UNIVERSITY OF TORONTO

INSTITUTE FOR AEROSPACE STUDIES

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

# BATTERY RECYCLING MACHINE

*prepared by*

**Team 07 – Monday**

Omar Hegazi (1002311116)
William Kingsford (999914222)
Michael Xu (1002394820)

*prepared for*

Prof. M.R. Emami

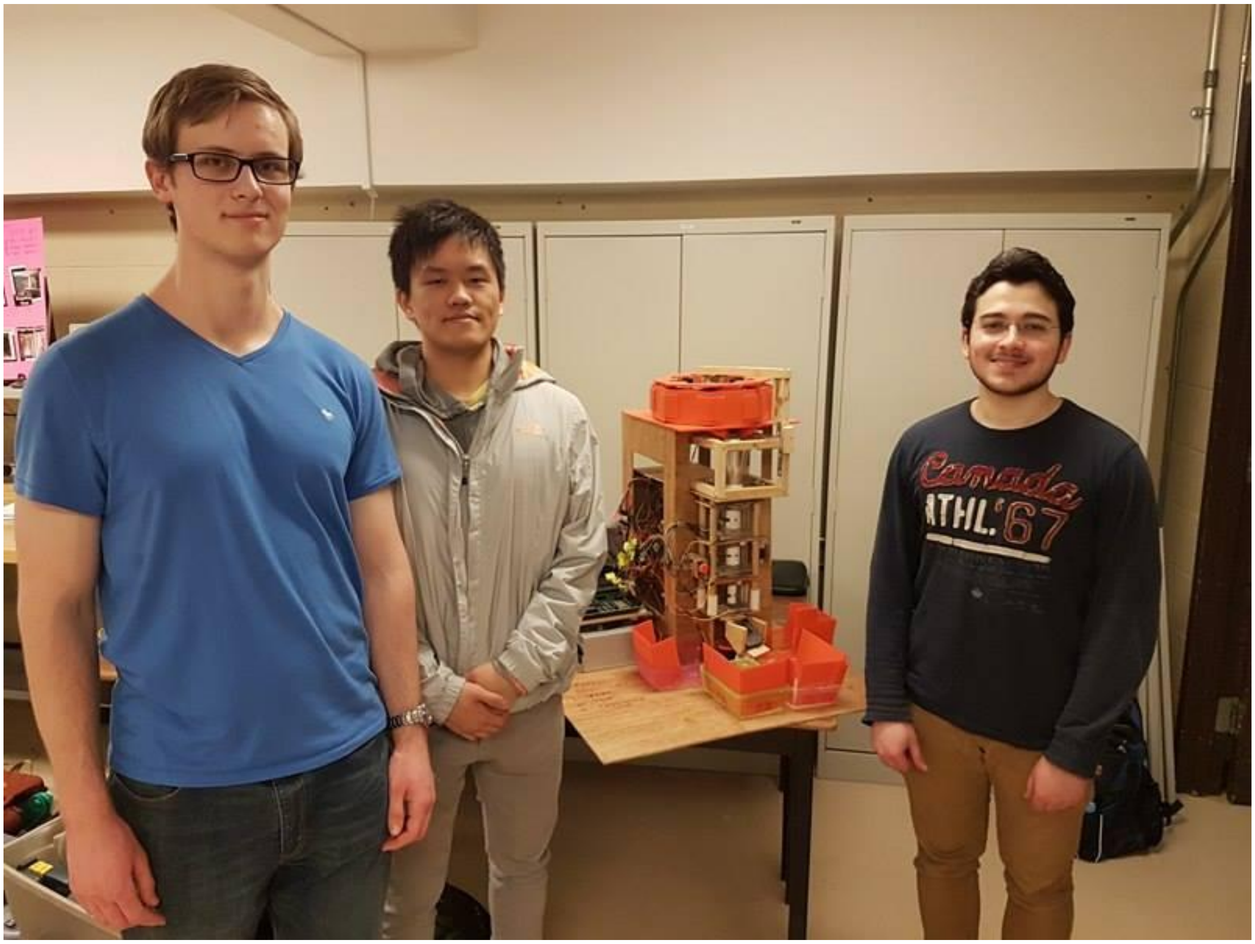
A technical report submitted for
AER201 – Engineering Design

TA: Nathan Cole

April 13, 2017

# Battery Recycling Machine
# Final Report

April 13, 2017




AER201 - Engineering Design

Professor: Dr. Reza Emami

TA: Nathan Cole

Team 7

Omar Hegazi, William Kingsford, Michael Xu

# 1. Acknowledgement

We would like to thank all of those who helped us in this course. In particular, Professor Reza Emami for organizing the course and encouraging us to take a professional approach to the design, our TA Nathan Cole for his suggestions for mechanical designs, and Colin Harry for his enormous assistance in accurately fabricating our design and for allowing us to borrow power tools outside of the machine shop when we needed them to make deadlines.

# 2. Abstract

This report presents a design for an autonomous machine for sorting a set of batteries based on their type and whether or not they are charged. The primary objective of the design is to accurately sort the batteries, with secondary goals of speed, reliability and maintainability. The high-level design of the machine is to have a rotating loading bay at the top of the machine, with battery-sized slots embedded into its floor. This causes batteries to be dropped one at a time into a mechanical system which sorts them based on size, then brings them to electrodes for their voltages to be measured. Finally, the batteries are dropped into separate containers depending on their type and charge level.

# 3. Table of Contents

# 4. Symbols and Abbreviations

**CAD:** Computer-Aided Design

**FSM**: Finite State Machine

**LCD**: Liquid Crystal Display

**PIC**: Peripheral Interface Controller

**PWM**: Pulse Width Modulation

# 5. Introduction

This technical report will outline the team's design process and implemented solution to RFP #1: “The Battery Recycling Machine” from the AER201 Engineering Design course. The RFP asked for a design of an autonomous machine capable of taking a set of randomly mixed 9V, AA and C batteries and sorting each battery into a different category based on whether it is discharged, a charged AA, a charged C, or a charged 9V.

The first step was to formulate the problem in terms of clearly defined technical requirements. After this, the team's overarching design goals of modularity, reliability, and accuracy were established. These informed all of the team's subsequent design choices, and the team paired them with a set of metrics in order to accurately determine how well a given option aligns with the stated values.

Next, this report outlines the research carried out by the team in order to be well-informed of existing methods to solve this type of battery sorting problem. In addition, the report presents an overview of the prices and specifications of various fundamental components such as motors and electronics in order to be able to make informed judgements of whether or not a candidate solution was feasible within the $230 CAD budget.

Once this background information has been laid out, the report proceeds to describe the most significant design decisions made by the team to come to the design selected. Naturally there were far too many choices to explain and justify all of them, but the aforementioned section gives a good idea of both the wide range of candidate solutions that were considered as well as the thought process that led to the team's final design.

The next section contains first a description of the complete design selected, followed by a detailed technical overview of the system. 3D models of the mechanical system are used to clearly explain the working of the machine, while block diagrams of the circuits and microcontroller code show the operation of the electronic side of the system. These diagrams are supplemented with photographs of the completed prototype of the machine.

Next, the initial and accomplished project schedules are discussed. Gantt charts are used to describe and justify the project timeline. The first milestone we specified is Individual Completion on February 17th, upon which each team member should have finished design and fabrication of each of their separate components. Next is the Subsystem Integration milestone on February 28th, when relevant subsystems must be connected and functioning, for example the electronics should be correctly driving the motors and the microcontroller should be correctly interfacing with the electronics. Finally, the Full System Integration milestone was on March 13th, where the entire machine should be assembled and functioning, allowing for 2 weeks of debugging and verification before the system functionality assessment. After outlining the plan and milestones, division of labour was then determined based on critical path analysis and the specialized role of each team member, and the budget was allocated between the various tasks.

Finally, the results of the project are reviewed. The final design is described and assessed according to the criteria laid out earlier in the report, and potential improvements are discussed.

# 6. Objectives, Constraints, and Acceptance Criteria in Decision Making

## 6.1 Objective

The overall objective of this project is to build an autonomous machine capable of receiving a mixed set of charged and drained 9V, AA and C batteries and sorting them into 4 buckets for drained batteries, charged 9V, charged AA and charged C batteries. It must be able to process up to 15 batteries in a single run. The batteries cannot be manually pre-sorted or aligned in any way before being inserted in the machine, and they must be deposited undamaged into easily removable containers.

## 6.2 Technical Specification

- The machine should follow the process below:
    - Initially the machine should be in standby mode.
    - Once the user has loaded the mixed batteries into the machine, the user can press a <start> button to begin operation.
    - The machine must autonomously sort the batteries in less than 3 minutes, with no user input required during this process.
    - At the end of the process the machine should return to the standby mode, displaying a termination message which states whether the operation was successful or aborted, along with the total number of number of batteries received, the number of each type of battery and the time taken.
- Supplied batteries are of heavy duty (ZnC/ZnCl) or alkaline (non-rechargeable) types, and may be different brands.
- Batteries are taken to be drained if they are below 85% of their nominal rating; otherwise they are taken to be charged.
- The machine should be portable with no need for installations.
- The user is expected to be able to accept user input and give output through a keypad and LCD display.
- The menus should be straightforward and self-explanatory.
- There must be an easily-accessible emergency stop button.
- The machine may be plugged into an AC outlet.

The relevant properties of the types of battery the machine is required to sort is listed below:

**Table 6.2. Properties of the 3 different types of batteries [1, 2, 3]**

| **Battery Type** | **Height** (mm) | **Diameter** (mm) | **Depth** (mm) | **Width** (mm) | **Weight** (g) |
|---|---|---|---|---|---|
| **9-volt** | 48.5 | n/a | 17.5 | 26.5 | 37 (ZnC), 45 (Alkaline) |
| **C** | 50 | 26.2 | n/a | n/a | 48 (ZnC), 65 (Alkaline) |
| **AA** | 49.2 - 50.5 | 13.5 - 14.5 | n/a | n/a | 19 (ZnC), 24 (Alkaline) |

## 6.3 Constraints

- Operation of the machine must be completely autonomous, with the exception of start, stop and pause commands. No interaction with an external PC or remote control is permitted during the operation.
- The entire machine shall fit within a 0.7m x 0.7m x 0.7m envelope at all times during the operation, including the space taken up by the batteries, excluding the power cable.
- The weight of the machine shall not exceed 10kg.
- The total device must not exceed $230 CAD in cost, before shipping and taxes. Foreign currency will be converted to Canadian dollars based on the exchange rate reported by the Central Bank of Canada at the end of the business day of January 9th, 2017. The labor costs are not counted, with the exception of time spent using a 3D printer of CNC machine, in which case labor is counted at $5 CAD per hour. The G-code and exact manufacturing time for such parts must be reported.
- Containers for the sorted batteries must be easily removable and clearly labelled.
- Containers must not contain any power supply, actuator, electronic board or part, such as active sensor etc.
- Loading and unloading the batteries must be convenient for the user, with no need to disassemble any part of the machine.
- The time required for loading the batteries into the machine must not exceed 1 minute. The type and orientation of the batteries must remain undetermined during the loading period.
- Each run is considered “complete” when all batteries are sorted into their designated containers and the display shows a message indicating completion.
- After the operation and unloading of the containers, if a battery is deformed or noticeably scratched then it is considered “damaged”.
- The operation time is the duration between the pressing of the <start> button on the keypad and when the machine shows the completion or termination message on the

LCD. No actuation or sensing must occur prior to the <start> button being pressed, and the total time shall not exceed 3 minutes.
- The recorded and displayed time is considered "correct" if it agrees with the referee's recorded time to within 1 second. Otherwise it is considered "incorrect"
- Each run is "qualified" for scoring if the machine sorts correctly at least 1 battery in each category that is present in the supplied mix within 3 minutes, returns to standby mode so the containers can be unloaded normally, displays the completion or termination message at the end of its operation, and is able to communicate the operation information.
- A run is "disqualified" if the machine structurally collapses, falls over, hangs or jams for more than 3 minutes, or terminates the operation before sorting at least 4 batteries, or does not display the termination or completion message on the LCD at the end of its operation, or runs more than 3 minutes, or takes more than 1 minute to load the batteries into the machine and start the operation, or our team declares the termination. If any of the above happens to the first run, we will have 3 minutes to fix the system and run the next time, if we wish.
- Our team will have a period of at most 2 minutes to set up the machine before it is ready to load the batteries for each run. If the preparation time exceeds 2 minutes then the run is "disqualified".
- There will be no control over the conditions of the contest environment.
- The machine must pose no hazard to the users, and shall not be perceived as hazardous.

## 6.4 Team Goals

The design values of the team were:

1. **Modularity:** the machine should be able to be taken apart into a large number of small, independent units that can be independently tested and/or repaired. In addition, the interface between the circuits and microcontroller should be as small as possible so that integration is straightforward and does not introduce difficulties and additional sources of error.
2. **Reliability:** the machine should be designed to be resilient to a wide range of input conditions and corner cases, and as such a large number of tests of different situations should be carried out.
3. **Accuracy:** the machine should consistently place batteries into the correct containers without error.

In order to judge how a given design aligns with these values, the corresponding metrics were, respectively,

1. **Number of components that cannot be independently tested:** lower is better, with the target of 0. It should not be necessary to exchange information other than start, stop and pause commands from the microcontroller to the circuits, and number of each type of battery and completion signal from the circuits to the microcontroller.

2. **Number of corner-case tests that the system fails under:** must be difficult to produce a corner-case that causes a system failure, with the goal of it being impossible to make the system fail. There should be an emphasis on constructing worst-case inputs such as an input of all 9V batteries, intentionally loaded in an orientation that is unfavourable to the sorting mechanism.
3. **Success rate under randomized tests:** must successfully sort all batteries and complete without error in 90%+ of randomized tests, with the goal of succeeding in 99%+ of tests.

# 7. Perspective & Survey

## 7.1 History & Existing Solutions

In this section, registered ideas, i.e. reference designs, from patent offices are outlined.

<u>Agitation System</u> [4]

In this patent, an agitation system for measuring the settling rate of solids from a suspension is presented. Of interest to our design is the agitation system, which includes a motor (11 in image) attached to an agitator (14) placed in a solid suspension (21). After the system is turned on, time is allowed for the suspension to be well mixed and for the torque measurement on the agitator to become stable. The agitation system is then stopped.

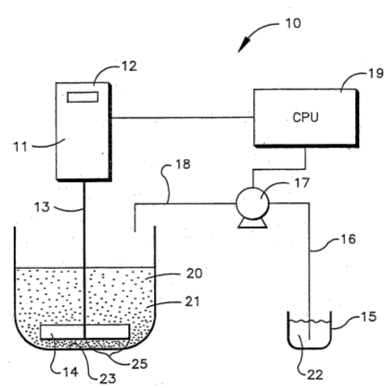



**Figure 7.1: Motor attached to an agitator in solid suspension**

<u>Sorting Batteries Based on Charge Level</u> [5]

In this patent, batteries are supplied on a conveyor and make their way to a testing apparatus, where each battery is passed over a bridged discontinuity on the track. The bridget discontinuity supports the battery, which becomes part of an electrical circuit that tests its voltage level. The potentiometer included in the circuit activates an electromagnet only if the voltage level is above a predetermined value. This activates a solenoid release mechanism that releases the bridge and sets free the battery and separates it from the rest of the batteries awaiting the test.

Sorting Randomly-oriented Objects of Varying Sizes [6]

In this patent, a system that sorts randomly-oriented objects of various sizes is presented. The system is required to know the size of each object before starting the sorting process. Objects are transported along a conveyor and an image system captures the image of each object to be sorted. An image processor then uses the captured image to detect the maximum and minimum linear dimension of each object in order to determine the shape of the object.  The object is then categorized by size based on the the selected maximum or minimum dimension.

# 7.2 Survey

## 7.2.1 Market Survey

In this section, various products on the market that sort different types of batteries are outlined:

Optisort [7]

The Optisort uses computer vision to distinguish batteries. Large batteries are first removed by human inspection before they are fed to a vibrating feeder. This feeder lines the batteries up one by one and sent into a computer vision reader. After distinguishing them, they are separated and literally sent flying by air jets. The Optisort has a >98% purity rate in its sorting of batteries. The limitation of this design is that it is only used for dead-end batteries. It does not detect voltages, which is a key factor in our project.

OBS 600 battery sorter [8]

The OBS 600 battery sorter is an updated version of Optisort. It follows the same principles of the computer vision separation design, but it does it at a much faster rate. The air jet mechanism also makes a return in this design, showing that it is a very effective separation method.

OBS 200 battery sorter [9]

The OBS 200 is a newer and smaller version of the OBS 600. It introduces new ideas with the rotary feeder, which uses centripetal force to send batteries into the final conveyor belt. The initial loading of the batteries is also done by loading the batteries into a bay. Batteries are then pulled up from this bay by using a diagonal conveyor belt with perpendicular platforms to grab the batteries. It is from here that the batteries are dropped down vertically on the rotary feeder.

### 7.2.2 Literature Survey

#### Industrial and Household Recycling of Batteries

A large amount of electrical batteries are discarded regularly by municipalities and industries [5]. Large quantities of such batteries are often used by individual companies and therefore, an annual supply represent a significant economic investment [5]. In many cases, many of the discarded batteries have considerable residual life and yet, no attempts are made to salvage such batteries [5]. As a result, municipalities and industries need a way to test and separate charged batteries from drained ones. However, the sorting process must be efficient and cost-effective so that it would not offset the cost of conserving the batteries [5].

New environmental regulations have been developed in various countries to recycle batteries. Recycling process evaluated by the European Portable Battery Association have proved that metallurgical processes, i.e. those process that involve the chemical and physical behaviour of the elements in a battery, can be used to recycle zinc-carbon batteries. Some companies such as the RMC in Ontario already use such processes to recycle batteries. The products of the recycling process include metallic alloys or metal ions, which are then separated based on density, conductivity, and other properties. An example of a recycling process is the vacuum milling process, where a vacuum system is used to pass the batteries through a cutting chamber to open the casing and release the stored hydrogen. After the batteries are stabilized in a collecting container, the material, for example nickel, can be extracted and reused as a significant alloying component in the production of stainless steel. The information in this paragraph is obtained from [14].

