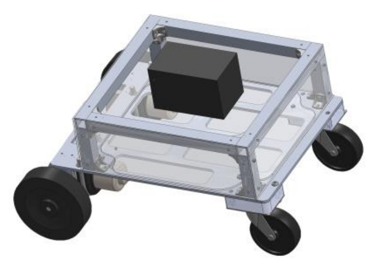
KYLE is an automated, user friendly robotic follower, developed to assist persons of all ability levels in daily tasks. Built around the parameters of simplicity and affordability, KYLE is intended to be a regular companion, carrying shopping bags, school books, laundry and more.

KYLE

The Robotic Follower



Timm Bischoff
Steven Krygier
Stephanie Legg
Melinda Kathryn Liegl

Fall 2016 Capstone Project

Mechanical Engineering
University of South Florida

Table of Contents

List of Figures

List of Tables

Introduction

Mobility is key to quality of life.

It feeds into one's sense of independence and self-worth in both subtle and profound ways. The sudden or gradual loss of one's mobility can be deeply traumatic. The need for an assistive device, such as a wheelchair, walker or crutches, is likely for all at some point in their lives. While technology is absolutely expanding in this market, the basic design hasn't altered so much as to be unidentifiable; crutches bought today are very similar to those used by our grandparents' grandparents.

Kyle, the Robotic Follower is an easy-to-use, modular and affordable device designed specifically to expand carrying capacity and extend independence. It can be used in tandem with an assistive device, or by anyone in need of a little help accomplishing their daily physical tasks. Whether carrying books across campus for a student who uses crutches, following an elderly couple in the airport with their luggage, dutifully carrying groceries around a store and to a car, or shuttling laundry baskets for someone recovering from surgery, Kyle is a safe, simple, dependable assistant.

Project Description

From day one, this project was focused on the end user.

The question of who would eventually use, need or enjoy this kind of assistance was so paramount, that several group meetings were structured specifically around defining and expanding the consumer demographic. At first, the scope was narrowly defined to include only wheelchair users in a grocery store. However, it was quickly expanded to include more people as the sketches were developed, and it became apparent that the modular design encouraged a whole host of components/additions that would make a niche product more universal.

Initially envisioned to include a scissor lift (helpfully provided by Steven Sundarro from a previous Capstone group) and a basket attachment, Kyle evolved into a scalable basic component that could be modified in multiple ways to meet multiple needs.

Figure 1: Parameters and Specifications

In **Figure 1** (above), Kyle's parameters and specifications are laid out in a priority pattern. While there are 18 key points defined for the design, the three most important are safety, reliability and accuracy. Most of the sweat and tears in developing this project were a result from trying to ensure it would work accurately, and not follow unintended objects or people. As the design team is made of 100% mechanical engineering students, programming is not the main focus of the scholastically supplied skillset.

The second level of importance was centered on human factors, and the need to design a device that wouldn't add to stigma felt by users. Many people who could use audial assistive devices eschew them based on the psychological impact of perceived diminished strength and ability, so it was important to consider how the design would make a user feel if they used it in public, or began to rely upon it at home. One solution to reduce the stigma of needing/wanting this kind of help was expanding the market for a robotic follower by branding it as a universally useful "techy" item, like an iPhone. If having a robotic follower assistant became common, then the more vulnerable initial target demographic wouldn't hesitate to also use this product.

The third and fourth levels were so closely related in importance, they could be interchangeable. If those who could benefit could be convinced to actually *use* a follower, then it absolutely needed to be efficient, affordable and easy to use. For the prototype purposes, Kyle is currently heavier (around 30 lbs.) than the next iteration would be. If work continued on this project and a second version was made, more plastics and acrylic would be utilized to reduce cost and weight. The most expensive single item in the Bill of Materials (that became part of the presented prototype), was the metal (more than $90). So many devices in the rehabilitation field are prohibitively expensive, especially for those who need short term assistance. In high production, this product's cost should aim to cost less than $200 to manufacture and retail for less than $400.

So much emphasis has been placed on this simple product's modular concept, because that is what sets Kyle apart from other similar products in development or already available in the market. This robotic follower is adaptable in a way that promotes creativity from manufacturers and a potential online community of end users. The code can be made open source, inviting people to tweak and post revised versions and new ways to get Kyle to help meet new challenges. The company making this product would focus only on the scale/size constraints, motor torque, and safety. Perhaps also offering a number of colors or finishing options. Other manufacturers could capitalize on this and provide attachments to fit on, into or over Kyle. This removes the limitation that it is only a shopping cart, and opens the possibility of Kyle as a gardening assistant, stroller (with car seat attachment), autonomous luggage set, a toy car for visiting grandkids, search and rescue device, household chore helper, etc.

