

const char happynewyear[7] = {  0x30, //45 Seconds
```

```
                        0x53, //59 Minutes
                        0x11, //24 hour mode, set to 23:00
                        0x07, //Saturday
                        0x04, //31st
                        0x02, //December
                        0x17};//2016
void set_time(void);
void set_time(void){
    I2C_Master_Start(); //Start condition
    I2C_Master_Write(0b11010000); //7 bit RTC address + Write
    I2C_Master_Write(0x00); //Set memory pointer to seconds
    for(char i=0; i<7; i++){
        I2C_Master_Write(happynewyear[i]);
    }
    I2C_Master_Stop(); //Stop condition
}


void I2C_Master_Init(const unsigned long c)
{
  // See Datasheet pg171, I2C mode configuration
  SSPSTAT = 0b00000000;
  SSPCON1 = 0b00101000;
  SSPCON2 = 0b00000000;
  SSPADD = (_XTAL_FREQ/(4*c))-1;
  TRISC3 = 1;        //Setting as input as given in datasheet
  TRISC4 = 1;        //Setting as input as given in datasheet
}

void I2C_Master_Wait()
{
  while ((SSPSTAT & 0x04) || (SSPCON2 & 0x1F));
}

void I2C_Master_Start()
{
  I2C_Master_Wait();
  SEN = 1;
}

void I2C_Master_RepeatedStart()
{
  I2C_Master_Wait();
  RSEN = 1;
}

void I2C_Master_Stop()
{
  I2C_Master_Wait();
  PEN = 1;
}

void I2C_Master_Write(unsigned d)
{
  I2C_Master_Wait();
  SSPBUF = d;
}

unsigned char I2C_Master_Read(unsigned char a)
{
  unsigned char temp;
  I2C_Master_Wait();
  RCEN = 1;
  I2C_Master_Wait();
  temp = SSPBUF;
  I2C_Master_Wait();
```

```
  ACKDT = (a)?0:1;
  ACKEN = 1;
  return temp;
}

void delay_10ms(unsigned char n) {
    while (n-- != 0) {
        __delay_ms(5);
    }
}
```

lcd.c (adapted from sample code)

```
#include <xc.h>
#include "configBits.h"
#include <stdio.h>
#include "lcd.h"
#include "constants.h"
#include "macros.h"

void initLCD(void) {
    __delay_ms(15);
    lcdInst(0b00110011);
    lcdInst(0b00110010);
    lcdInst(0b00101000);
    lcdInst(0b00101000);
    lcdInst(0b00001100);
    lcdInst(0b00000110);
    lcdInst(0b00000001);
    __delay_ms(15);
}

void lcdInst(char data) {
    RS = 0;
    lcdNibble(data);
}

void putch(char data){
    RS = 1;
    lcdNibble(data);
}

void lcdNibble(char data){
    // Send of 4 most sig bits, then the 4 least sig bits (MSD,LSD)
    char temp = data & 0xF0;
    LATD = LATD & 0x0F;
    LATD = temp | LATD;

    E = 0;
    __delay_us(LCD_DELAY);
    E = 1;
    __delay_us(LCD_DELAY);

    data = data << 4;

    temp = data & 0xF0;
    LATD = LATD & 0x0F;
    LATD = temp | LATD;

    E = 0;
    __delay_us(LCD_DELAY);
    E = 1;
    __delay_us(LCD_DELAY);
}
void print_lcd20(char* msg){
```

```
    int i=0;
    while (i<20){
        char c = msg[i];
        if (c == '\0')break;
        putch(c);
        i++;
    };

}


void print_lcd(char* msg1, char* msg2, char* msg3, char* msg4){

clear_lcd();
    __lcd_l1();
    print_lcd20(msg1);

    __lcd_l2();
    print_lcd20(msg2);

    __lcd_l3();
    print_lcd20(msg3);

    __lcd_l4();
    print_lcd20(msg4);

}

void clear_lcd(){

    char* msg = "                    ";
    __lcd_l1();
    print_lcd20(msg);