## 7.3 Manuals

We surveyed a number of technical manuals in the design process, a selection of which are presented below:

### 7.3.1 Microcontroller & Circuits:

The most recently updated manuals and reference guides for the De0 - Nano are listed here:

*De0 - Nano User Manual, 2012*
*Nios II Classic Processor Reference Guide, 2016*

Specific assembly language references for the Nios II processor:
*Instruction Set Reference for Nios II, Altera, 2016*
*Altera Nios II Architecture and Programming, 2008*

### 7.3.2 Electromechanical:

*Oriental Motor General Catalog 2003/2004, Technical Reference - Section F*

# 8. Budget

| Item | Reference | Price per unit | Number | Total Price |
|---|---|---|---|---|
| **Electromechanical** | | | | |
| Plywood (⅛” thickness) | https://www.homedepot.ca/en/home/p.std-hardboard-panel-18-inches-x-2-feet-x-2-feet.1000434563.html | $8.70 for 48”x48” board | 1 board | $8.70 |
| Plywood (¼” thickness) | https://www.homedepot.ca/en/home/p.std-hardboard-panel-14-inches-x-2-feet-x-2-feet.1000434567.html | $3.28 for 24”x24” board | 12”x12” board | $0.82 |
| ⅛” Polycarbonate sheets | https://www.mcmaster.com/#polycarbonate-sheets/=174x2w8 | $77.64 for 48”x48” sheet | 24”x18” sheet | $14.56 |
| Brackets | https://www.amazon.ca/25x25x16mm-Degree-Metal-Bracket-Support/dp/B00AUBEZBY | $2.82 for 10 | 3 sets of 10 | $8.46 |
| Screws | https://www.rona.ca/en/flat-head-wood-screws-square-5-1-22-box-1399241 | $2.49 for box of 22 screws | 2 boxes | $4.98 |
| 9g servo | http://www.gearbest.com/power/pp_235842.html?currency=CAD | $3.38 | 6 | $20.28 |

| | | | | |
|---|---|---|---|---|
| Continuous Servo Motor | http://www.robotshop.com/ca/en/9g-continuous-rotation-micro-servo.html?gclid=COfY3LaMmtMCFYGFaQodwFUPYQ | $5.63 | 1 | $5.63 |
| 1/32” Thickness Aluminum Sheets | http://www.onlinemetals.com/merchant.cfm?pid=8025&step=4&showunits=inches&id=748&top_cat=60 | $3.99 for 12”x12” | 1 | $3.99 |
| Gears | http://shop.sdp-si.com/catalog/product/?id=A_1M_2-Y48066 | $1.24 | 6 | $7.44 |
| Racks | https://www.mcmaster.com/#standard-gears/=174xjmd | $5.82 per 12” | 18” | $8.73 |
| Corrugated plastic | http://www.canadiandisplay.ca/18-x-24-Blank-Corrugated-Plastic-Sheets-White-p/BCP04W-18X24.htm?gclid=CIjTyYCHmtMCFQQKaQodrRgEHQ | $2.70 for 18”x24” sheet | 2 18”x24” sheets | $5.40 |
| Plastic bins | https://www.dollartree.com/Cleaning-Storage/storage-organization/Plastic-Storage-Boxes-with-Lids/1470c541c541p330495/index.pro | $1.00 | 4 | $4.00 |
| Velcro patches | https://www.wal | $4.97 for 15 | 8 pairs | $2.65 |

| | | | | |
|---|---|---|---|---|
| | mart.ca/en/ip/velcro-sticky-back-coins-15-sets-white/6000114809910 | pairs | | |
| PVC pipes | http://www.reefsupplies.ca/online-store/PVC-tubing-3-4.html?gclid=CMDq9JyImtMCFQ-GaQodJskHTA | $4.95 for 2.5 feet | Total of 4 feet | $7.92 |
| **Subtotal** | | | | **$103.56** |
| **Electronics** | | | | |
| Jumper wires (braided-core) | https://www.digikey.ca/product-detail/en/bud-industries/BC-32625/377-2093-ND/4156446 | $5.07 | 2 | $10.14 |
| Protoboard | https://www.digikey.com/product-detail/en/seeed-technology-co-ltd/319030010/319030010-ND/5488239 | $3.50 | 1 | $3.50 |
| Resistor set | https://www.digikey.com/product-detail/en/susumu/RR0510RD-1-KIT/RR0510RD-1-KIT-ND/701267 | $8.04 | 1 | $8.04 |
| LM358 op amp | https://www.digikey.com/product-detail/en/texas-instruments/LM358P/296-1395-5-ND/277042 | $0.135 | 9 | $1.215 |
| SN54HC34N | https://www.digik | $0.22 | 1 | $0.22 |

| | | | | |
|---|---|---|---|---|
| Quad Or Gate | ey.ca/product-detail/en/texas-instruments/SN74HC32N/296-1589-5-ND/277235 | | | |
| SN74HC14N Hex Schmitt Inverter | https://www.digikey.ca/product-detail/en/texas-instruments/SN74HC14N/296-1577-5-ND/277223 | $0.323 | 2 | $0.646 |
| 2x4 IC brackets | https://www.digikey.com/product-detail/en/assmann-wsw-components/A-08-LC-TT/AE9986-ND/821740 | $0.076 | 5 | $0.38 |
| 2x7 IC brackets | https://www.digikey.com/product-detail/en/assmann-wsw-components/A-14-LC-TT/AE9989-ND/821743 | $0.092 | 1 | $0.092 |
| 2x8 IC brackets | https://www.digikey.com/product-detail/en/assmann-wsw-components/A-16-LC-TT/AE9992-ND/821746 | $0.104 | 2 | $0.208 |
| PDV-P8103 Photoresistors | https://www.digikey.com/product-detail/en/luna-optoelectronics/PDV-P8103/PDV-P8103-ND/480610 | $0.31 | 3 | $0.93 |

| | | | | |
|---|---|---|---|---|
| 1N4148TR Diodes | https://www.digikey.com/product-detail/en/fairchild-on-semiconductor/1N4148TR/1N4148FSTR-ND/458811 | $0.0086 | 8 | $0.069 |
| Red LED’s | https://www.digikey.com/products/en?mpart=LTL2R3KRD-EM&v=160 | $0.051 | 3 | $0.153 |
| Solder | https://www.digikey.com/product-detail/en/multicore/389261/82-105-ND/2498903 | $11.47 / roll | <5% of a roll | $0.574 |
| 20K Potentiometers | https://www.digikey.ca/product-detail/en/bourns-inc/PDB12-H4151-203BF/PDB12-H4151-203BF-ND/3780656 | $0.718 | 3 | $2.154 |
| ATX Power Supply | https://www.alibaba.com/product-detail/Computer-Power-Supply-ATX-power-supply_170150481.html | $10.00 | 1 | $10.00 |
| **Subtotal** | | | | **$38.32** |
| **Microcontroller** | | | | |
| PIC DevBugger Debugger Board | Project Kit | $50 | 1 | $50 |
| Driver board | Project Kit | $25 | 1 | $25 |
| LCD+Keypad | Project Kit | $6 | 1 | $6 |

| | | | | |
|---|---|---|---|---|
| Real-time clock chip and coin battery | Project Kit | $5 | 1 | $5 |
| **Subtotal** | | | | **$86.00** |
| **TOTAL** | | | | **$227.88** |

# 9. Division of the Problem

We decomposed the problem into general objectives and problem-specific objectives, then determined how these are related to each subsystem of the design.

General objectives:

- Cost
- Volume
- Weight
- Safety
- Accuracy
- Reliability
- Modularity
- Speed
- User Interface

Problem-specific objectives:

- Battery detection
- Voltage sensing
- Battery loading
- Sorting

| **Objective** | **Microcontroller** | **Circuits & Sensing** | **Electromechanical** |
|---|---|---|---|
| Cost | Relatively fixed cost | Relatively easy to keep cheap. Power supply is a significant factor. | Main determinant of overall cost. Major factors:<br>● Material selection<br>● Actuator selection |
| Volume | Negligible | Negligible | Determined by frame size |

| | | | |
|---|---|---|---|
| Weight | Negligible | Negligible, except for power supply | Determined by frame size and material choice |
| Safety | Needs to have emergency stop functionality | Needs to avoid excessive temperature and risk of electrical shock | Needs to be a strong structure that does not excessively vibrate, mechanical systems must be easy to disassemble in case they pinch someone |
| Accuracy | Needs to have error-free code | Needs to have accurate sensors and noise-resistant circuits | Needs to be fabricated to a high degree of precision, actuators need to have high resolution of motion |
| Reliability | Needs to have error-free code, needs to be easily debuggable | Needs to work the same independent of environmental conditions such as temperature, should be resistant to human error such as accidental shorting of terminals | Needs to be made of strong materials, edges should be rounded to reduce friction, design should be resistant to manufacturing error |
| Modularity | Code should be neatly divided into functions which perform atomic operations | Circuit segments should be electrically isolated to reduce noise. Multiple power rails should be used to make expansion easier and avoid tangling/crossing wires | Should avoid chained actuators, where multiple actuators need to perfectly sync up. Should be able to replace all physical components (particularly actuators) with minimal difficulty |
| Speed | Negligible | Sensors should be able to respond quickly to inputs | High-speed motors should be used. System should not rely heavily on chance (e.g. agitating randomly until a battery falls into place) |
| User Interface | Should be intuitive | N/A | N/A |

| | | | |
|---|---|---|---|
| | and easy to use | | |
| Battery Detection | Circuit/microcontroller interface should be small, to minimize integration challenges | Should be able to detect the presence of batteries quickly and easily | Should make it so that batteries can be characterized by their type without the need for elaborate sensors |
| Voltage Sensing | Circuit/microcontroller interface should be small, to minimize integration challenges | Should be able to detect voltages of batteries correctly regardless of orientation, to reduce the need for a complex mechanical design | Should be easy to bring electrodes into physical contact with batteries |
| Battery Loading/Sorting | N/A | N/A | Should be able to load batteries completely unsorted and mechanically sort them to be ready to sense voltage and process |

# 10. Subsystems

## 10.1. Electromechanical Subsystem

### 10.1.1 Assessment of the Problem

The electromechanical subsystem is responsible for designing and building the feeding mechanism, i.e. loading the batteries into the machine. It also includes the construction of the containers (which contains the sorted batteries), the overall structure and frame of the entire machine, as well as the actuators and mechanisms required by the machine. In addition, mounts for sensor and microcontroller components as well as the assignment of their locations are necessary for integration with the circuits and microcontroller subsystems.

### 10.1.2 Solution

#### 10.1.2.1 Rate Limiter

The first step of the recycling process involved collecting the batteries and orienting them in a vertical orientation. The batteries are dumped into a circular container made of corrugated plastic. The container has holes for the batteries to rest in and a dome that directs the batteries into the holes. The purpose of the rate limiter was to ensure only a few batteries at a time are passed to the funnel.

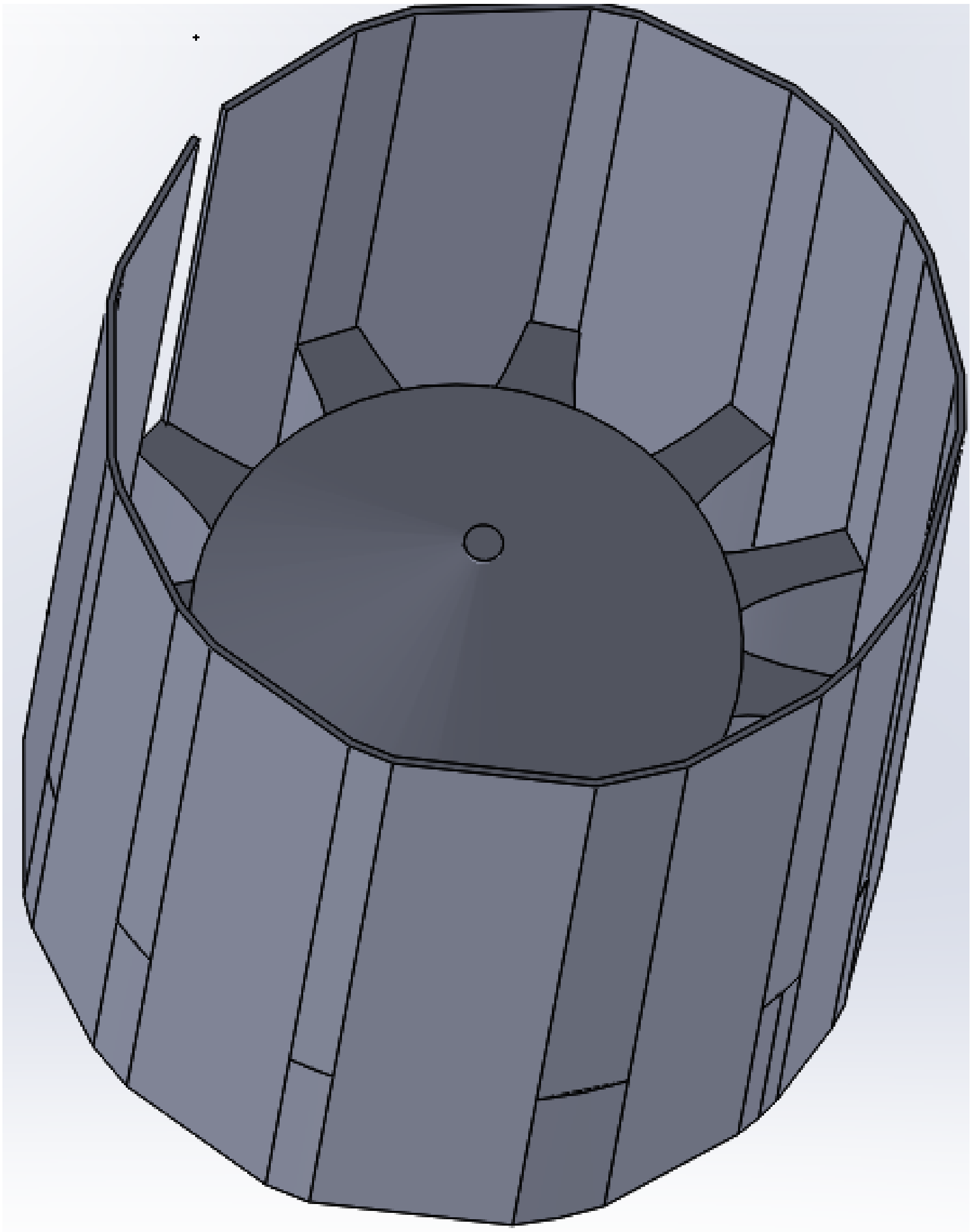


**Figure 10.1.2.1.1** Isometric view of CAD (Computer-aided Design) of rate limiter

This entire mechanism has a diameter of 20 cm, which can fit up to fifteen C batteries according to the dimensions in Table 10.1.2.1 in any orientation. Since the C battery is the largest, the rate limiter can fit fifteen batteries of any type at a time. The height of the rate limiter is 7 cm (more than the longest dimension of the batteries) to ensure that all of them are transferred to the funnel, i.e. none of them would fall over the wall.

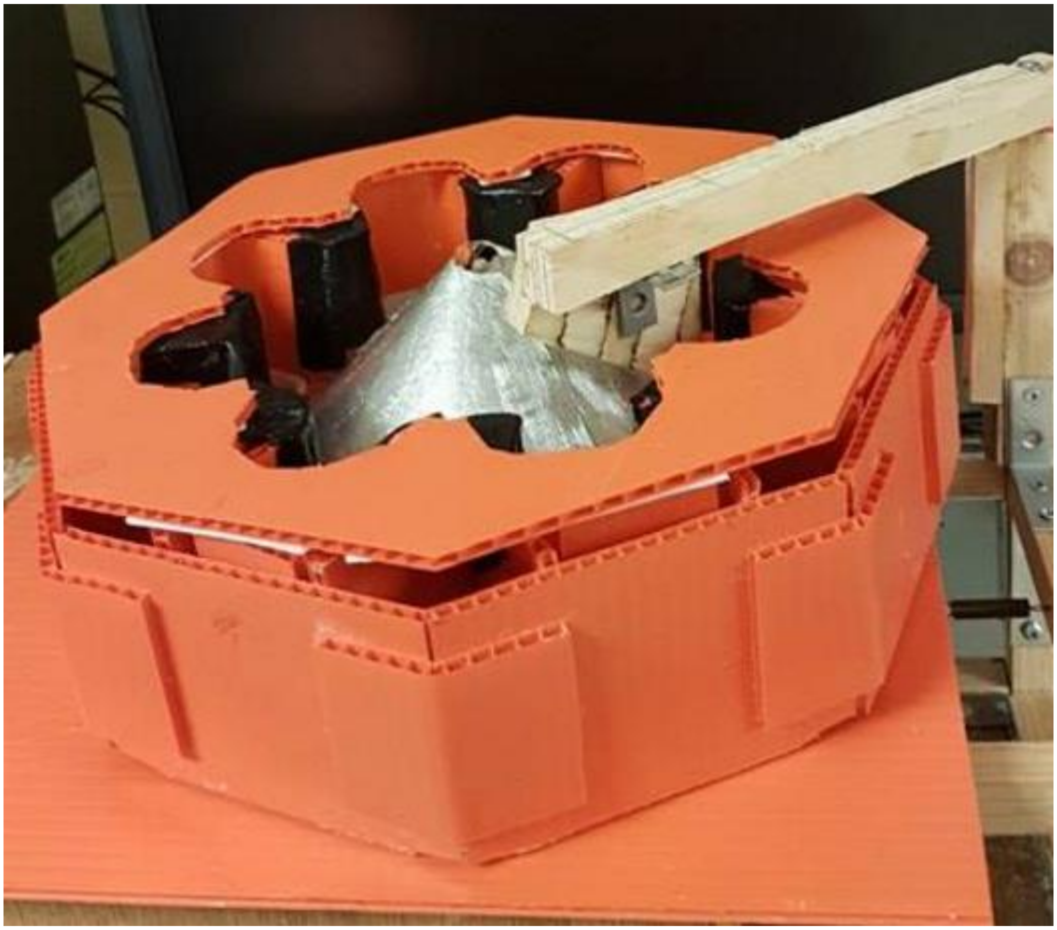


**Figure 10.1.2.1.2** Rate limiter made of corrugated plastic

**Table 10.1.2.1.** Properties of the 3 different battery types [1, 2, 3]

| **Battery Type** | **Height** (mm) | **Diameter** (mm) | **Depth** (mm) | **Width** (mm) | **Weight** (g) |
|---|---|---|---|---|---|
| **9-volt** | 48.5 | n/a | 17.5 | 26.5 | 37 (ZnC), 45 (Alkaline) |
| **C** | 50 | 26.2 | n/a | n/a | 48 (ZnC), 65 (Alkaline) |
| **AA** | 49.2 - 50.5 | 13.5 - 14.5 | n/a | n/a | 19 (ZnC), 24 (Alkaline) |

The rate limiter is controlled by a continuous servo motor (Fitec FS90R) [15] to allow for a full 360-degree motion, a good control over the speed as well as precise motion. It also saved costs for the circuits as it did not require an H-bridge. This motor provided 0.013 Nm of torque which was more than enough to overcome the frictional force (0.42 N) between the corrugated plastic surface and 15 batteries with a gear of 1cm diameter (a sample calculation is provided below). The factor of safety was 3. The rate limiter rotated until a battery falls from one of the holes into the funnel.