Of course, in order to be useful in such a myriad of ways, the battery has to be small/light enough not to burden the end user, safe enough for a wide variety of uses and efficient/reliable enough to hold adequate charge. The specifications for this need were based on those of a mid-line electric wheelchair, and it was decided that it needed an operating life of at least two hours. This would naturally change given the load that it was under, but two solid hours of use was the defined end goal.

Levels five and six may have fallen to the end of the priority tree, but they are still important in Kyle's design. In order to be adaptable, the design needed to be pleasant, simple, have clean lines and let whatever attachment was in use dominate the look. In order to be more sustainable, it was important for Kyle to use readily available, inexpensive, standard parts that promote the repair of the device, over its destruction. Publishing repair guides, encouraging users to post their own and providing a website with links to all the parts and fasteners would hopefully keep many Kyles out of the landfill. Having the code be open source would also extend the life of this follower, by providing options for current and future cell phone software developments. Since cost is so important, deciding from the beginning to create a basic, easy to repair, easy to modify device, would hopefully prevent the current tech trend of controlling all aspects of the product and its peripherals, rendering current iterations useless after one year.

Final Design

After several designs were generated and meetings were held, the final design version of Kyle incorporates almost all of the initial parameters and specifications. In **Figure 2** (page 9), the basic dimensions are shown, but the main purpose to is show Kyle in context of its application. In **Figure 3** (page 9), some its "elevator speech" talking points are shown, but the CAD model shows the follower, without the electrical box, and with transparent sides.