    __lcd_l2();
    print_lcd20(msg);

    __lcd_l3();
    print_lcd20(msg);

    __lcd_l4();
    print_lcd20(msg);

}

void print_lcd_long(char* msg){
    int i=0;
    int line = 1;
    char c = 0;

    while (i<80){
        c = msg[i];
        if (c == '\0')break;


        if (i % 20 == 0){
            switch(line){
                case 1:
                    lcdInst(0b10000000);
                    break;

                case 2:
                    lcdInst(0b11000000);
                    break;

                case 3:
```

```
                lcdInst(0b10010100);
                break;

            case 4:
                lcdInst(0b11010100);
                break;
        };
        line++;
    };

    putch(c);

    i++;
  };
}
```

macros.h

```
#ifndef MACROS_H
#define MACROS_H

#define __delay_1s() for(char i=0;i<10;i++){__delay_ms(98);}
#define __lcd_newline() lcdInst(0b11000000);// 0/1 => secondrow/thridrow, thirdrow

#define __lcd_l1() lcdInst(0b10000000);
#define __lcd_l2() lcdInst(0b11000000);
#define __lcd_l3() lcdInst(0b10010100);
#define __lcd_l4() lcdInst(0b11010100);

#define __lcd_clear() lcdInst(0b1);
#define __lcd_home() lcdInst (0b10000000);
#define __bcd_to_num(num) (num & 0x0F) + ((num & 0xF0)>>4)*10;


#endif  /* MACROS_H */
```

constants.h

```
#ifndef CONSTANTS_H
#define CONSTANTS_H         //Prevent multiple inclusion

//LCD Control Registers
#define RS          LATDbits.LATD2
#define E           LATDbits.LATD3
#define LCD_PORT    LATD   //On LATD[4,7] to be specific
#define LCD_DELAY   25


#endif  /* CONSTANTS_H */
```

configBits.h

```
// PIC18F4620 Configuration Bit Settings

// 'C' source line config statements

// CONFIG1H
#pragma config OSC = HS         // Oscillator Selection bits (HS oscillator)
#pragma config FCMEN = OFF        // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock
Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover bit (Oscillator
Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF       // Power-up Timer Enable bit (PWRT disabled)
```

```
        #pragma config BOREN = SBORDIS  // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
        #pragma config BORV = 3         // Brown Out Reset Voltage bits (Minimum setting)

        // CONFIG2H
        #pragma config WDT = OFF        // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
        #pragma config WDTPS = 32768    // Watchdog Timer Postscale Select bits (1:32768)

        // CONFIG3H
        #pragma config CCP2MX = PORTC   // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
        #pragma config PBADEN = ON      // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
        #pragma config LPT1OSC = OFF    // Low-Power Timer1 Oscillator Enable bit (Timer1 configured for higher power operation)
        #pragma config MCLRE = ON       // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

        // CONFIG4L
        #pragma config STVREN = ON      // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
        #pragma config LVP = OFF        // Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
        #pragma config XINST = OFF      // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))

        // CONFIG5L
        #pragma config CP0 = OFF        // Code Protection bit (Block 0 (000800-003FFFh) not code-protected)
        #pragma config CP1 = OFF        // Code Protection bit (Block 1 (004000-007FFFh) not code-protected)
        #pragma config CP2 = OFF        // Code Protection bit (Block 2 (008000-00BFFFh) not code-protected)
        #pragma config CP3 = OFF        // Code Protection bit (Block 3 (00C000-00FFFFh) not code-protected)

        // CONFIG5H
        #pragma config CPB = OFF        // Boot Block Code Protection bit (Boot block (000000-0007FFh) not code-protected)
        #pragma config CPD = OFF        // Data EEPROM Code Protection bit (Data EEPROM not code-protected)