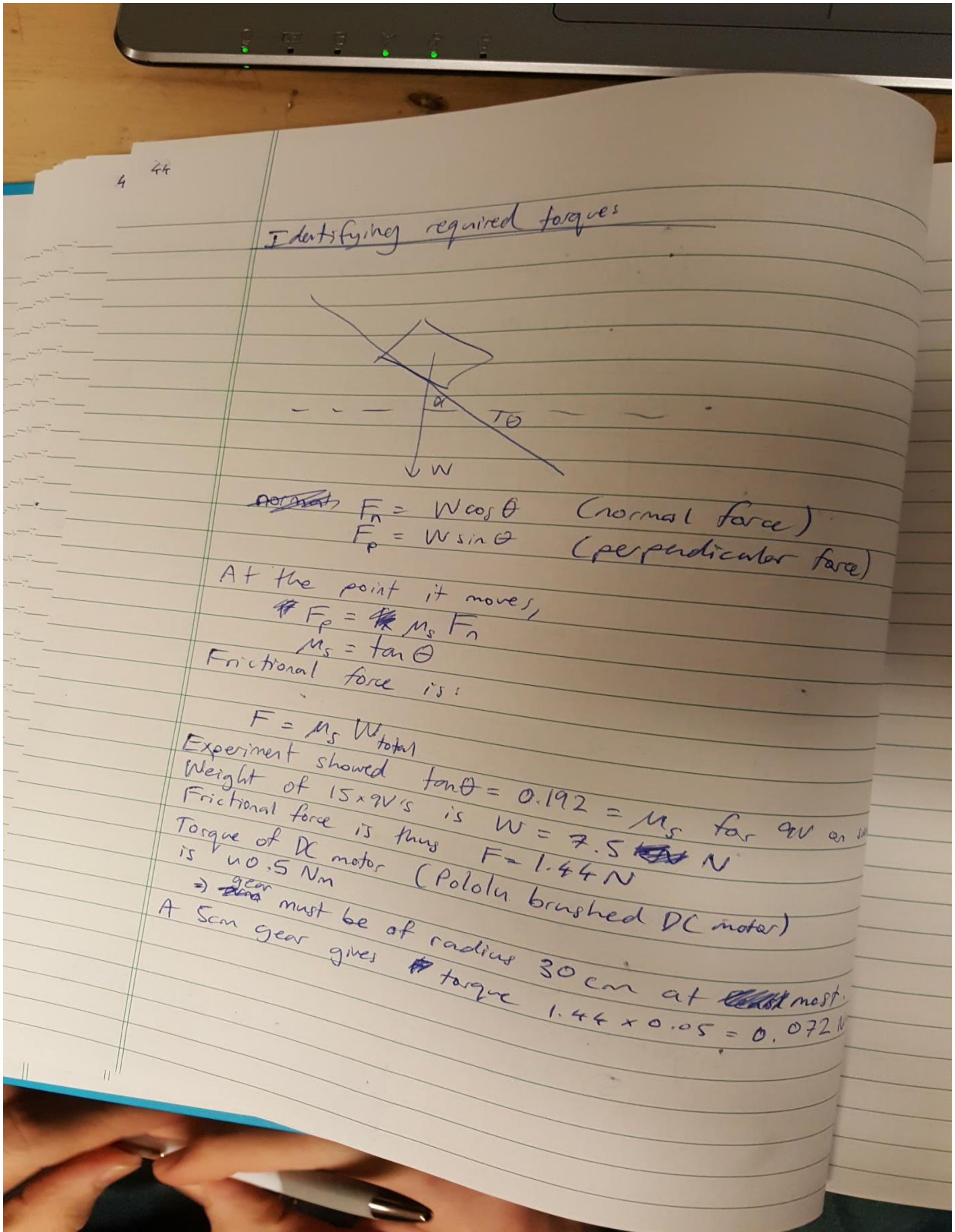



**Figure 10.1.2.1.1** Supporting torque calculation for the selection of the motor

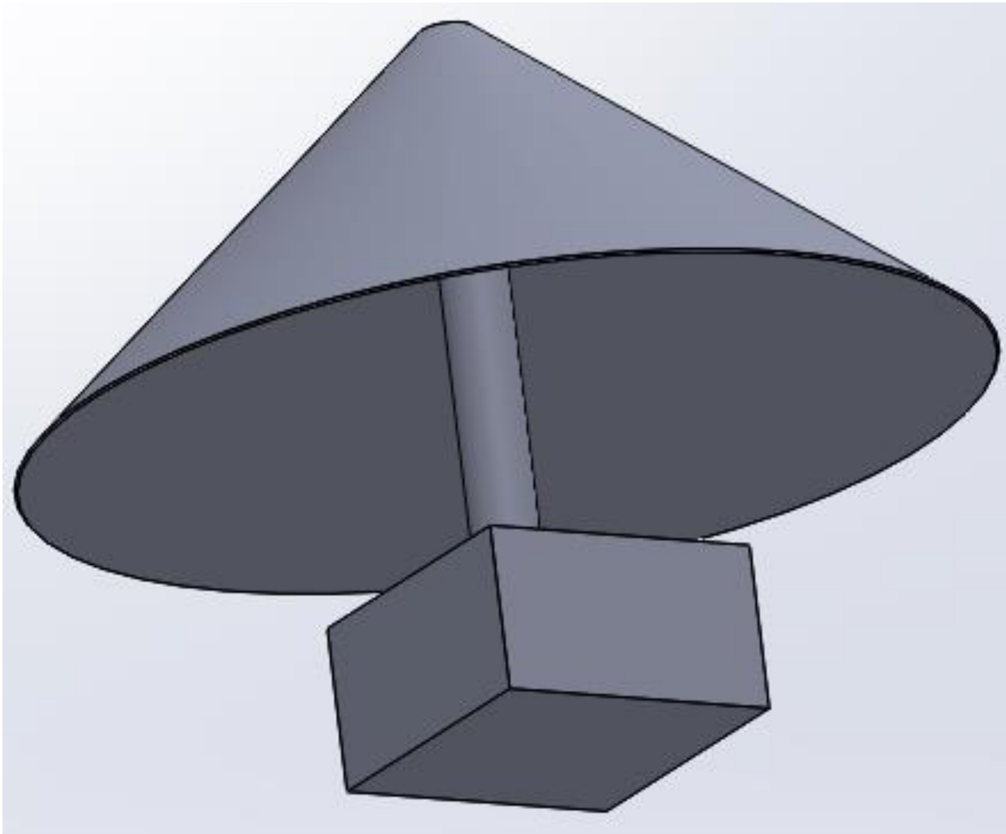


**Figure 10.1.2.1.3** CAD drawing of motor mounted to the dome of the rate limiter with an axle

Moreover, in order to prevent batteries from jamming, a scoop-like mechanism (scopper) was built using wood, screws, and epoxy. This mechanism arched over the rate limiter and its purpose was to push any jammed batteries into the holes.

**Figure 10.1.2.1.4** The scooper mechanism

**Figure 10.1.2.1.5** Top view of rate limiter showing the dome and the holes for the batteries

### 10.1.2.2 Funnel

Once a battery are in one of the holes in the rate limiter, the motor rotates the rate limiter until a battery drops into the funnel. Once in the funnel, the battery is agitated in order to be oriented vertically. The funnel was made of thin metal sheet because it is malleable and can be easily folded into a funnel shape. Being a metal, it is also strong and durable.

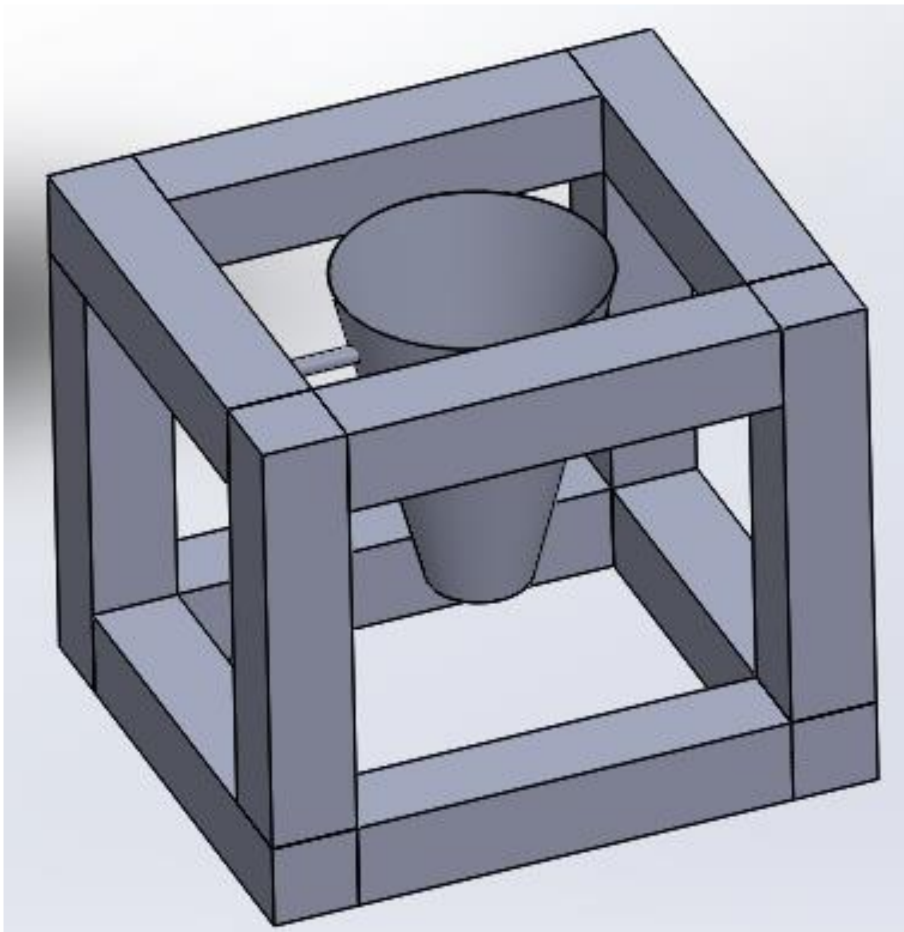


**Figure 10.1.2.2.1** Isometric view of CAD of funnel system

The funnel starts with a 10cm diameter and gradually decreases to a 3cm diameter to ensure the vertical orientation of each battery. The height of it was 7cm to fit around 5 batteries at a time according to the dimensions in Table 10.1.2.1.

**Figure 10.1.2.2.2** Isometric view of funnel and springs. It is positioned right beneath the rate limiter (orange corrugated plastic)

The funnel is supported by springs attached at the top. As a battery falls through, it would push the funnel back and forth. Along with the flexibility offered by the springs, the funnel was agitated and allowed the batteries to pass through it in a vertical orientation. Four springs with a stiffness of 1.58 lb/in (23 N/m) to ensure that the weight of the batteries does not stretch the springs beyond their elastic limit (experiments and calculations support this as show below).

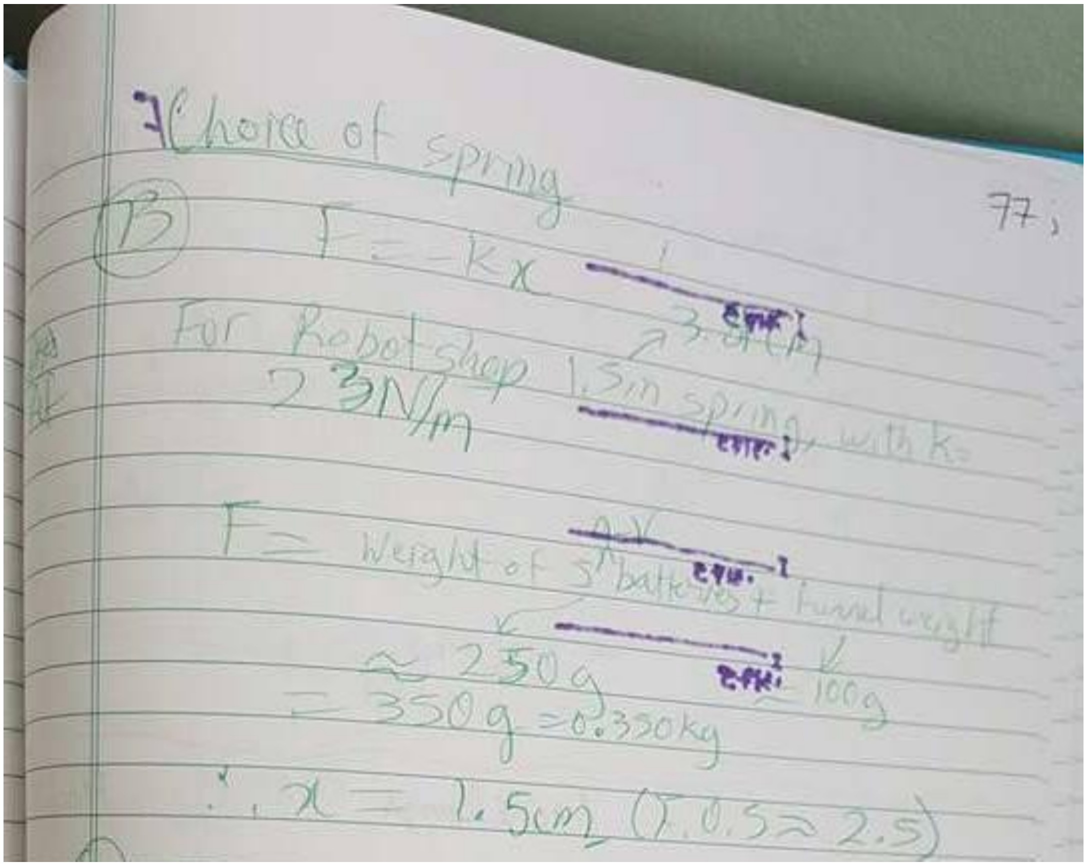



**Figure 10.1.2.2.1** Supporting calculation for choice of spring

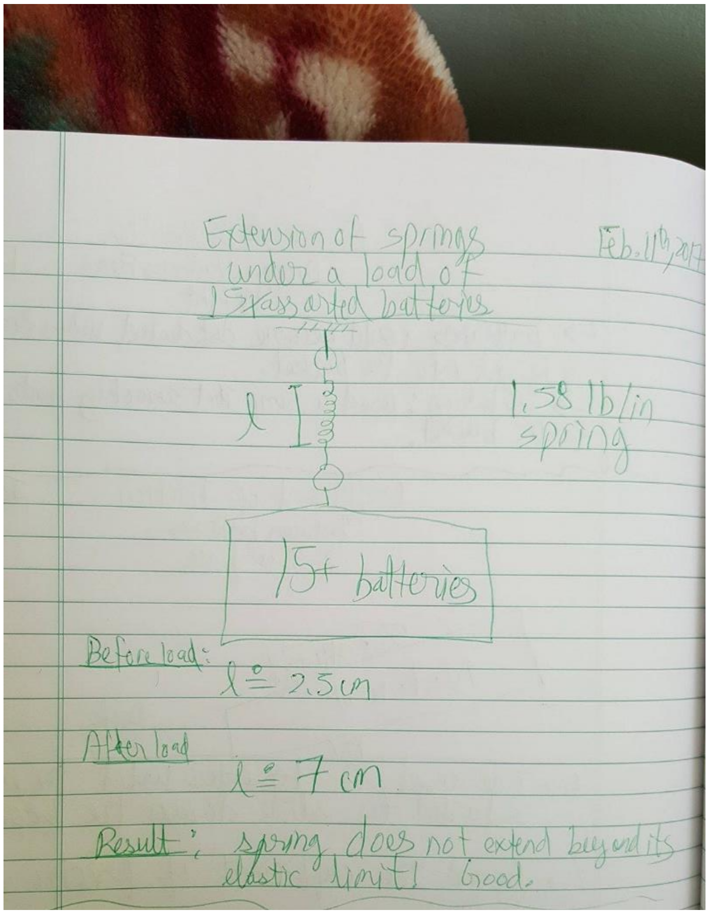



**Figure 10.1.2.2.2** Experiment to select the right stiffness for spring

This entire funnel and spring mechanism is supported on a wooden rectangular prism frame (attached using brackets and screws) with a height of 14cm and a cross-section dimension of 18cm by 18cm (these dimensions are enough to encompass the entire mechanism). The rate limiter along with the funnel constitute the feeding mechanism.

**Figure 10.1.2.2.6** Front view of wooden rectangular prism that supports the funnel

### 10.1.2.3 Battery Blocks

Depending on the battery type, it will fall from the funnel into one of three hollow polycarbonate blocks. The top block has a slot for 9-volt battery, followed by the block that has a slot for the C battery, and finally the block that has the slot for the AA battery (based on the dimensions in Table 1). Both the C and AA batteries pass through the 9-volt battery block, while the AA battery passes through C battery block and into its slot.

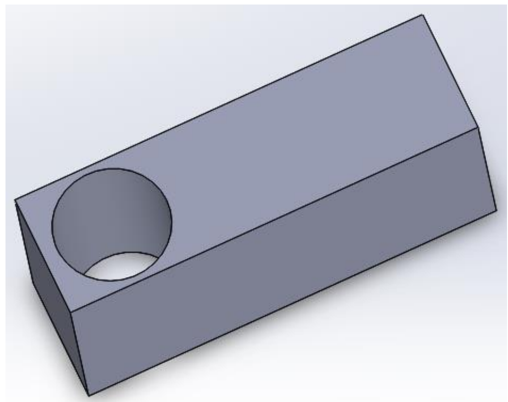


**Figure 10.1.2.3.1** Battery block CAD for 9V battery

The length of the blocks is 14 cm to allow room for pushing the block after a battery has fallen into its slot. The height is 5cm (battery height) and the width is 3cm (battery diameter) which is enough to fit a slot for each type of battery, according to the dimensions in Table 1.

**Figure 10.1.2.3.2** Side view of the 9V block

Attached to each block is the Fitec micro servo [17] which is used with a rack and pinion to push the blocks (once has a battery has fallen into its slot) towards the electrodes, where the voltage of the battery is measured. The servo motor is used because of its accuracy and the rack and pinion because it is cheap, robust, and the easiest method to convert rotational motion into linear motion. The factor of safety is around 3 (determined similarly to the sample calculation in 10.1.2.1).

**Figure 10.1.2.3.3** Rack and pinion mechanism for the C battery block

#### 10.1.2.4 Battery Block Stack

The blocks are stacked on top of each other in a polycarbonate frame, which holds them in place while waiting for a battery to fall through their respective slots and while the voltage is being measured. The dimensions are approximately 27 cm by 5 cm by 8 cm to fit the blocks and provide room for the circuitry of the electrodes.

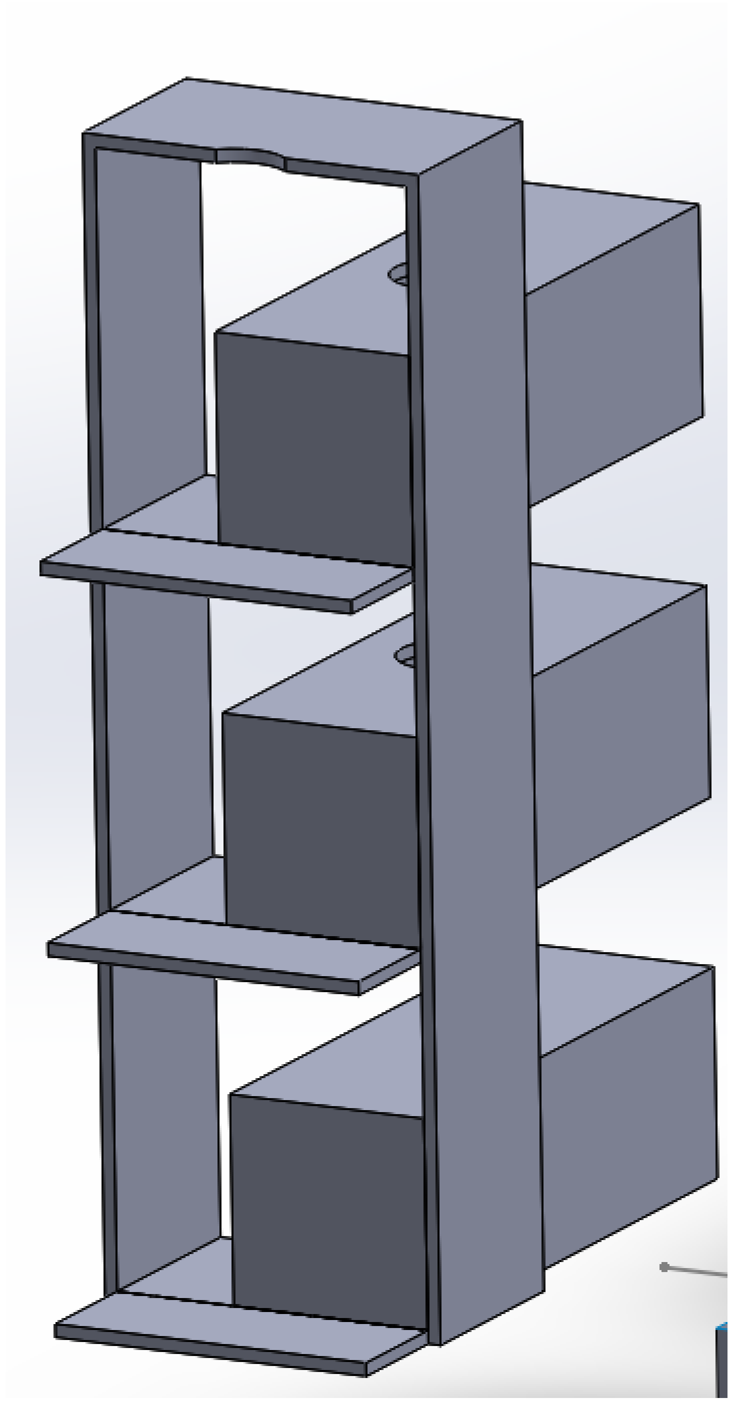


**Figure 10.1.2.4.1.** Stack CAD that supports battery blocks

Plastic was used because it is cheap, strong and also has a relatively low friction. The team chose polycarbonate plastic over acrylic for several reasons. Polycarbonate is stronger and harder to crack under stress than acrylic because it has 250 times the impact resistance of glass while acrylic has 17 times the impact resistance of glass. Polycarbonate is also unlikely to melt when cut using the machines in the shop as it can handle much higher temperature than acrylic. Polycarbonate also does not crack when drilled unlike acrylic which is likely to crack.