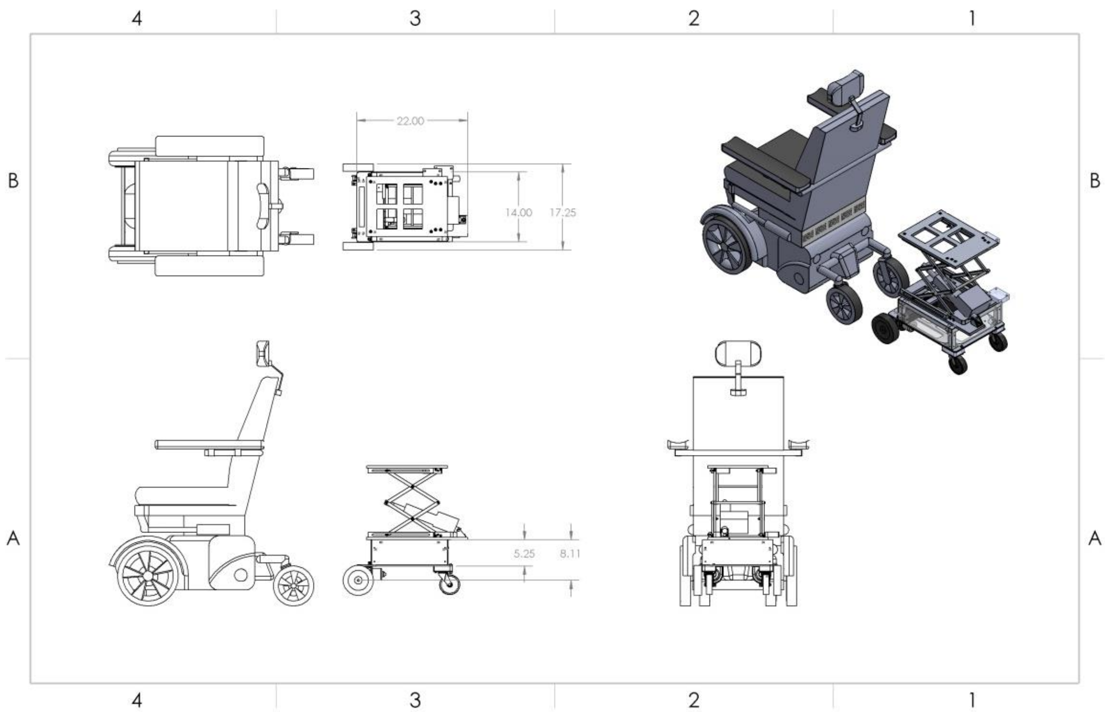


Figure 2: Kyle with a scissor lift attachment, following an electric wheelchair

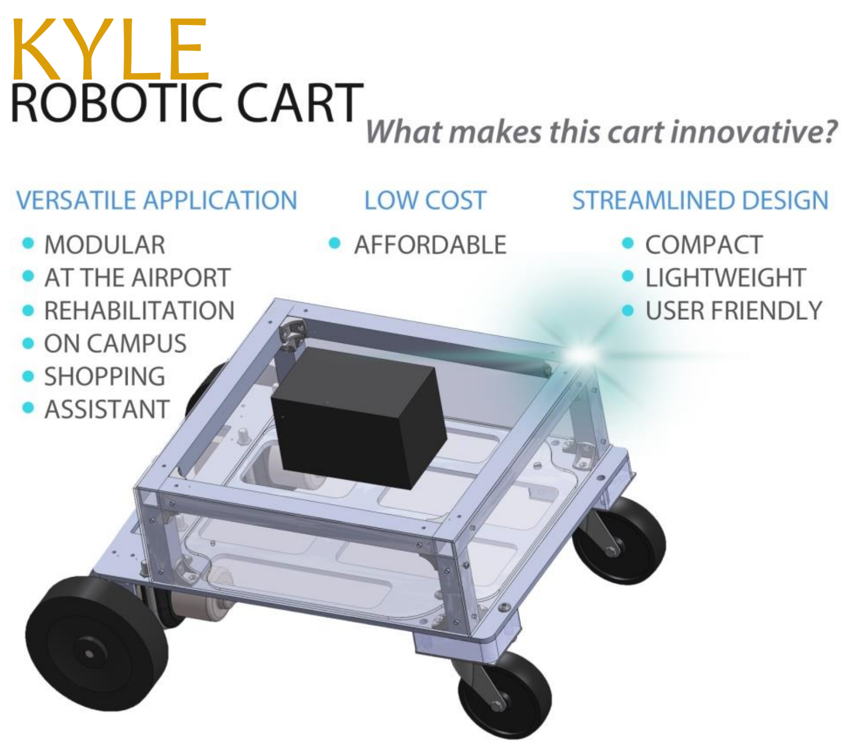


Figure 3: Kyle; modular, affordable, user friendly

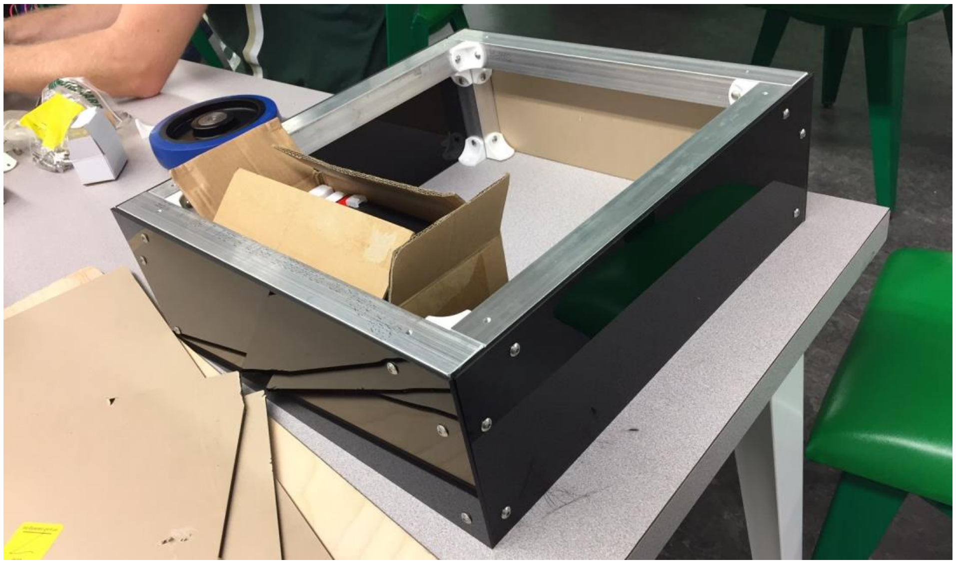


Figure 4: In the assembly, Kyle has black acrylic sides

Figure 5: Stephanie & Timm assembling the base, Steven working on electronics

Figure 4 (on page 10) shows the robotic follower being assembled in the DfX Lab. While the parts didn't come from the machine shop with the designed rounding of the baseplate, that is a small issue that could be addressed in a future iteration of Kyle, and doesn't affect the outcome of the prototype at this stage. Otherwise, the machined parts were very nicely finished and allowed for simple assembly. While not clearly shown in the **Figure 5** (page 10), Steven, who also worked on the CAD models and finding materials, was working hard on getting the electronics and coding to work well, once the base was assembled. Stephanie, shown stabilizing the base, has also put days in on the coding and design of the electronics, and learned Python for this project. Timm's manufacturing background has helped at every step in the CAD design and the ordering of parts; the final CAD design taken to the shop was his. Kat, who took the photograph, has been helpful in the presentations, writing, keeping to the timelines and in general project organization.