        // CONFIG6L
        #pragma config WRT0 = OFF       // Write Protection bit (Block 0 (000800-003FFFh) not write-protected)
        #pragma config WRT1 = OFF       // Write Protection bit (Block 1 (004000-007FFFh) not write-protected)
        #pragma config WRT2 = OFF       // Write Protection bit (Block 2 (008000-00BFFFh) not write-protected)
        #pragma config WRT3 = OFF       // Write Protection bit (Block 3 (00C000-00FFFFh) not write-protected)

        // CONFIG6H
        #pragma config WRTC = OFF       // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) not write-protected)
        #pragma config WRTB = OFF       // Boot Block Write Protection bit (Boot Block (000000-0007FFh) not write-protected)
        #pragma config WRTD = OFF       // Data EEPROM Write Protection bit (Data EEPROM not write-protected)

        // CONFIG7L
        #pragma config EBTR0 = OFF      // Table Read Protection bit (Block 0 (000800-003FFFh) not protected from table reads executed in other blocks)
        #pragma config EBTR1 = OFF      // Table Read Protection bit (Block 1 (004000-007FFFh) not protected from table reads executed in other blocks)
```

```
        #pragma config EBTR2 = OFF      // Table Read Protection bit (Block 2 (008000-00BFFFh) not
protected from table reads executed in other blocks)
        #pragma config EBTR3 = OFF      // Table Read Protection bit (Block 3 (00C000-00FFFFh) not
protected from table reads executed in other blocks)

        // CONFIG7H
        #pragma config EBTRB = OFF       // Boot Block Table Read Protection bit (Boot Block
(000000-0007FFh) not protected from table reads executed in other blocks)