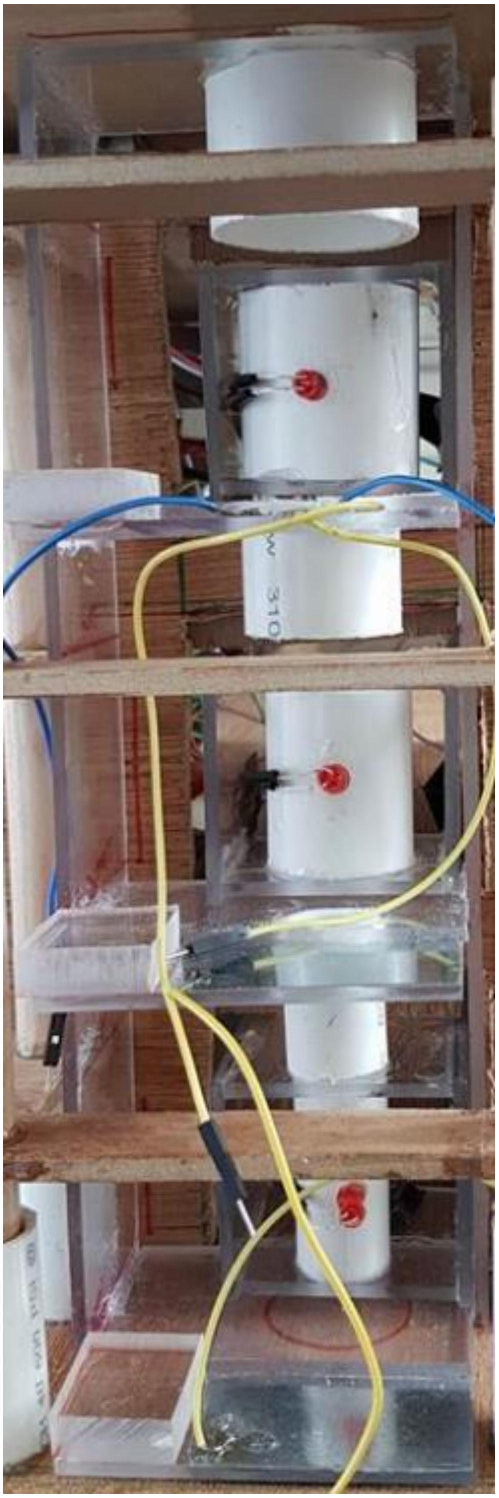


**Figure 10.1.2.4.2.** Front view of stack.

Although polycarbonate costs 35% more than acrylic, the above advantages make up for this disadvantage. This information is obtained from [16]. Epoxy was used because it glues plastic well and creates a strong bond.

### 10.1.2.5 Electrode Frame

Once the batteries are in the slot, the rack and pinion mechanism moves it away from the funnel towards the electrodes. Due to the length of the block, it will block any batteries from falling from the funnel. There are three electrodes, one for each type of battery, controlled by two micro servo motors and a rack and pinion mechanism. A servo motor is used for the aforementioned reasons and a rack and pinion is used to lower the electrodes onto the batteries and measure the voltage.

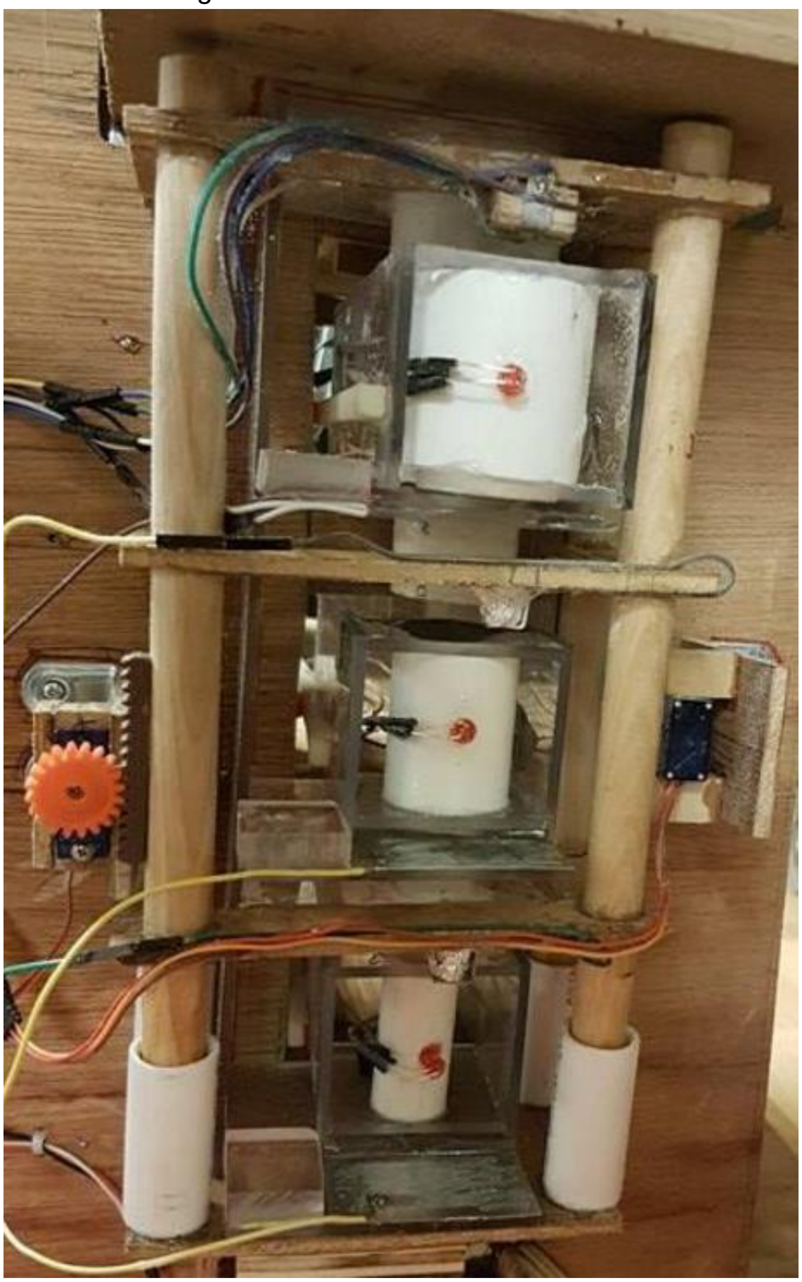


**Figure 10.1.2.5.1** Front view of electrode frame around the stack and battery blocks. The rack and pinion mechanism as well as the micro servo can be observed on the left hand side.

### 10.1.2.6 Ramp

Once it is determined whether or not a battery is charged, it is guided along a ramp (made of wood) into one of four buckets: drained batteries, charged 9-volt, charged C, charged AA. The ramp is controlled by the micro servo motor [17] and based on the battery type, it would align itself with one of the four buckets.

**Figure 10.1.2.6.1** Side view of the ramp positioned directly underneath the AA battery block.

The servo motor is used for its high precision. A layer of foam was attached to the surface of the ramp using epoxy to cushion the batteries as they fall from a height. The ramp is angled at 45 degrees to overcome the friction between the batteries and the foam. This takes into account a factor of safety of 1.33.

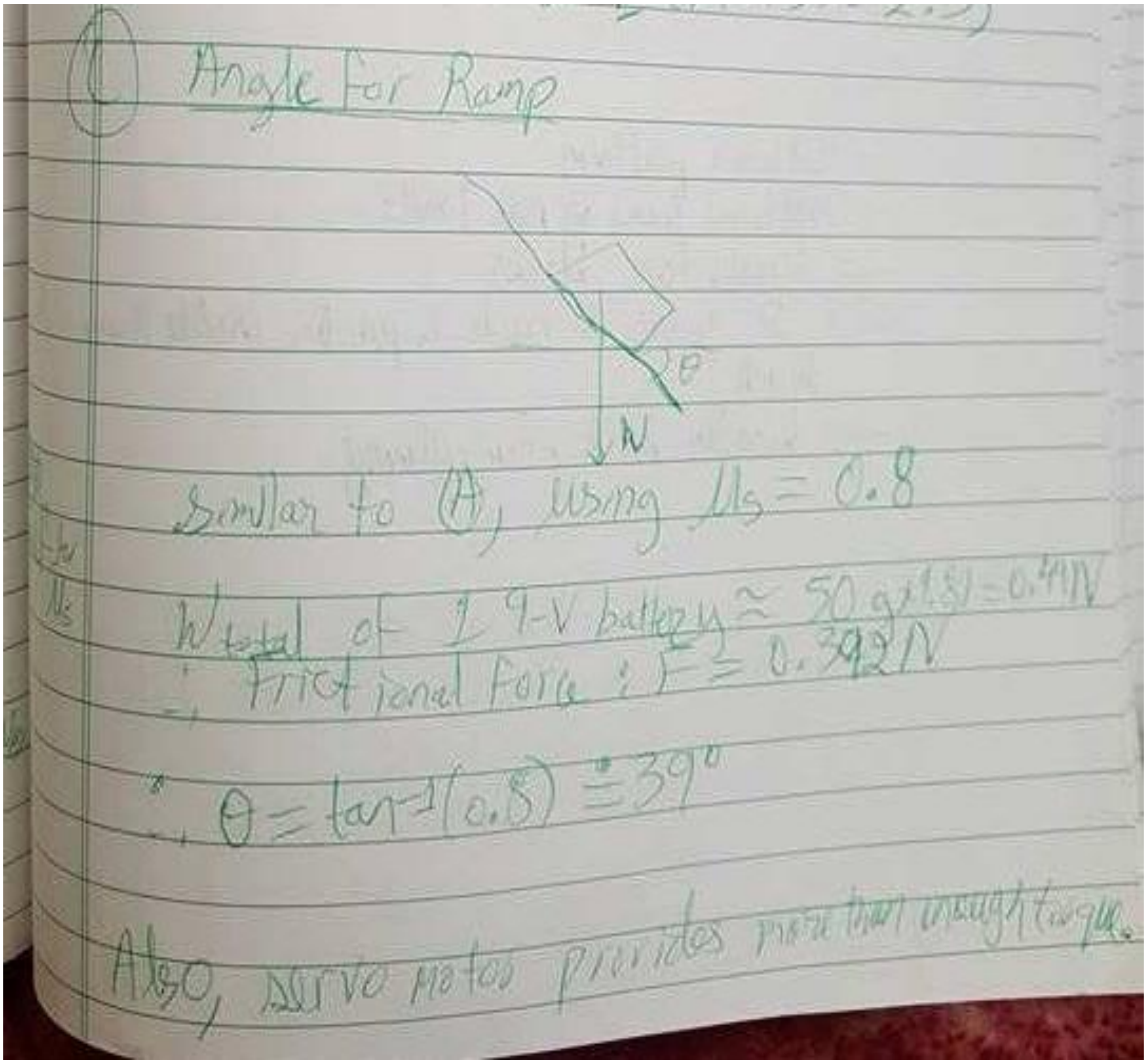



**Figure 10.1.2.6** Supporting calculation for the angle of ramp.

### 10.1.2.7 Containers

The containers were bought off-the-shelf. They had dimensions 6cm by 10cm by 15cm and could fit up to 20 batteries stacked on top of each other. Walls with a height of 4in were glued using epoxy to the containers to prevent a battery from bouncing outside of the machine.

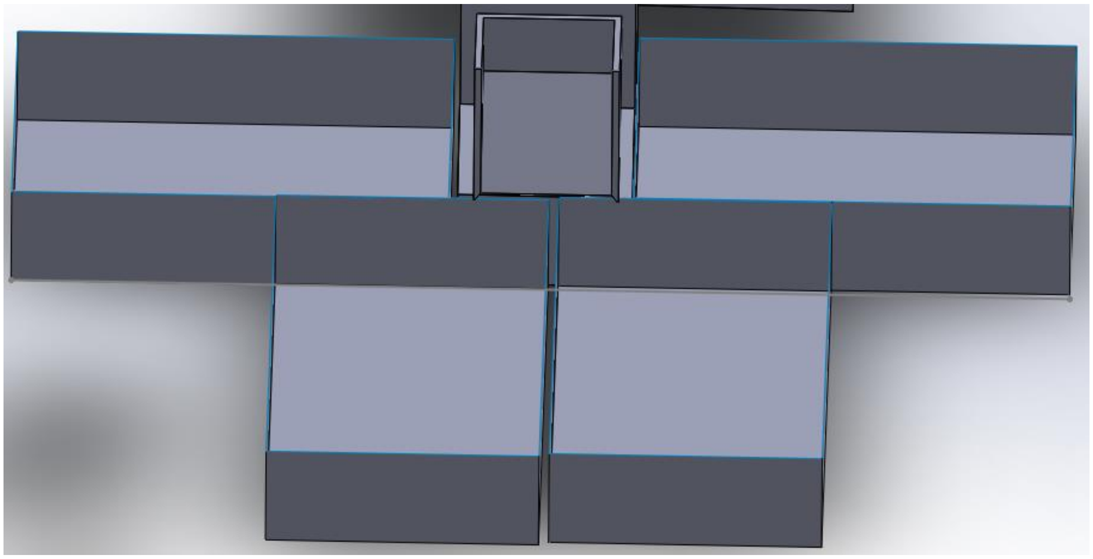


**Figure 10.1.2.7.1** Front view of CAD of the ramp and the four containers.

**Figure 10.1.2.7.2** Isometric view of the four containers and the ramp as designed and built by the team.

### 10.1.2.8 Frames and Structure

The rate limiter as well as the circuits were supported on a 55cm by 25cm by 30cm platform made of wood. This structure also supported the mounts for the servo motors of the three blocks. A smaller platform with dimensions of 12cm by 8cm by 5cm supported the battery blocks stack. The entire machine sat on a 69cm by 69cm base which would help the user transport the machine.

**Figure 10.1.2.8.1** Platform that supports rate limiter, circuitry, and battery blocks

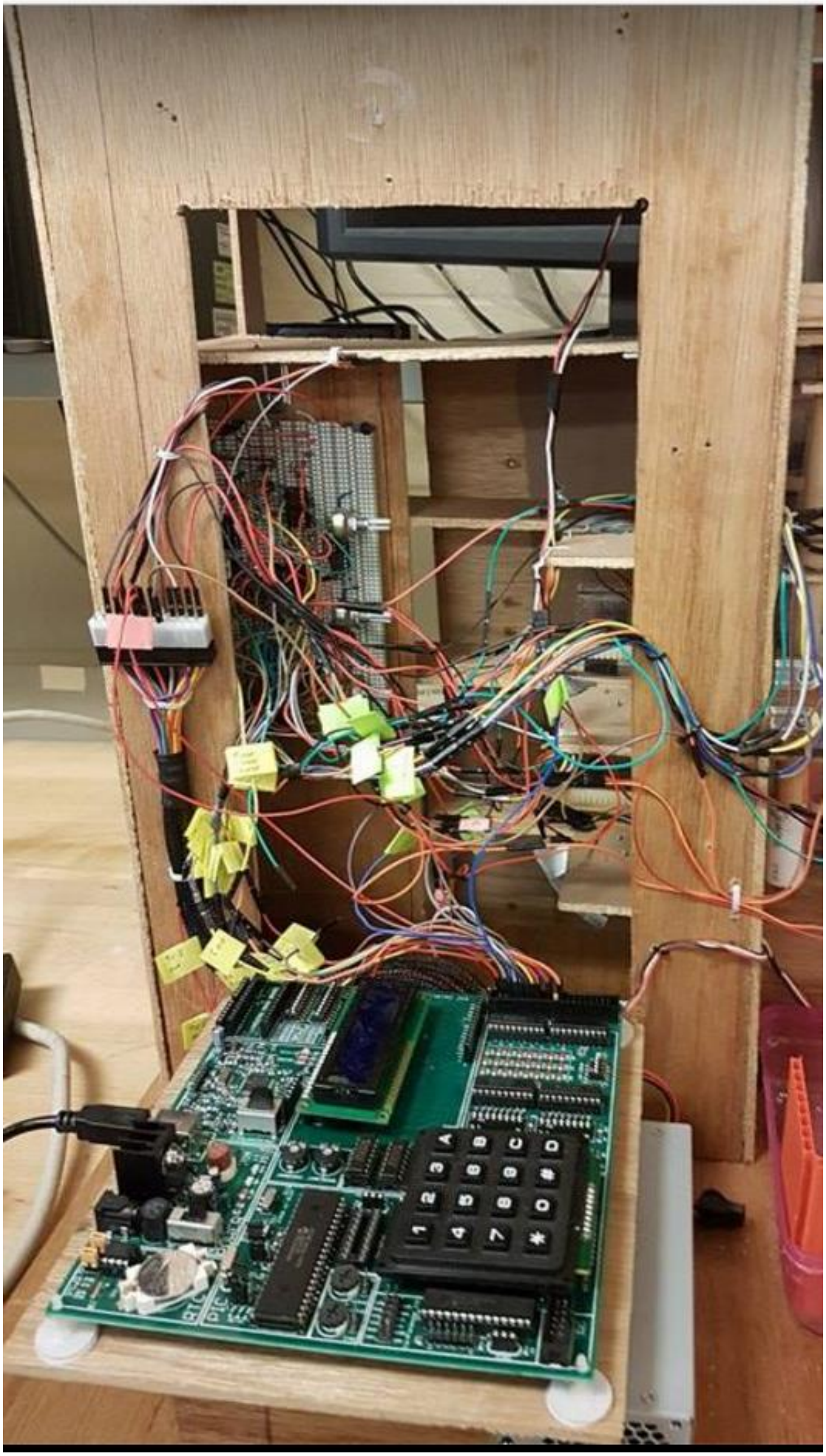


**Figure 10.1.2.8.2** Platform after all the circuitry and microcontroller and power supply are attached

## 10.1.3 Suggestions for Improvement of the Subsystem

### 10.1.3.1 Ramp for all Blocks

An important improvement for the subsystem would be to incorporate a ramp into each of the three battery blocks. The team was able to implement this ramp mechanism for the 9V block but there was little time left to incorporate it into the C and AA blocks.

**Figure 10.1.3.1** Front view of the ramp mechanism (made out of wood) in the 9V block.

This ramp mechanism prevents any jams in the occasional case that 2 batteries fall at a time into the block system. As the block moves forward, the ramp allows the extra battery to slide over it and remain there until the block returns to its initial position. Without the ramp, if two batteries fell at a time then one would fill the hole in the block and the other would partially fit into the hole. The battery that fits partially would prevent the block from moving. With the ramp, however, it would slide over and remain there until the battery occupying the hole is disposed.

#### 10.1.3.2 More Holes in Rate Limiter

Another improvement to prevent jams would be to have more holes in the rate limiter. In the current design, there are 8 holes which proved to be insufficient as the batteries were crammed outside the holes and caused nasty jams that prevented the machine from completely sorting every battery. With more holes, however, it would be more likely for a battery to fall into one of them and thus, reducing the likelihood for jams.

#### 10.1.3.3 Bigger Rate Limiter

Similar to the above section, a bigger room for the batteries means that they are more spread out and jams would be less likely to occur.

#### 10.1.3.4 Better Scooping Mechanism for Rate Limiter

As mentioned before, the purpose of the scooping mechanism is to exert and a horizontal force and push any jammed batteries into one of the holes in the rate limiter. It was noted, however, through test runs that the design is only useful when < 3 batteries are involved in the jam. When many batteries are involved in a jam, it was observed that the horizontal force pushes the batteries against the walls of the rate limiter and thus, accomplishes nothing useful. An improved design should allow the scooping mechanism to exert a vertical force such that it would push up any jammed batteries, thereby preventing them from getting stuck against the walls.

## 10.2 Circuits Subsystem

### 10.2.1 Assessment of the Problem

The primary function of the circuits of the machine is to sense the presence of batteries in the blocks, and to read the voltage of each battery once the block has been moved to the voltage measuring state. This information must be made available to the microcontroller as a set of logic signals, and should be as resistant to noise and electrical cross-talk as possible. It is also necessary for the circuits subsystem to provide electrical power not just for itself but also for the full range of actuators in the electromechanical system.

It is also highly desirable for these circuits to be robust to human or mechanical error, for example shorting two terminals which are not expected to be connected. Wherever possible, this design factor has been taken into consideration.

## 10.2.2 Voltage Sensors

The selected solution for the voltage sensors was to sense battery voltages using a pair of op-amp comparators that compare the voltage across the terminal pairs to positive and negative reference voltages, then pass both of these inputs through an OR gate to be sent to the microcontroller.

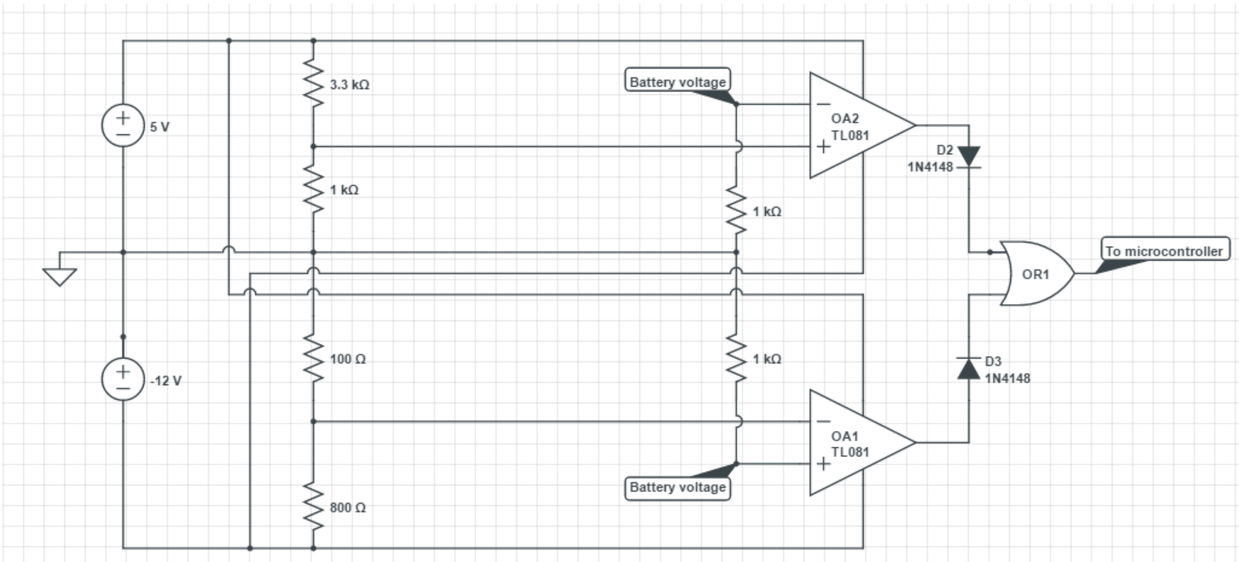



**Figure 10.2.2: Voltage sensor circuit for AA/C batteries. The same battery voltage is connected at the noninverting input of one op amp and the inverting input of the other. The op amp comparators have reference voltages of +1.275V and -1.275V respectively, so at least one will saturate to the +5V rail if the absolute value of the battery voltage is above 1.275V. Diodes are used at the output of the op amps to prevent negative voltages from passing to the OR gate, which uses a +5V positive rail and 0V GND rail and thus cannot handle negative voltages.**

### 10.2.2.1 Calculations

The desired threshold voltages were 0.85 * 1.5V = 1.275V for AA and C batteries and 0.85 * 9V = 7.65V for 9V batteries. Given that the largest positive voltage in the circuits was +5V, the 9V battery voltages were passed through a 1kΩ + 1kΩ voltage divider to halve their voltage, and were compared to a reference voltage of 0.5 * 7.65 = 3.825V. To produce the reference voltages for these given voltage rails of +5V, 0V, and -12V, and using only standard resistor values, the voltage dividers were made with the resistances listed in the table below.

**Table 10.2.2.1: Resistances used to produce reference voltage for voltage sensors.**

| Desired voltage (V) | $R_1$ (kΩ) | $R_2$ (kΩ) | Achieved voltage (V) |
|---|---|---|---|
| 1.275 (between 5V and 0V rails) | 1 | 3.3 | 1.163 |
| -1.275 (between 0V | 0.1 | 0.7 + 0.1 | 1.333 |

| | | | |
|---|---|---|---|
| and -12V rails) | | | |
| 3.825 (between 5V and 0V rails) | 3.3 | 1 | 3.837 |
| -3.825 (between 0V and -12V rails) | 10 | 21 | 3.871 |

### 10.2.3 Light Sensors

Light sensors were selected as the method to detect if a battery was present in one of the slots in the blocks. The method was to have an LED embedded in one side of the PVC pipe and a photoresistor embedded in the other side, with the photoresistor forming part of a voltage divider, the output of which was connected to a comparator. When the PVC pipe is empty, the light passes uninterrupted from the LED to the photoresistor, so the photoresistor has a low resistance and the input voltage to the comparator is low, causing the output to saturate to the negative rail (logic 0). When the PVC pipe contains a battery, the photoresistor is obstructed from almost all light, so the resistance increases dramatically and the input voltage to the comparator is high, causing the output to saturate to the positive rail (logic 1).The reference voltage of the comparator was taken over a rheostat, so that the light sensitivity of the sensors could be tuned to match the lighting conditions.

Since the photoresistor changes resistance so rapidly when exposed to light, this light sensor design exhibits a sharp cutoff and so is very consistent in detecting batteries.

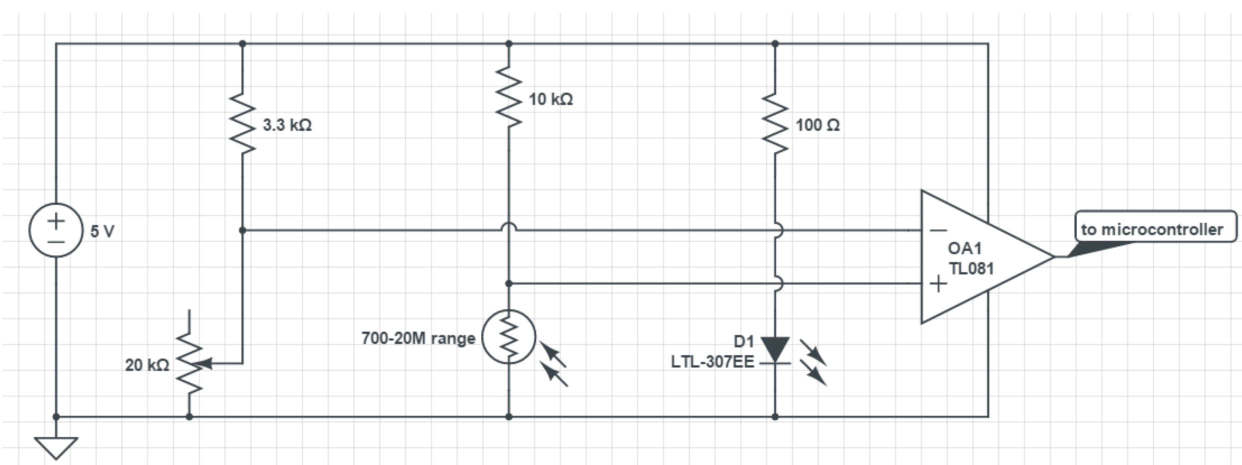



**Figure 10.2.3: Light sensor circuit. The photoresistor is embedded in the PVC tube in the block, directly opposite from the LED. When the light is obstructed, the photoresistor's resistance increases so that the output goes from 0V to 5V, indicating to the microcontroller that a battery is in the slot.**

## 10.2.4 Power Supply

In order to effectively supply the range of voltages required by the design, as well as providing enough power to supply all 7 of the servos in the design under a worst-case load of 1.5A each, the group selected a 500W ATX computer power supply. This proved to be both a highly reliable and convenient solution, as the power available and number of pins of each voltage far exceeded our requirements.

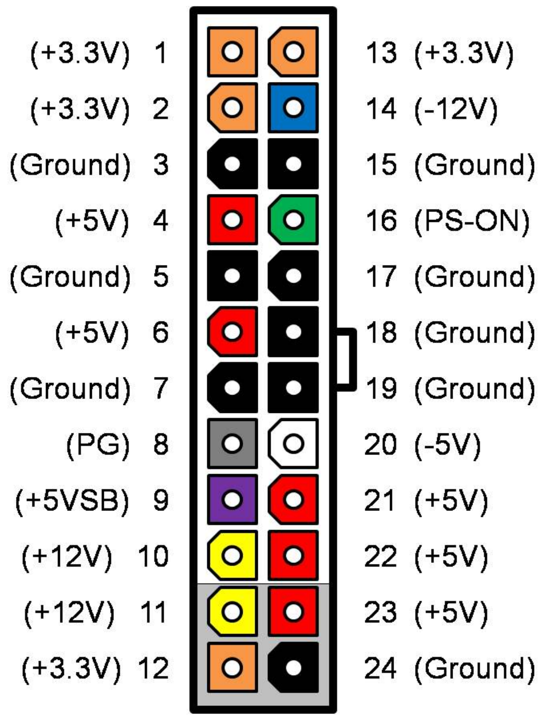



**Figure 10.2.4: ATX pin layout, showing a large number of pins with a range of voltages, making it a very convenient choice.**

### 10.2.4.1 Calculations

The stall current of the SG90 servo motors that were used is 680±80mA [18]. Since 7 servos were used in the design, this gives an upper bound of 5.32A at 5V, for a power drain of 26.6W. The voltage and light sensor circuits described above all use a negligible amount of power, since the outputs are all connected to digital I/O pins with negligible current draw compared to that used by the servos. The 3 LED's were connected in parallel with each other and in series with a 50Ω resistor, with a 5V power supply, and given their constant voltage drop of 1.65V [19] the total current drawn by the LED's was (5-1.65)/50 = 0.067A, for a power consumption of 5V * 0.067A = 0.335W. Thus the maximum total power consumed by the system is 27W, far below the available power of 500W.

## 10.2.5 Suggestions for Improvement of the Subsystem

The subsystem had a very simple and robust design, so no issues were encountered during its operation. However, the design could have conceivably been made more robust by replacing the protoboard design with a PCB so as to reduce the risk of wires falling out, and reduce noise and interference.

# 10.3 Microcontroller Subsystem

## 10.3.1 Introduction

The microcontroller system was designed based on the requirements of the electromechanical and circuits systems, as well as user-interface requirements. The microcontroller that was used was the Peripheral Interface Controller (PIC) Microcontroller. The microcontroller contained multiple components. The components that were used were the power, the PIC18F4620, the PIC284620 I/O, the Debugger switches and lights for debugging purposes, the LCD, and the PICKit3 Programmer to be used with MPLAB X IDE.

The microcontroller must control the servos driving the electromechanical system, and read the sensors from the circuits and sensors system.

The program is running off the internal oscillator which is set at 8 MHz. PLL is enabled so the processor runs at 32 MHz.

## 10.3.2 Program Overview and Flowchart

### 10.3.2.1 Global constants and variables

This program used constant integers as global constants to be used throughout the program. Variable arrays and integers were also used globally. This provides the advantage of reducing bugs related to memory. This also makes the code highly modular. All global variables and constants were defined at the beginning of the program.

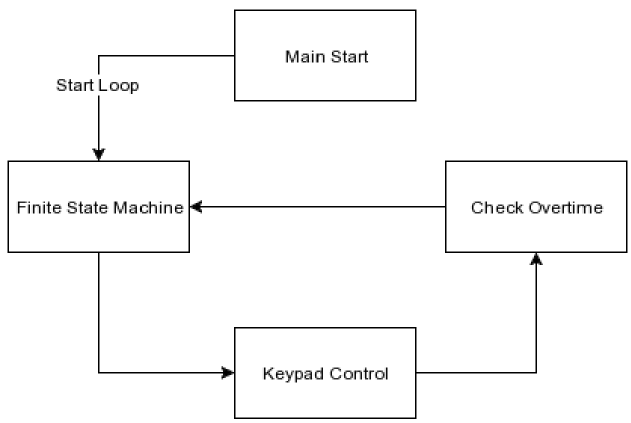
### 10.3.2.2 Pin Assignments

The pin assignments are listed below to prevent confusion when reading program code which will be presented later:

Table 10.3.1 Pin Assignments

| **Pin** | **I/O Assignment** |
| --- | --- |
| RA0 | 9V block photoresistor input bit |
| RA1 | C block photoresistor input bit |
| RE0 | AA block photoresistor input bit |
| RA3, RA4, RA5 | 9V block voltage input bits |
| RC0 | 9V block PWM servo output |

| | |
|---|---|
| RC1 | C block PWM servo output |
| RC2 | AA block PWM servo output |
| RC3 | Electrode tower PWM servo output |
| RC4 | Ramp PWM servo output |
| RC6 | Rate limiter PWM continuous servo output |
| B | Occupied by Keypad |
| D | Occupied by LCD |




### 10.3.2.3 Flow Charts

Figure 10.3.1 Main Loop Architecture

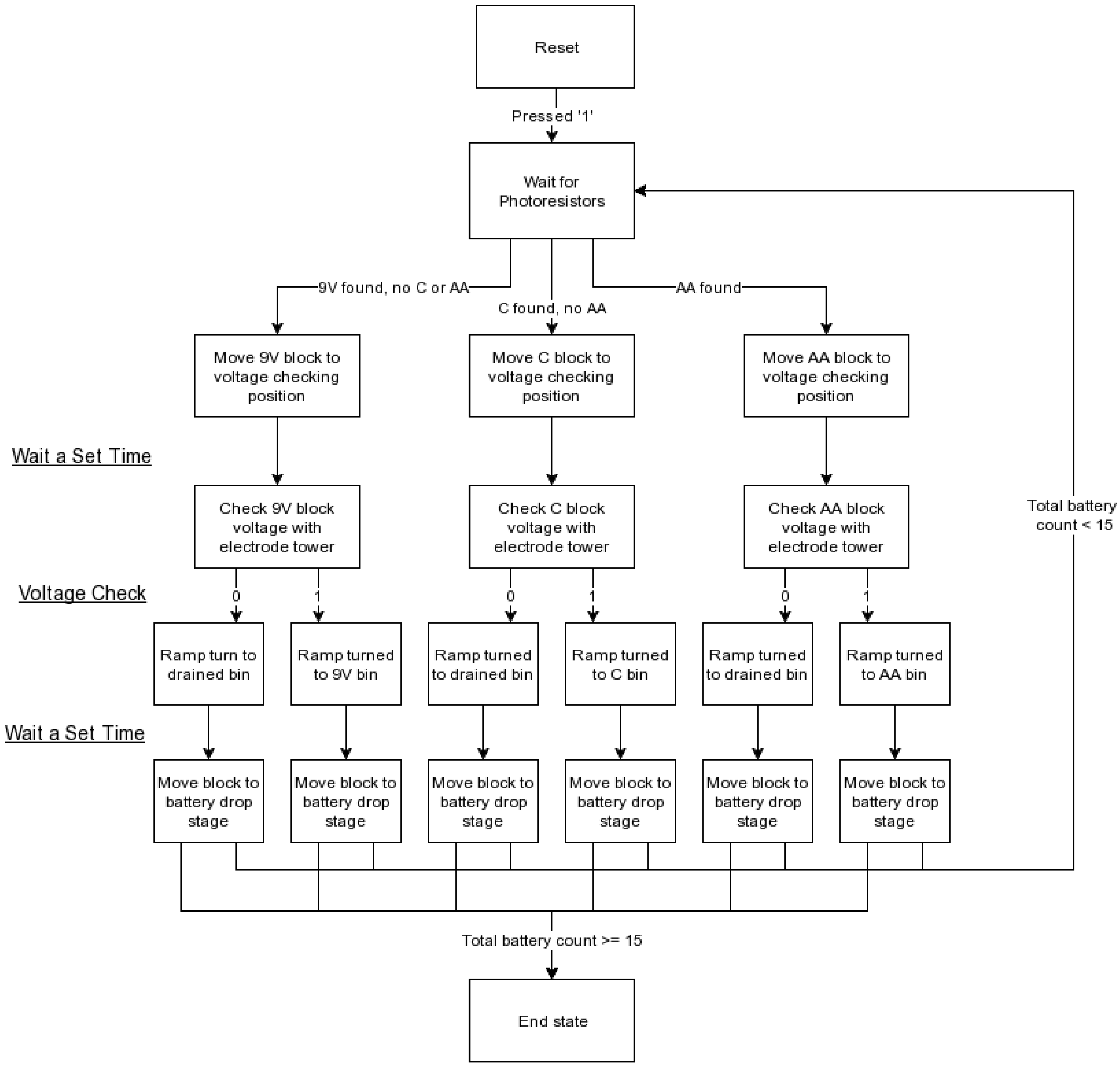



Figure 10.3.2 Finite state machine

## 10.3.3 Servo control

### 10.3.3.1 Problem Assessment

For the electromechanical system requirements, the microcontroller system needs to be able to control 5 different servos for the 3 block system, the electrode tower system, and the ramp-bin system. The microcontroller system also needs to control 1 continuous servo for the rate limiter system. This means that at least 6 output pins must be dedicated to the electromechanical system control.

The normal servos will have their position controlled by PWM. The continuous servo will have its speed controlled by PWM.