        #include <xc.h>



        #include <string.h>

        #define _XTAL_FREQ 10000000      // Define osc freq for use in delay macros
```

Appendix C: Detailed Budget

Item Description	Item Code	Shop	Number of Items	Cost per Item	Total Cost	Link
Development Board						
PIC DevBugger Development Board	N/A	AER201	1	$50.00	$50.00	N/A
Keyboard	N/A	AER201	1	$3.00	$3.00	N/A
RTC Chip and Coin Battery	N/A	AER201	1	$5.00	$5.00	N/A
20x4 LCD	N/A	Amazon	1	$6.19	$6.19	https://www.amazon.ca/SODIAL-Character-Display-Backlight-Arduino/dp/B00L8VCHJC/ref=sr_1_2?s=electronics&ie=UTF8&qid=1491709987&sr=1-2&keywords=20x4+lcd
SODIAL(R) Red Mushroom Cap 1NO 1NC DPST Emergency Stop Push Button Switch AC 660V 10A	N/A	Amazon	1	$3.20	$3.20	https://www.amazon.ca/SODIAL-Mushroom-Emergency-Button-Switch/dp/B00H3CY432/ref=sr_1_2?s=hi&ie=UTF8&qid=1491710879&sr=1-2&keywords=emergency+stop+button
PIC Pin Board						
10 Pcs 2x20 Pin 2.0mm Pitch Double Row IDC Pin Headers Connectors	N/A	Amazon	0.1	$6.97	$0.70	https://www.amazon.ca/2-0mm-Pitch-Double-Headers-Connectors/dp/B00899WMRI/ref=sr_1_fkmr2_1?s=hi&ie=UTF8&qid=1491710807&sr=8-1-fkmr2&keywords=40+PIN+%2820X2%29+IDC+SHROUDED+HEADER
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
Arduino Nano Board						
XCSOURCE 5pcs Mini USB Nano V3.0 ATmega328P 5V 16M Micro Controller Board F Arduino TE359	N/A	Amazon	0.2	$25.99	$5.20	https://www.amazon.ca/XCSOURCE-ATmega328P-Controller-Arduino-TE359/dp/B015MGHH6Q/ref=sr_1_1?ie=UTF8&qid=1491708579&sr=8-1&keywords=Arduino+nano
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
SODIAL(R) 50 Pcs 103 10K ohm 3296W Trim Pot Trimmer Potentiometer 25 Turn	N/A	Amazon	0.02	$5.53	$0.11	https://www.amazon.ca/SODIAL-3296W-Trim-Trimmer-Potentiometer/dp/B00SUVREIC/ref=sr_1_sc_1?s=electronics&ie=UTF8&qid=1491710074&sr=1-1-spell&keywords=timmer+potentiometer
Power Board						
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
20pcs 5V 0.3 A Mini Size Black SPDT Slide Switch for Small DIY Power Electronic Projects	N/A	Amazon	0.3	$1.96	$0.59	https://www.amazon.ca/20pcs-Black-Switch-Electronic-Projects/dp/B00BIAVQSG/ref=sr_1_1?s=hi&ie=UTF8&qid=1491710181&sr=1-1&keywords=small+slide+switches
Mr.Geeker 10 Pcs Male Power Adapter DC Barrel to Screw Plug Jack Connector 2.1 x 5.5MM	N/A	Amazon	0.1	$9.99	$1.00	https://www.amazon.ca/Mr-Geeker-Power-Adapter-Barrel-Connector/dp/B01N75YG6H/ref=pd_sbs_147_1?_encoding=UTF8&psc=1&refRID=6RRHT0DSM8DCJC3QZH76
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.01	$17.98	$0.18	https://www.amazon.ca/Gikfun-Yellow-Assorted-Arduino-300pcs/dp/B01N527GKZ/ref=sr_1_11?s=electronics&ie=UTF8&qid=1491708879&sr=1-11&keywords=green+led
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.02	$17.98	$0.36	https://www.amazon.ca/Gikfun-Yellow-Assorted-Arduino-300pcs/dp/B01N527GKZ/ref=sr_1_11?s=electronics&ie=UTF8&qid=1491708879&sr=1-11&keywords=green+led
1/4W 5% RESISTOR (10 PACK) 220 Ohms	RESIS-500025	Creatron Inc.	0.3	$0.25	$0.08	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11

1/4W 5% RESISTOR (10 PACK) 10k Ohm Ohms	RESIS-500025	Creatron Inc.	0.7	$0.25	$0.18	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
Rectifier Board						