#### 10.3.3.2 Pulse Width Modulation

The pulse width modulation (PWM) was controlled with a timer module. The timer is calibrated in the main function for a set amount of time. The time was set the equation:

$$(2^{16} - 1)\text{-S(clock frequency in MHz)}/4 * 1/\text{prescaler}$$

Calculation 10.3 Timer module interrupt time calculation

to get a 16 bit number, where the 8 largest bits are set for TMR1H (TIMER1 High) and the 8 lowest bits are used for TMR1L (TIMER1 Low). A prescaler of 1 is used for TIMER1. TIMER1 is calibrated so that it sends an interrupt signal once every 0.02 ms. After the calibration, the timer is controlled using an interrupt handler. The software code for the TIMER1 in the interrupt handler written in C is shown below:

Code Snippet 10.1 PWM output

```
if(TMR1IF){
        TMR1IF = 0; // Reset the interrupt flag

        TMR1ON = 0; // Turn off TIMER1 while interrupt is handled

        // Initialize timer again
        T1CON = 0b10000000;
        TMR1H = 0b11111111; // TIMER1 High bits
        TMR1L = 0b01011111; // TIMER1 Low bits

        if (timer_on == 1) { // timer_on is the ultimate control variable,
which is set to 0 when the emergency stop is pressed

        // PWM
        // Apply voltage low to servos depending on their duty cycle
        if (state != EMERGENCY) {
            timer_counter++; // the variable counter controlling the duty
cycles of the various servos

            // 9V
            if (timer_counter >= servo_duties[SERVO_9V]) {
                if (LATCbits.LATC0 != 0)
                    LATCbits.LATC0 = 0;
            }

            // C
            if (timer_counter >= servo_duties[SERVO_C]) {
                if (LATCbits.LATC1 != 0)
                    LATCbits.LATC1 = 0;
            }

            // AA
            if (timer_counter >= servo_duties[SERVO_AA]) {
                if (LATCbits.LATC2 != 0)
```

```
                LATCbits.LATC2 = 0;
            }

            // Electrode
            if (timer_counter >= servo_duties[SERVO_ELEC]) {
                if (LATCbits.LATC3 != 0)
                    LATCbits.LATC3 = 0;
            }

            // Ramp
            if (timer_counter >= servo_duties[SERVO_RAMP]) {
                if (LATCbits.LATC4 != 0)
                    LATCbits.LATC4 = 0;
            }

            // Rate Limiter
            if (timer_counter >= servo_duties[SERVO_RL]) {
                if (LATCbits.LATC6 != 0)
                    LATCbits.LATC6 = 0;
            }

            // reset to 0 after 20 ms
            if (timer_counter >= 500) {
                timer_counter = 0;

                // apply voltage high to all servos
                LATCbits.LATC0 = 1;
                LATCbits.LATC1 = 1;
                LATCbits.LATC2 = 1;
                LATCbits.LATC3 = 1;
                LATCbits.LATC4 = 1;

                if (rl_state == 1)
                    LATCbits.LATC6 = 1;

            }
        }

        // USED FOR RATE LIMITER
        if (timer_counter3 < RL_WAIT_STANDARD)
            Timer_counter3++;

        // Other logic code regarding the other voltage and photoresistor
pins...

        TMR1ON = 1;

        }
    }
```

The basic idea of this PWM code is that the variable timer_counter is constantly incrementing, until it reaches a number (500), where it then goes back to 0 and sets all PWM outputs on voltage high. An array called servo_duties holds the duty cycles of each of the servos. The contents of this array are modified by the program. The elements of the array reflect at which number timer_counter is on for the specific servo for that element to be set at voltage low, thus creating the duty cycle and PWM.

#### 10.3.3.3 Rate Limiter continuous servo control

The rate limiter uses a continuous servo, so the PWM sent to it controls speed and direction rather than position.

The variable timer_counter3 is used in logic for the rate limiter, which controls the direction and speed of it. The logic for the rate limiter is as follows: The rate limiter makes a long turn clockwise and then a short turn counter clockwise each cycle, which is timed by timer_counter3. Each cycle is done X amount of times, starting at X = 3. After X cycles, the directions reverse, and the rate limiter does a short turn clockwise and long turn counter clockwise. This repeats every X cycles. X increases based on operation time. The reason for this complicated rate limiter logic is that it is an optimization of the rate limiter to prevent jamming. The code for the rate limiter logic in the wait state can be seen below:

Code Snippet 10.2 Rate Limiter Control

```
// RATE LIMITER LOGIC
// stop the rate limiter ASAP once batteries are detected
// otherwise just time it on and off
if (sensor9V > 5 || sensorC > 5 || sensorAA > 5) {
    servo_duties[SERVO_RL] = RL_OFF;
    rl_change_dir_counter = 0;
}
else if (sensor_state != RESET_SENSOR) {
    if (rl_direction == 1) {
        if (timer_counter3 >= rl_wait_1) {
            timer_counter3 = 0;
            if (rl_state == 1) {
                rl_state = 0;
                servo_duties[SERVO_RL] = RL_OFF;
            }
            else {
                rl_state = 1;
                servo_duties[SERVO_RL] = rl_dir_1;

                // changes the direction of turning
                rl_direction = 2;
            }
        }
    }
```

```
        else if (rl_direction == 2) {
            if (timer_counter3 >= rl_wait_2) {
                timer_counter3 = 0;
                if (rl_state == 1) {
                    rl_state = 0;
                    servo_duties[SERVO_RL] = RL_OFF;
                }
                else {
                    rl_state = 1;
                    servo_duties[SERVO_RL] = rl_dir_2;

                    rl_direction = 1;

                    // increases direction changing counter
                    rl_change_dir_counter += 1;
                }
            }
        }

        // swap the directions
        int max_rl_change = MAX_RL_CHANGE + operation_time/20;
        if (max_rl_change > 9)
            max_rl_change = 9;
        if (rl_change_dir_counter >= max_rl_change) {
            if (rl_dir_1 == RL_CCW)
                rl_dir_1 = RL_CCW_FAST;
            else
                rl_dir_1 = RL_CCW;

            if (rl_dir_2 == RL_CW)
                rl_dir_2 = RL_CW_SLOW;
            else
                rl_dir_2 = RL_CW;

            rl_change_dir_counter = 0;
        }
    }
```

#### 10.3.3.4 Servo calibration

Due to numerous sources of error ranging from manufacturing to noise, the exact duty cycles for the servos were not calculated. They were instead calibrated by trial and error. The numbers in the servo_duties array were physically tested using a small program written for the keypad to change the servo duty cycles during run time. When the blocks were in the correct positions, the servo duty cycle displayed on the keypad screen would be recorded and the global constant that referred to that duty cycle would be changed in the program.

## 10.3.4 Sensor input

### 10.3.4.1 Problem Assessment

For the circuits and sensors system, the microcontroller system needs to read the voltage bits and photo-resistor bits. Since the circuits system is taking control of distinguishing when a battery is charged or uncharged, the microcontroller system needs to only process a 1 or 0 digital input rather than an analog voltage input. The circuits system is sending 3 inputs for the 9V battery voltage detector, and 1 input each for the C battery voltage detector and AA battery voltage detector. This requires a total of 5 input pins so far. Also, the sensors system has one photoresistor for each block of the block system, which means the microcontroller needs to receive 3 more input pins for a total of 8 input pins.

### 10.3.4.2 Solution

The solution to reading the inputs is straightforward. The program checks if those input pins send a high signal in their respective states. In the wait state, the input pins for the photoresistor inputs are checked. In the voltage checking states, the voltage pins for 9V, C, or AA voltage input pins are checked. They are checked to bet set to high for a certain amount of continuous time to eliminate possible errors.

Code Snippet 10.3 Sensor input reading in TIMER1 module:

```
// Checking photo-resistors (Used in WAIT state)
// 9V block photo-resistor
if (PORTAbits.RA0 == 1) {
    if (sensor9V < SENSOR_WAIT + 1) {
        sensor9V++;
    }
} else {
    if (sensor9V > SENSOR_INSTANT + 0) {
        sensor9V--;
    }
}

// C block photo-resistor
if (PORTAbits.RA1 == 1) {
    if (sensorC < SENSOR_WAIT + 1) {
        sensorC++;
    }
} else {
    if (sensorC > SENSOR_INSTANT + 0) {
        sensorC--;
    }
}

// AA block photo-resistor
if (PORTEbits.RE0 == 1) {
    if (sensorAA < SENSOR_WAIT + 1) {
```

```
                sensorAA++;
            }
        } else {
            if (sensorAA > SENSOR_INSTANT + 0) {
                sensorAA--;
            }
        }

        // Checking voltages (Used in V_9V, V_C, V_AA states)
        voltcheck_1 = PORTEbits.RE1;
        voltcheck_2 = PORTAbits.RA4;
        voltcheck_3 = PORTAbits.RA5;

        // 9V block voltage checker
        if ((voltcheck_1 == 1) || (voltcheck_2 == 1) || (voltcheck_3 == 1)) {
            if (volt9V < VOLT_WAIT + 1000)
                volt9V += 10;
        } else {
            if (volt9V > VOLT_INSTANT + 0) { // This is for hardware reasons
relating to the PWM
                volt9V--;
            }
        }

        // C block voltage checker
        if (PORTEbits.RE2 == 1) {
            if (voltC < VOLT_WAIT + 1000)
                voltC += 10;
        } else {
            if (voltC > VOLT_INSTANT + 0) { // This is for hardware reasons
relating to the PWM
                voltC--;
            }
        }

        // AA block voltage checker
        if (PORTCbits.RC7 == 1) {
            if (voltAA < VOLT_WAIT + 1000)
                voltAA += 10;
        } else {
            if (voltAA > VOLT_INSTANT + 0) { // This is for hardware reasons
relating to the PWM
                voltAA--;
            }
        }
```

These incremented values (volt9V, voltC, voltAA, sensor9V, sensorC, sensorAA) are checked in their respective voltage checking or block waiting states.

## 10.3.4 User Interface

### 10.3.4.1 Problem Assessment

The user must be able to press the keypad buttons and be able to see feedback on the LCD screen. The user must be able to start, reset, and emergency stop the process.

### 10.3.4.2 LCD

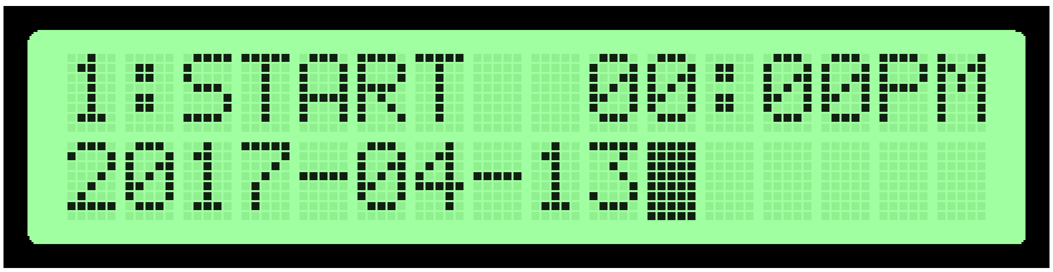
The LCD is controlled by the microcontroller C code. This controlled by a function called putch(), which sends a 4 bit or 8 bit signal to the LCD to display a character. The LCD functions can be seen below:

Code Snippet 10.4 LCD functions:

```
void initLCD(void) {
    __delay_ms(15);
    lcdInst(0b00110011);
    lcdInst(0b00110010);
    lcdInst(0b00101000);
    lcdInst(0b00001111);
    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);
}
```

These functions are then used to print characters on the LCD screens. The order of which they do this is stored in functions specific to different states. The screens that the normal user will




see can be seen below:

Figure 10.3.3 LCD start screen (RESET state)

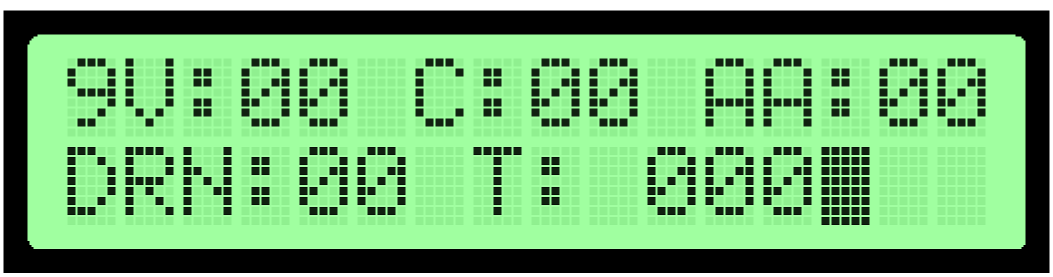



Figure 10.3.4 End screen. This displays the number of batteries per the different categories (9V, C, AA, drained), as well as the total operation time

### 10.3.4.3 Keypad

There are various features that the keypad has for normal use and debugging. The debouncing of the keypad is controlled by an internal circuit. Below is an image of the keypad:

Figure 10.3.5 Keypad

The code for reading the keypad presses every main loop is shown below:
Code Snippet 10.5 Reading keypad:
```
if (PORTBbits.RB1 == 1) {
        // read the 4 bit character code
        keypress = (PORTB & 0xF0)>>4;
        temp = keys[keypress];

        Nop();  //Apply breakpoint here because of compiler optimizations
        Nop();

        switch (temp) {
            case '1':
                  // set up the variables and states for the WAIT state
                  break;
            case ‘2’:
            // ...
            // Rest of buttons and code
            // ...
            default:
                  break;
      }
}
```

When pressing ‘1’, the program goes to the WAIT state, and clears all variables to their default values. When pressing ‘2’, the program goes to the RESET state, and clears all variables to their default values. When pressing ‘3’, the program emergency stops, immediately stopping all PWM, LCD, and keypad functions.

## 10.3.5 Miscellaneous Features and Optimizations

### 10.3.5.1 Servo Calibration

The servo duty cycles can be calibrated manually in a test mode, where keys ‘4’ and ‘A’ decrease and increase the duty cycle by 1 point of precision respectively, and the key ‘5’ toggles through which servo that is currently being calibrated.

### 10.3.5.2 Block servo agitation

The block servos agitate back and forth in the WAIT state based on a timer_counter4 variable that is incremented in the TIMER1 module. This is done to optimize the chance of batteries falling down correctly. The code for it can be seen below:

Code Snippet 10.6 Block agitation in the WAIT state:

```
// AGITATION OF BLOCKS IN WAIT STATE
      if (sensor_state != RESET_SENSOR) {
          if (timer_counter4 >= AGITATE_WAIT_1) {
              if (servo_duties[SERVO_AA] == RESET_POS_AA) {
                  servo_duties[SERVO_9V] = RESET_POS_9V - 3;
                  if (sensor_state == WAIT_SENSOR_C)
                      servo_duties[SERVO_C] = FORWARD_POS_0_C - 3;
                  else
                      servo_duties[SERVO_C] = RESET_POS_C - 4;
                  servo_duties[SERVO_AA] = RESET_POS_AA - 3;
              } else {
                  servo_duties[SERVO_9V] = RESET_POS_9V;
                  if (sensor_state == WAIT_SENSOR_C)
                      servo_duties[SERVO_C] = FORWARD_POS_0_C;
                  else
                      servo_duties[SERVO_C] = RESET_POS_C;
                  servo_duties[SERVO_AA] = RESET_POS_AA;
              }
              timer_counter4 = 0;
          }
      }
```

### 10.3.5.3 Operation and Real Time

The time displayed at the end screen as seen in figure 10.3.4 is calculated by a separate timer module (TIMER0), which is calibrated for interrupts every 1 second. The TIMER0 module is also used for real time counting, which can be seen in figure 10.3.3. The code for TIMER0 can be seen below:

Code Snippet 10.7 Operation and Real Time counting in TIMER0:

```
if (TMR0IF) {
        TMR0IF = 0;
        TMR0ON = 0;

        // Initialize timer again

        T0CON = 0b00000111;
        TMR0H = 0b10000101;
        TMR0L = 0b11101101;

        if (state != RESET && state != EMERGENCY && state != END) {
            operation_time += 1;
        }

        // REAL TIME CLOCK
        rt_s += 1;
        if (rt_s >= 60) {
            rt_m += 1;
            rt_s = 0;
        }
        if (rt_m >= 60) {
            rt_h += 1;
            rt_m = 0;
        }

        TMR0ON = 1;
    }
```

## 10.3.6 Suggestions for Improvement of the System

### 10.3.6.1 Assembly

While C code is convenient to write, and every team member is very familiar with C, it is not the best language for a hardware project like this. Many of the bugs discovered were hardware related rather than logical errors in the code. One example is that using the timer module for PWM output caused that any slight change to the functions in the timer module cause slight changes in the PWM. This resulted in changes in the servo positions. Before, the TIMER1 module had different lines of code executed based on the state of the program, causing changes to the time in the PWM. This was obviously true, but it was unexpected that certain lines of code could cause such a large change in the time. It was discovered that checking pins for values caused a significant change in the time of the TIMER1 module functions, resulting in different PWM outputs. In the future, Assembly should be used for this project, since the majority of bugs discovered will be hardware related and down to the very lines of machine code executed.

# 11. Integration

## 11.1 Stage 1: Preliminary Microcontroller & Circuit Integration (28/02/2017)

The first integration milestone was to integrate the microcontroller with the circuits and sensors. This was done once the circuits had been fully assembled on the solderless breadboard, and the core of the microcontroller code had been written. It consisted of testing that the microcontroller was able to correctly read whether or not a battery was charged when the battery's terminals were connected to the voltage reading circuit's input using alligator clips, and verifying that the microcontroller was able to read whether or not the photoresistor was manually shaded from light.

No problems were encountered in this integration stage.

## 11.2 Stage 2: Further Microcontroller and Circuit Integration (06/03/2017)

This integration milestone consisted of repeating the same tests as the previous integration milestone, but with the circuit finalized and soldered and the microcontroller code updated to implement the full FSM, so that it could be verified that no errors had been introduced in these steps.

No problems were encountered in this integration stage.

## 11.3 Stage 3: Initial System Integration (11/03/2017)

This integration milestone involved constructing the full frame for the electromechanical system, mounting the servos to control the blocks and ramp and rate limiter, and connecting the electromechanical system to the microcontroller and circuits.

Two major problems were encountered in this integration stage:

- The servos were found to twitch as a result of an unknown error. This was later discovered to be a result of an error in the PWM code.
- The light sensors were found to always show up as blocked, since the tubes themselves blocked a considerable amount of light. This was resolved by adding LED's opposite the sensors.

## 11.4 Stage 4: Ongoing System Integration (24/03/2017)

This integration milestone involved a significant number of steps towards integration:

- Attaching electrode tower to the machine, soldering on electrodes to integrate with circuits, mounting servos controlled by the microcontroller to raise and lower the electrode tower.
- Finish main wiring and soldering.
- Finish software so that it can do a complete run with a single battery.

Two major issues were encountered:

- Batteries had a strong tendency to jam when the blocks tried to move forward; this was later resolved through sanding of the floors of the polycarbonate stack.
- Servos continually lost calibration for no apparent reason. The issue turned out to be variable time taken by processing that altered the pulse-width output to the servos by delaying the code that produces the PWM.

## 11.5 Stage 5: Ongoing System Integration (30/03/2017)

This integration stage revolved around constructing a new rate limiter design that did not have the issue that the previous one had, where multiple batteries would be dropped at a time. The new design was relatively effective, however it was still susceptible to jams when many batteries were loaded at once.

## 11.6 Stage 6: Completed System Integration (10/04/2017)

This integration stage was the final one, requiring the team to complete the machine to be ready for the project demo. A number of tasks were carried out:

- Attach a scooping arm to the rate limiter to mitigate jamming.
- Considerable amounts of sanding to reduce jamming.
- Update code to agitate with actuators in order to clear jams, and to correctly detect and handle jams.

The key problem of this integration stage was the difficulty in fully avoiding jamming in the system. The final machine was reliable and successful under favourable loading conditions, but was not able to avoid jams in all cases, such as when the batteries were poured into the rate limiter in an unordered fashion.

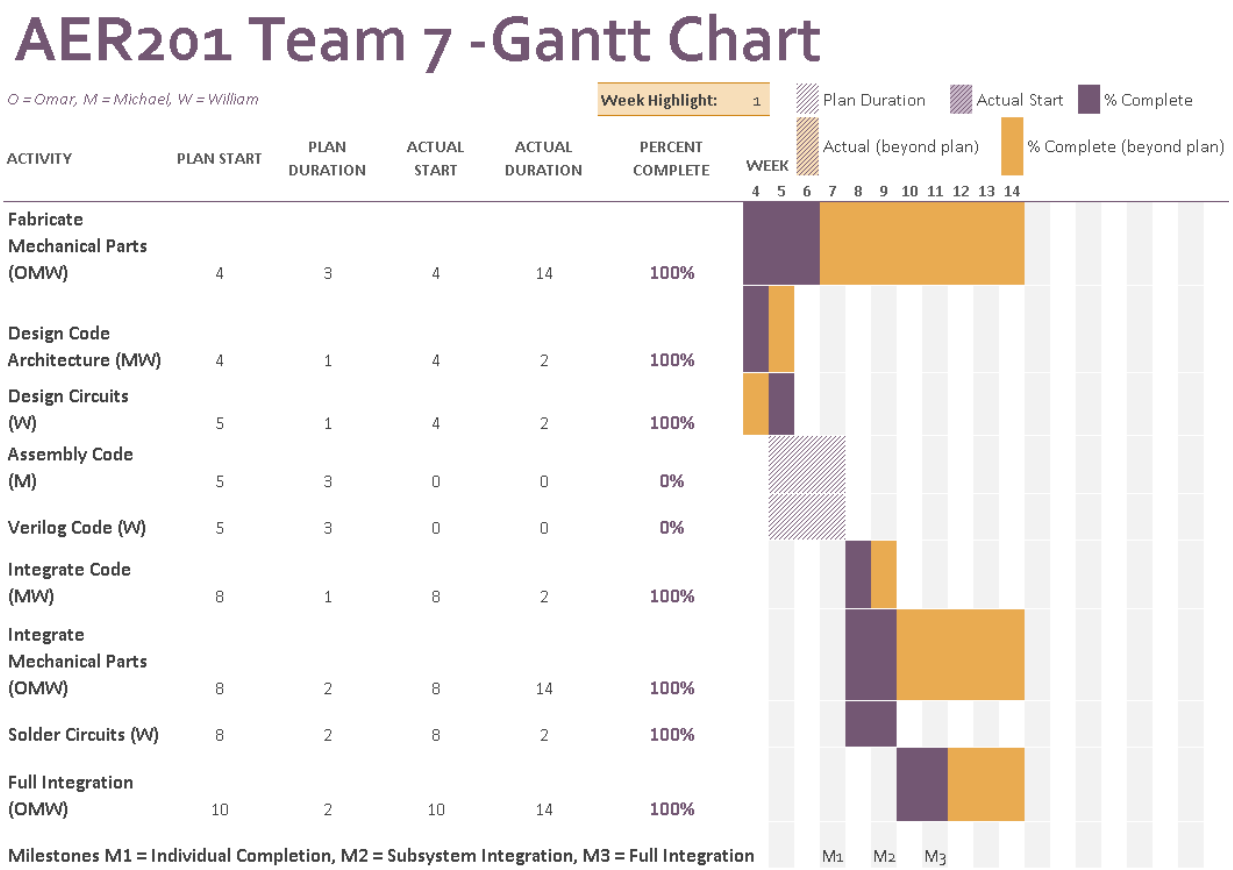
# 12. System Improvement Suggestions

The design suffered from a number of design limitations that prevented it from optimally solving the problem and meeting the team's design goals:

- The design was inherently sensitive to slight errors in fabrication, causing it to jam. This was exacerbated by the team's relative inexperience in carpentry and machining, and the rushed nature of the project. This could be improved by a design that placed more emphasis on passive sorting and less on precise and repeating mechanisms that need to be calibrated perfectly.
- The design was purely serial, imposing an upper bound on how quickly it is capable of sorting batteries. A design that could avoid this might involve sorting the batteries by type first and then measuring the voltages of the 3 types of batteries separately and in parallel, to triple the throughput.
- The physical construction of the machine relied too heavily on glue and the positioning of the circuits relative to the actuators meant that parts did not tend to be removable and it was difficult to work with parts inside the machine without disturbing the wiring. This led to difficulty in maintaining and modifying the machine once it was constructed and caused delays in the construction and debugging process.
- The lower electrodes for the 9V battery voltage sensors were unreliable, as they did not always show a voltage for charged 9V batteries with their terminals down because the batteries would often be raised by the thickness of the wires leading to the electrodes. This problem could have been avoided by embedding the 9V electrodes directly into the floor, with the wires coming out the bottom.

Overall these design weaknesses were a result of the team not having experience doing this type of design project, and their discovery was a valuable learning experience for the members of the team.

# 13. Initial and Accomplished Schedule (Gantt charts)



## AER201 Team 7 -Gantt Chart

*O = Omar, M = Michael, W = William*

Week Highlight: 1

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | ACTUAL DURATION | PERCENT COMPLETE |
|---|---|---|---|---|---|
| Fabricate Mechanical Parts (OMW) | 4 | 3 | 4 | 14 | 100% |
| Design Code Architecture (MW) | 4 | 1 | 4 | 2 | 100% |
| Design Circuits (W) | 5 | 1 | 4 | 2 | 100% |
| Assembly Code (M) | 5 | 3 | 0 | 0 | 0% |
| Verilog Code (W) | 5 | 3 | 0 | 0 | 0% |
| Integrate Code (MW) | 8 | 1 | 8 | 2 | 100% |
| Integrate Mechanical Parts (OMW) | 8 | 2 | 8 | 14 | 100% |
| Solder Circuits (W) | 8 | 2 | 8 | 2 | 100% |
| Full Integration (OMW) | 10 | 2 | 10 | 14 | 100% |



Milestones M1 = Individual Completion, M2 = Subsystem Integration, M3 = Full Integration

# 14. Description of Overall Machine (Figures, Tables)

## 14.1 Full machine

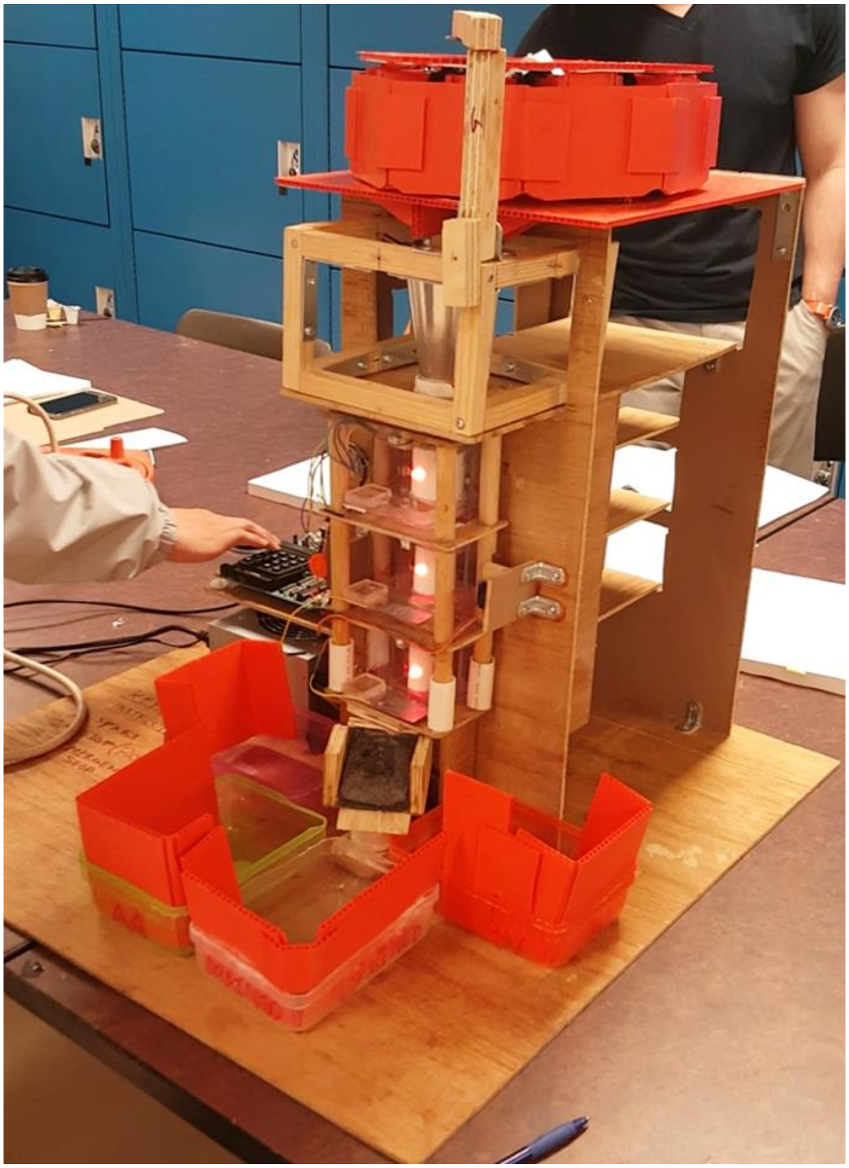


**Figure 14.1** Full image of the machine

**Figure 14.1.2** Front view of main machine. Rate limiter, funnel, block system + electrode tower, ramp + bins

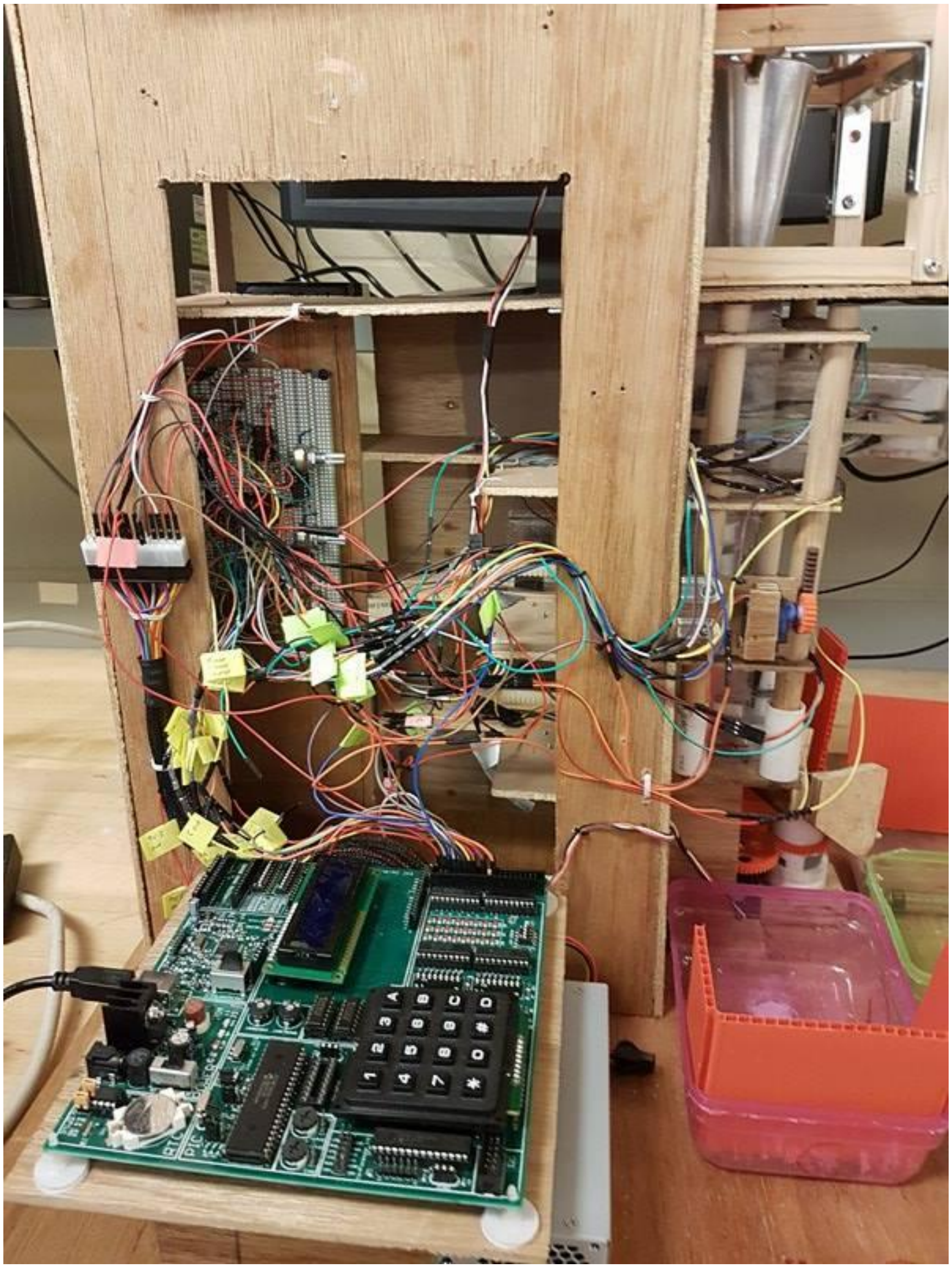


**Figure 14.1.3** Side view of funnel, block system + electrode tower, ramp + bins, circuit board and PIC microcontroller.

## 14.2 Rate Limiter

At the top of the machine is the rate limiter, where the unsorted batteries are placed. The metallic dome allowed the batteries to slide into one of the 8 holes positioned along the circumference of the rate limiter.

**Figure 14.2.1** Top view of rate limiter

Arched over the rate limiter was the scooper whose purpose was to scoop up any jammed batteries and push them into one of the holes around the circumference.

**Figure 14.2.2** Close up view of the scooper in rate limiter

The rate limiter is positioned right above the funnel

**Figure 14.2.3** Side view of rate limiter on top of machine

## 14.3 Funnel

The funnel collects the batteries dropped form the rate limiter and orients them vertically.

**Figure 14.3.1** Diagonal top view of funnel system below rate limiter opening

## 14.4 Block system and Electrode Tower

The blocks are responsible for collecting the batteries and determining their type (C, AA, or, 9V). The electrode tower measures the voltage in order to determine whether they are charged or uncharged.

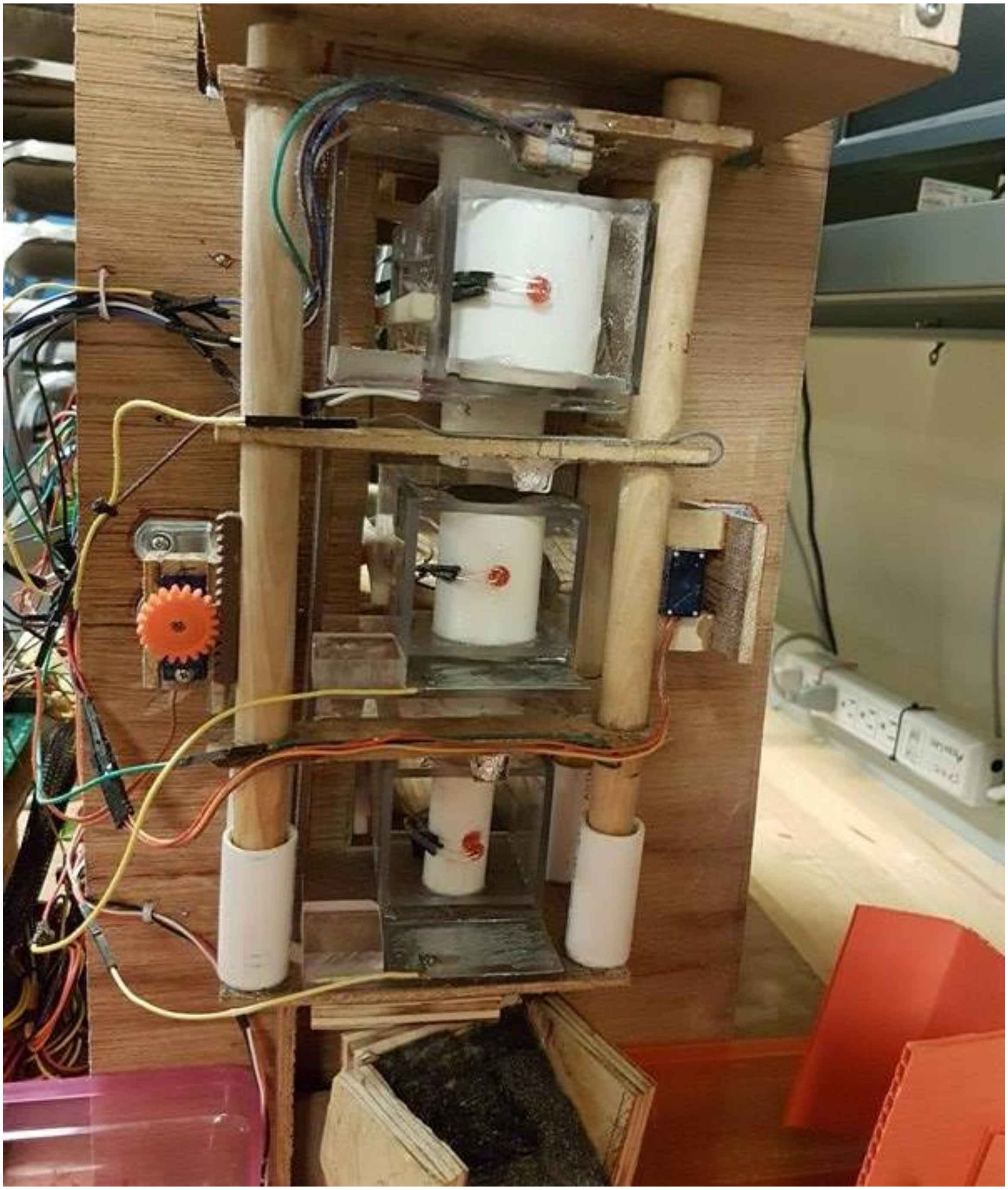


**Figure 14.4.1** Front view of block system and electrode tower

**Table 14.4** Position of the battery blocks relative to the system

| Relative Height of Block | Battery Type in Block |
|---|---|
| Top | 9V |
| Middle | C |
| Bottom | AA |

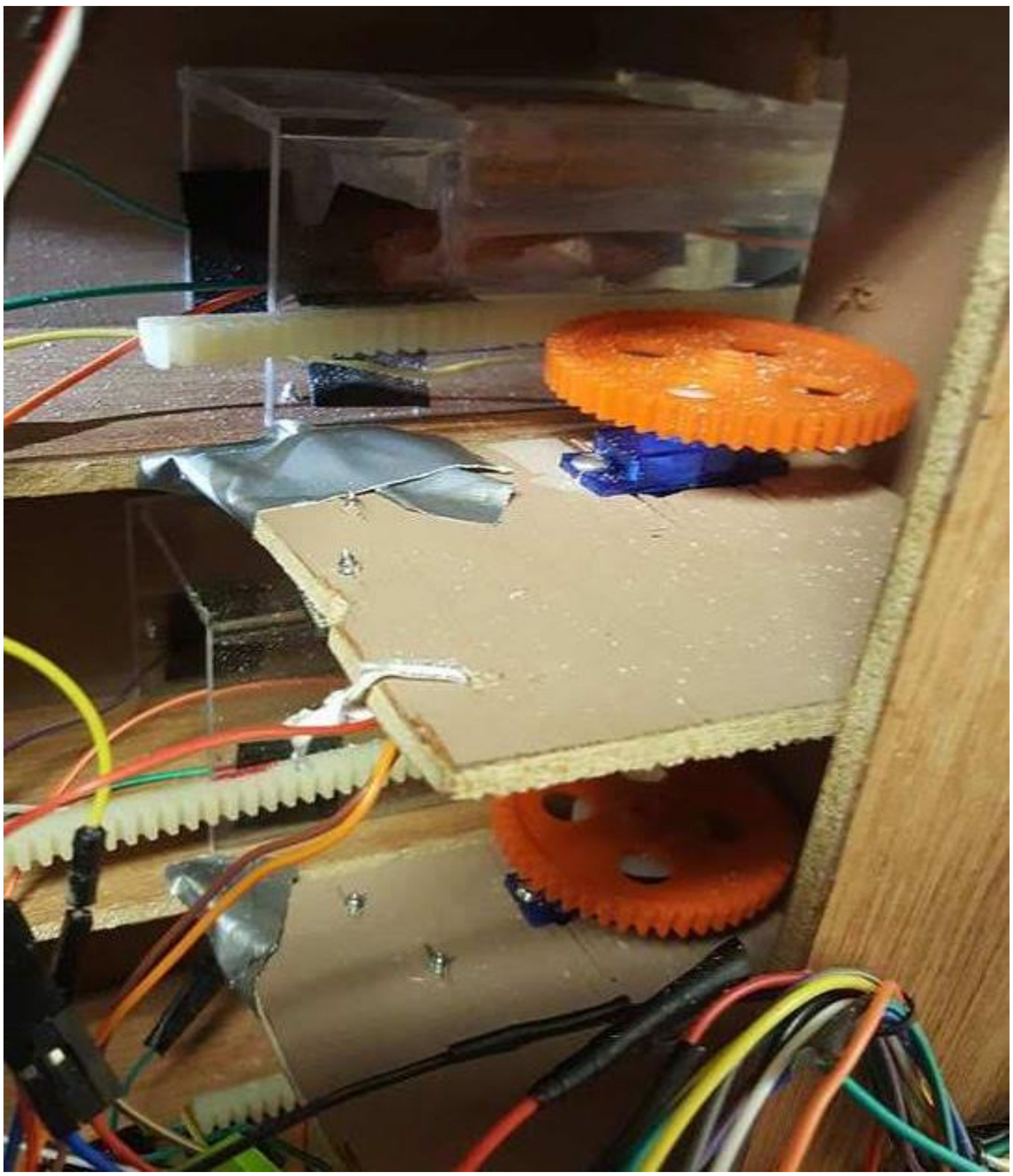


**Figure 14.4.2** View of servo gear on rack and pinion

**Figure 14.4.3** Image of 9V block extended

Once a battery falls into the slot in a battery block, it is moved forward, and the electrode tower lower to measure the voltage.