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
1N914 - 100V 0.2A RECTIFIER DIODE (5 PACK)	DIODE-000914	Creatron Inc.	0.8	$0.35	$0.28	https://www.creatroninc.com/product/1n914-100v-02a-rectifier-diode-5-pack/?search_query=diode&results=72
IC OPAMP GP 50KHZ RRO 8DIP	LMC6462BIN	DigiKey Inc.	4	$4.24	$16.96	https://www.digikey.ca/product-detail/en/texas-instruments/LMC6462BIN-NOPB/LMC6462BIN-NOPB-ND/364327
1/4W 5% RESISTOR (10 PACK) 200k Ohm Ohms	RESIS-500025	Creatron Inc.	0.8	$0.25	$0.20	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
Voltage Divider Board						
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
1/4W 5% RESISTOR (10 PACK) 1k Ohms	RESIS-500025	Creatron Inc.	0.8	$0.25	$0.20	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
Limit Switch Board						
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
1/4W 5% RESISTOR (10 PACK) 220 Ohms	RESIS-500025	Creatron Inc.	0.1	$0.25	$0.03	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
1/4W 5% RESISTOR (10 PACK) 10k Ohms	RESIS-500025	Creatron Inc.	0.1	$0.25	$0.03	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
Gikfun 3mm 5mm LEDs Light White Yellow Red Green Blue Assorted Kit For Arduino DIY (Pack of 300pcs) EK8453	N/A	Amazon	0.006666666667	$17.98	$0.12	https://www.amazon.ca/Gikfun-Yellow-Assorted-Arduino-300pcs/dp/B01N527GKZ/ref=sr_1_11?s=electronics&ie=UTF8&qid=1491708879&sr=1-11&keywords=green+led
LED Control Board						
Ocr ™16PCS PCB Board Universal Double Sided Prototyping Breadboard Panel Multiple Sizes	N/A	Amazon	1	$0.80	$0.80	https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/?search_query=20x2+pin&results=13
TRANS NPN DARL 60V 5A TO220AB	TIP120GOS-ND	Digikey Inc.	3	$0.82	$2.46	https://www.digikey.ca/product-detail/en/on-semiconductor/TIP120G/TIP120GOS-ND/920325
1/4W 5% RESISTOR (10 PACK) 10k Ohms	RESIS-500025	Creatron Inc.	0.3	$0.25	$0.08	https://www.creatroninc.com/product/14w-5-resistor-10-pack/?search_query=220+ohm+resistor&results=11
10 Pcs DC 5V Coil 7A 240VAC 10A 125VAC/28VDC 5 Pins SPST Power Relay JQC-3F	N/A	Amazon	0.3	$4.94	$1.48	https://www.amazon.ca/240VAC-125VAC-28VDC-Power-JQC-3F/dp/B008SO6BDK/ref=sr_1_1?s=hi&ie=UTF8&qid=1491711055&sr=1-1&keywords=5V+relay
Mr.Geeker 10 Pcs Male Power Adapter DC Barrel to Screw Plug Jack Connector 2.1 x 5.5MM	N/A	Amazon	0.1	$9.99	$1.00	https://www.amazon.ca/Mr-Geeker-Power-Adapter-Barrel-Connector/dp/B01N75YG6H/ref=pd_sbs_147_1?_encoding=UTF8&psc=1&refRID=6RRHT0DSM8DCJC3QZH76
12 inch RGB LED Strip	N/A	Creatron	2	$5.99	$11.98	N/A
Actuators						
9g Continuous Rotation Micro Servo	RB-Fit-02	RobotShop	4	$6.65	$26.60	http://www.robotshop.com/ca/en/9g-continuous-rotation-micro-servo.html
RioRand® 5PCS x SG90 Micro 9g Servo For RC Airplane Car Boat Genuine	N/A	Amazon	0.6	$16.99	$10.19	http://www.robotshop.com/ca/en/dfrobot-micro-servo-motor.html

SODIAL(R) Long Straight Hinge Lever 3 Pins Basic NO NC Momentary Micro Switch 2 Pcs	N/A	Amazon	0.5	$2.10	$1.05	https://www.amazon.ca/SODIAL-Straight-Hinge-Momentary-Switch/dp/B00JFOQZEM/ref=sr_1_12?s=hi&ie=UTF8&qid=1491710298&sr=1-12&keywords=microswitch
VS-19 Pico Linear Servo	RB-Sbo-109	RobotShop	1	$12.95	$12.95	http://www.robotshop.com/ca/en/vs-19-pico-linear-servo.html
Stuctural Material						
8X24X.025 Aluminum Sheet Metal	142-550	Home Depot	1	$13.28	$13.28	https://www.homedepot.ca/en/home/p.8x24x025-aluminum-sheet-metal.1000126786.html
1/4 inch x 2 Feet x 2 Feet Birch Plywood Handy Panel	$621,615.00	Home Depot	1	$5.68	$5.68	https://www.homedepot.ca/en/home/p.14-inch-x-2-feet-x-2-feet-birch-plywood-handy-panel.1000434557.html
Flexible Wood (from Home Hardware)	N/A	Home Hardware	1	$3.68	$3.68	N/A
Small Acrylic Sheet	N/A	Home Depot	2	$2.00	$4.00	N/A
2-1/2 Inch Zinc Broad Hinge	859-110	Home Depot	2	$1.98	$3.96	https://www.homedepot.ca/en/home/p.2-12-inch--zinc-broad-hinge.1000773701.html
4-40 x 3/8 Inch Phillips Truss Head Machine Screws Fasteners 50 Pcs	N/A	Amazon	0.8	$5.49	$4.39	https://www.amazon.ca/Phillips-Truss-Machine-Screws-Fasteners/dp/B0143DZHQQ/ref=sr_1_5?s=hi&ie=UTF8&qid=1491711837&sr=1-5&keywords=machine+screws
SODIAL(R) Metric M3x0.5mm Stainless Steel Finished Hex Nut Silver Tone 50pcs	N/A	Amazon	0.8	$2.09	$1.67	https://www.amazon.ca/SODIAL-Metric-M3x0-5mm-Stainless-Finished/dp/B01GNVTSOS/ref=sr_1_39?s=hi&ie=UTF8&qid=1491711752&sr=1-39&keywords=machine+screws
SODIAL(R) 20mm x 15mm Metal Corner Brace Joint Right Angle Bracket Silver Tone 20Pcs	N/A	Amazon	1	$2.85	$2.85	https://www.amazon.ca/SODIAL-Metal-Corner-Bracket-Silver/dp/B00SUVP8Q2/ref=sr_1_cc_6?s=aps&ie=UTF8&qid=1491711651&sr=1-6-catcorr&keywords=metal+braces
PVC Pipe	CPLG-100	Home Depot	1	$0.77	$0.77	https://www.homedepot.ca/en/home/p.schedule-40-pvc-coupling--1-inch.1000100862.html
Miscellaneous Components						
1x40 Pins Male 2.54 mm Pitch Single Row Pin Header Strip 10 Pcs	N/A	Amazon	0.1	$2.36	$0.24	https://www.amazon.ca/1x40-Pitch-Single-Header-Strip/dp/B00HR8NCIK/ref=sr_1_4?s=electronics&ie=UTF8&qid=1491711194&sr=1-4&keywords=male+pins
10 Pcs 1x40 Pin 2.54mm Pitch Straight Single Row PCB Female Pin Headers	N/A	Amazon	0.2	$2.78	$0.56	https://www.amazon.ca/2-54mm-Straight-Single-Female-Headers/dp/B00899WQ6U/ref=sr_1_1?s=electronics&ie=UTF8&qid=1491711130&sr=1-1&keywords=female+pins
Jumper Wires Premium 6" F / F Pack of 20	RB-Ada-170	RobotShop	1	$2.60	$2.60	http://www.robotshop.com/ca/en/jumper-wires-premium-6-f-f-pack-of-20.html
PK-Power AC Adapter for DVE Switching Model DSA-24CA-05 050400 5V 4A Power Supply	N/A	Amazon	1	$7.82	$7.82	https://www.amazon.ca/PK-Power-Adapter-Switching-DSA-24CA-05-050400/dp/B01MCUSBKC/ref=sr_1_1?s=electronics&ie=UTF8&qid=1491709501&sr=1-1&keywords=5v+4a+power+supply
HDE US Plug AC100-240V to DC 12V 2A Power Supply for 3528 Flexible LED Light Strips	N/A	Amazon	1	$3.95	$3.95	https://www.amazon.ca/HDE-AC100-240V-Supply-Flexible-Strips/dp/B00KQV1912/ref=sr_1_2?s=electronics&ie=UTF8&qid=1491709250&sr=1-2&keywords=12v+2a+power+supply
Extension Cord	N/A	Amazon	1	$2.50	$2.50	https://www.amazon.ca/ALEKO%C2%AE-ECI2O6FT-Indoor-Extension-Approved/dp/B01IPTMAYG/ref=sr_1_3?ie=UTF8&qid=1491941181&sr=8-3&keywords=extension+cord+white
Total					$224.92	

Appendix C: Material Properties

Birch Plywood

Aluminum

Carbon Fibre Composite

Appendix E: Datasheets

Linear Servo Motor

Parallax (Futaba) Continuous Rotation Servo

RGB LED Strips

MicroServo

Relay

LMC6462

TIP120

Interlink Electronics 1.5” Square FSR

Battery Specifications