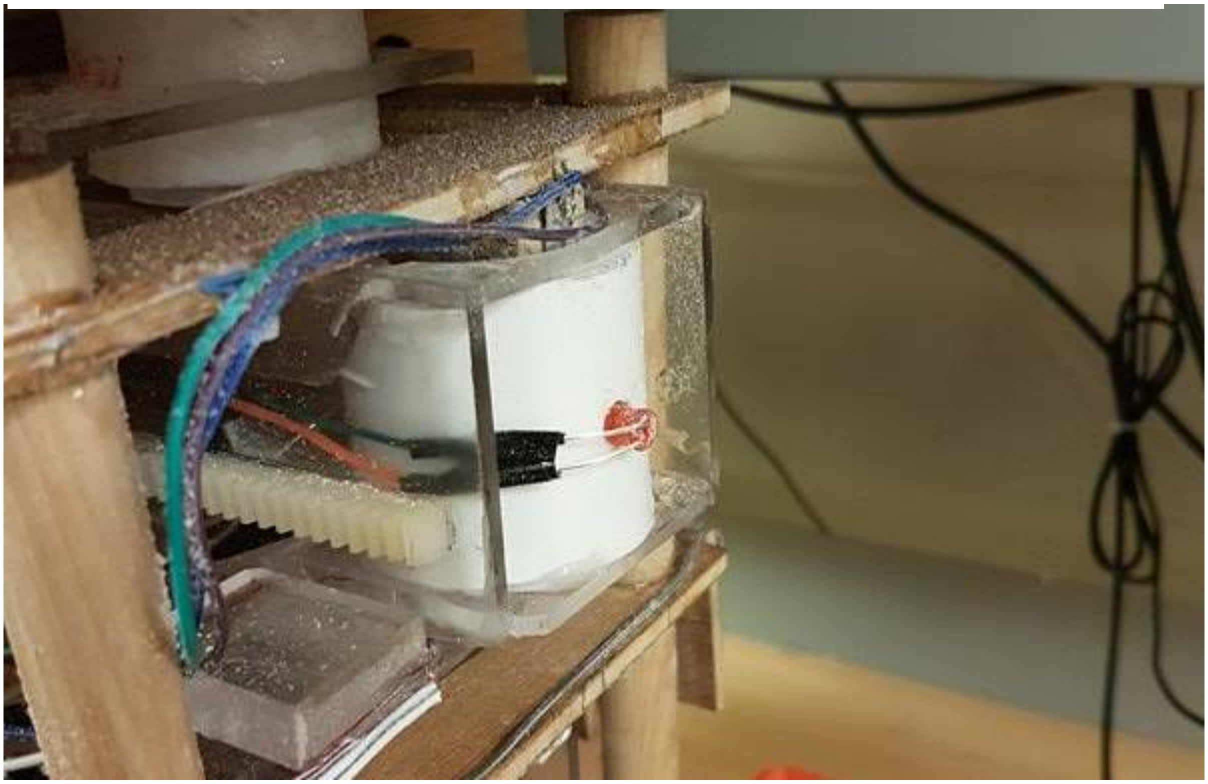


**Figure 14.4.4** Image of electrode tower lowered, with electrode inside 9V block

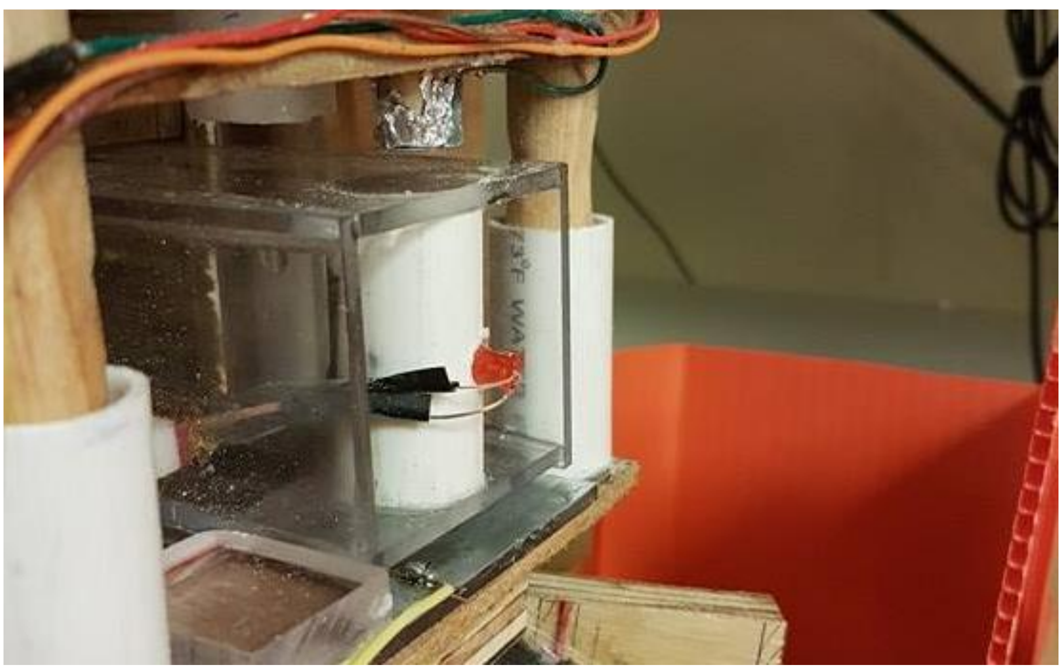


**Figure 14.4.5** Image of electrode tower raised above AA block

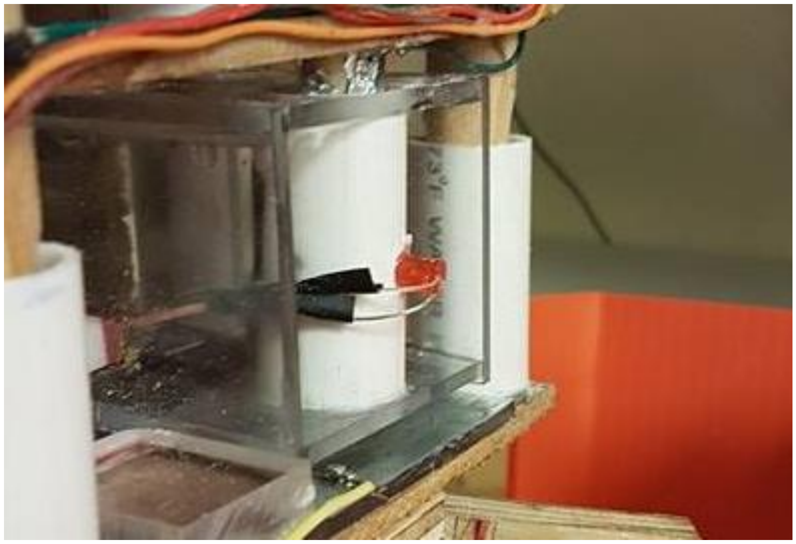


**Figure 14.4.6** Image of electrode tower lowered into AA block

## 14.5 Ramp and Container System

The ramp and container system is responsible for collecting the batteries and disposing them into the correct container.

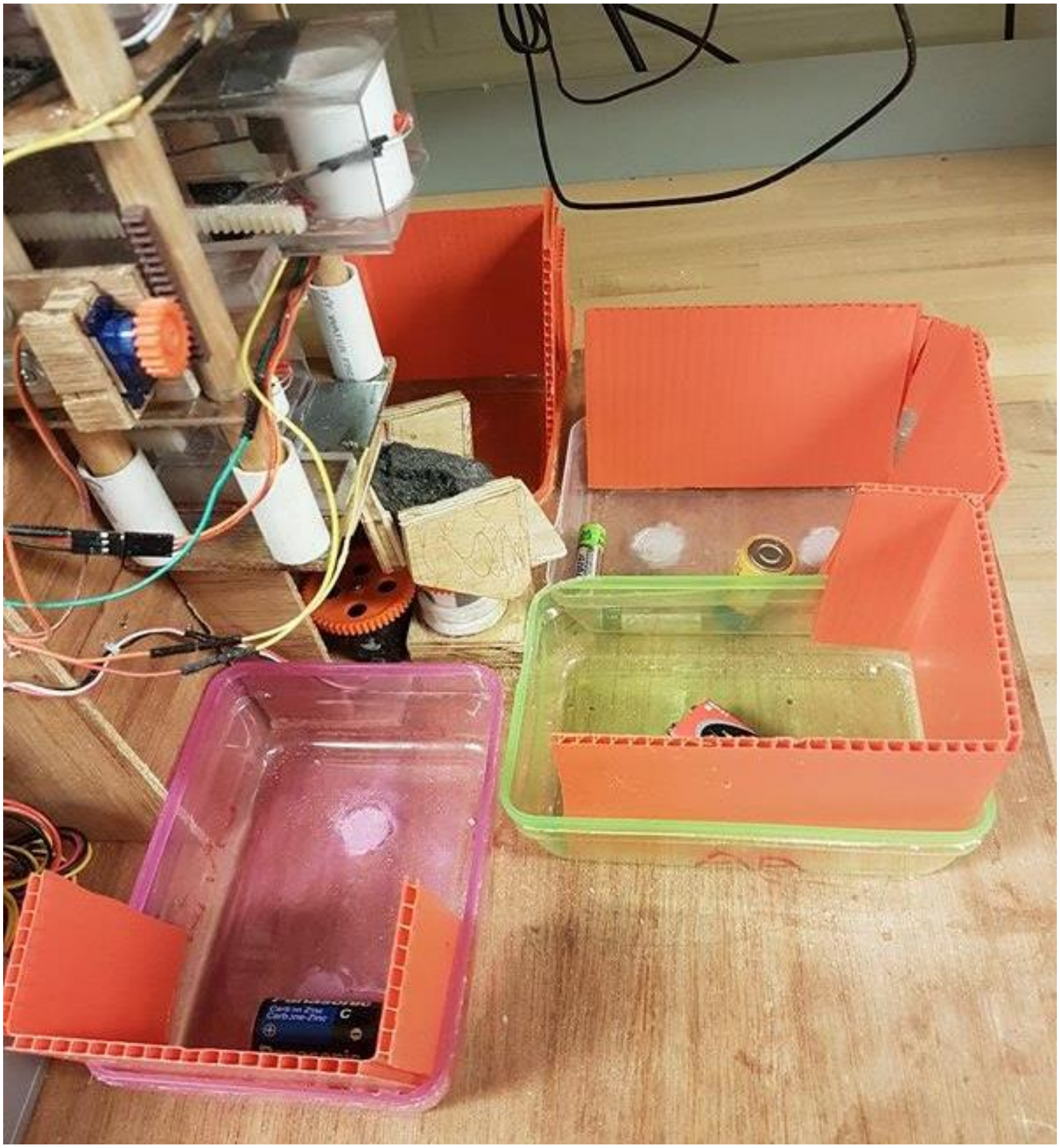


**Figure 14.5.1** View of the ramp system with the C block extended. A battery in the C block would fall down onto the ramp and land in the bin it is pointing to.

**Table 14.5** Container for each battery type

| Container Colour | Battery Type |
|---|---|
| Pink | C |
| Green | AA |
| White | Drained |
| Orange | 9V |

# 15. Standard Operating Procedure

The team's solution consisted of a rotating circular loading tray with a flat rim and domed interior, with battery-sized holes embedded into the floor around the rim in even intervals. The unsorted batteries were placed into the loading tray, where they would either naturally fall into the holes or be guided into them by a fixed wooden scoop mounted to the frame of the machine. The tray's rotation would cause the batteries to drop one at a time from the slots into a flexible spring-supported funnel. The opening at the bottom of the funnel was small enough that batteries could only pass through in a vertical orientation, and the springs were sufficient that the batteries randomly bounced enough that it was practically certain that the battery would align with the slot without the need for an active actuator.

Below the funnel was a segment of pipe with circular cross-section that is large enough to fit any of the 3 types of battery but small enough that it forced any 9V battery inside it to be directly upright. The pipe had exactly the same height as a 9V battery, so that if there was a 9V battery inside it, it fit tightly. The pipe segment was embedded within a larger block with a rack and pinion mechanism allowing a servo motor to move the block back or forward.

Additionally, it was equipped with a light sensor that allowed it to detect whether or not a battery was present in the slot. Below this pipe/block structure was another similar pipe and block, except this pipe had a radius such that a 9V battery could not fit inside it but a C or AA battery could, and below this was one that can only fit a AA battery. By having this system of tubes of decreasing radius, a battery that fell through the funnel at the top of the system would fall under gravity to the lowest slot that fit it. This allowed the system to easily separate a given battery into a different slot based on its type, under the power of gravity alone.

Once the light sensors detected that a slot contained the appropriate battery, the block was pushed forward to an electrode station where the system determined whether the battery was drained or charged. While this was happening, the solid top surface of the block prevented any additional batteries from falling through from the slots above. When the battery has been identified as charged or drained, the ramp was positioned towards the appropriate container so that the block can simply be moved further forward to a point where there is no floor below the tube, allowing the battery to fall out onto the ramp and then into the container due to gravity.

The block was then returned to its original position and the process repeated until all of the batteries have been processed.

# 16. Conclusions

Overall, the project was a mixed success. While the machine was unable to achieve a qualifying run during the project demo, it is the belief of the team that the core design of the machine was strong and given a chance to build a new one with the knowledge acquired along the way, would be able to perform very well.

To characterize the quality of the design, it is useful to refer back to the design objectives specified earlier in this document and evaluate the design according to them:

1. **Modularity:** the final design featured a high degree of modularity in principle, however the physical construction made some parts hard to reach. Overall, the machine does well according to this criterion, and the problems of hard to reach parts could be avoided with a relatively minimal redesign.
2. **Reliability:** the machine did not exhibit a high degree of reliability unless given batteries loaded in a favourable way. This is a sign of a defect in the machine's design and construction, but not to the extent of considering the design a failure. A redesign focusing on precision in fabrication and with a design more able to tolerate fabrication errors would not be difficult and would greatly mitigate this flaw.
3. **Accuracy:** the machine was highly consistent in placing batteries into the correct bin when it was able to sort them without a jam, so it rates highly in accuracy. The greatest weakness of the system was the unreliable lower electrodes for the 9V battery voltage readers, which did not always show a voltage for charged 9V batteries with their terminals down because the batteries would often be raised by the thickness of the wires leading to the electrodes. This problem could have been avoided by embedding the 9V electrodes directly into the floor, with the wires coming out the bottom.

Given this evaluation with respect to the design values, the design was overall a good one, but suffered from some avoidable design and fabrication errors. The team is glad to have had the chance to work on this project and has learned a considerable amount about practical engineering design, professional engineering practice and team organization.

# 17. References and Bibliography

[1]"Duracell vs Energizer 9 volt battery. Are all 9V batteries the same?", *Medicbatteries.com*, 2017. [Online]. Available: https://www.medicbatteries.com/Duracell-vs-Energizer-9-Volt-battery-9-volt-batteries-9v-battery-9v-batteries. [Accessed: 31- Jan- 2017].

[2]"Primary batteries – Part 2: Physical and electrical specifications", *IEC 60086-2*, 2017. [Online]. Available: http://image.sciencenet.cn/olddata/kexue.com.cn/bbs/upload/15791IEC60086-2_%7B10[1][1].1%7D(2001-10).pdf. [Accessed: 31- Jan- 2017].

[3]"AA Battery Data Sheet", 2017. [Online]. Available: http://data.energizer.com/pdfs/lr6cl_eu.pdf. [Accessed: 31- Jan- 2017].

[4]"Patent US20080037363 - Agitation system and method for measuring settling rate of solids from a suspension", *Google Books*, 2017. [Online]. Available: https://www.google.ch/patents/US20080037363. [Accessed: 31- Jan- 2017].

[5]"Patent US3480140 - Testing and sorting apparatus for batteries", *Google Books*, 2017. [Online]. Available: https://www.google.com/patents/US3480140. [Accessed: 31- Jan- 2017].

[6]"US5253765A - Sorting and grading system - Google Patents", *Patents.google.com*, 2017. [Online]. Available: https://patents.google.com/patent/US5253765A/en?q=sorting&q=based&q=shape. [Accessed: 01- Feb- 2017].

[7] Refind Technologies. (2012, Oct 31) Optisort Battery Sorter. [YouTube video]. Available: https://www.youtube.com/watch?v=OugqnVO7WiU. Accessed Jan, 2017.

[8] Refind Technologies. (2014, Mar 17). The OBS 600 battery sorter from Refind Technologies. [YouTube video]. Available: https://www.youtube.com/watch?v=zt_tvo2p264. Accessed Jan, 2017.

[9] Refind Technologies. (2016, Dec 2). The OBS 200 battery sorter from Refind Technologies. [YouTube video]. Available: https://www.youtube.com/watch?v=kp1TEkf_kyE. Accessed Jan, 2017.

[10] Technologies, Terasic. "Terasic - DE Main Boards - Cyclone - DE0-Nano Development And Education Board". *De0-nano.terasic.com*. N.p., 2017. Web. 1 Feb. 2017.

[11] *Nios II Classic Processor Reference Guide, 2016*

[12] *Introduction to the Altera Qsys System Integration Tool* N.p., 2017. Web. 1 Feb. 2017.

[13] "SB330S-E3/54 Vishay | Mouser Canada". *Mouser Electronics*. N.p., 2017. Web. 1 Feb. 2017.

[14] "Recycling of batteries: a review of current processes and technologies", *Sciencedirect.com*, 2017. [Online]. Available:

http://www.sciencedirect.com/science/article/pii/S0378775303012230. [Accessed: 01- Feb- 2017].

[15] "9g Continuous Rotation Micro Servo", *Robotshop.com*, 2017. [Online]. Available: http://www.robotshop.com/ca/en/9g-continuous-rotation-micro-servo.html?gclid=COfY3LaMmtMCFYGFaQodwFUPYQ. [Accessed: 12- Apr- 2017].

[16] "How Are Polycarbonate & Acrylic Different", *The Plastic People : : Blog*, 2017. [Online]. Available: http://blog.theplasticpeople.co.uk/2012/06/01/how-are-polycarbonate-acrylic-different/. [Accessed: 01- Feb- 2017].

[17] "DFRobot Micro Servo Motor", *Robotshop.com*, 2017. [Online]. Available: http://www.robotshop.com/ca/en/dfrobot-micro-servo-motor.html. [Accessed: 13- Apr- 2017].

[18] A. Accessories, "Addicore SG90 9g Mini Servo", *Addicore*, 2017. [Online]. Available: http://www.addicore.com/Addicore-SG90-Mini-Servo-p/113.htm. [Accessed: 12- Apr- 2017].

[19] *NTE3019 LED Datasheet*, 1st ed. NTE Electronics Inc., 2017.