Below (**Figure 6**) is the assembled final design for submission in USF's Fall 2016 Capstone.

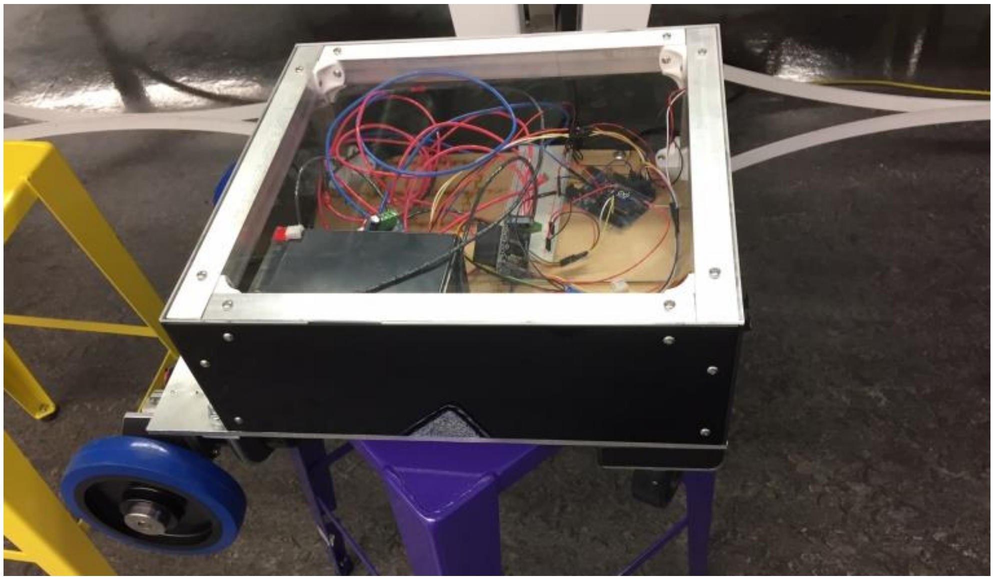


Figure 6: Final Assembly

Pros & Cons of the Design

The final design for this project was able to satisfy so many its initial design specifications and parameters, and that was not by accident. When vetting the designs each teammate produced, the following chart (**Figure 7**) was created and used as a tool to further qualify what was necessary to accomplish the task. It includes several ranked "customer attributes" from high (H), medium (M) and low (L) along the left hand side; more evident ones in yellow, more subconscious drivers in green. In blue, along the top, are listed some of the technical aspects within the engineers' design control. In the graph where the customer priorities and the engineer options meet, correlations were listed as strong connection (S), connected (C), and weak connection (W). Even correlations listed as W were still important enough to list, and were considered in the evaluation and eventual selection of the final design.

H= High Importance
M = Medium Importance
L = Low Importance

S = Strong Connection
C = Connection
W= Weak Connection

Customer Attributes	Importance	Light Materials	Multiple Functions	Function Obvious	Avoid Toxic Materials	Use Standard Parts	Deisgn for Assembly/ Disassembly	Use Raspberry pi/ Python	Use Arduino/ Sketches	Use Recylced Parts	Easy to Recycle
Reliable	H		S	C		S	W			S	
Easy to Use	H		S	S				S	S		
Affordable	M	S			S	S	S	C	C	S	
Safe	M			S	S	S	S			C	
Modular	H		S	S		S	S	C	C		S
Attractive	M			S						C	
Easy to Repair	M		S	S	S	S	S	S	S		S
Lightweight	L	S				S					
Battery Life	H	S	S			C		C	C		C
Efficient	M	S	C	C		C	C	C	C	C	
Long Life Cycle	H	C	C	S	S	S	S	C	C	C	S
Android/Apple Compatible	L							S	S		

Figure 7: Kyle's House of Quality

Because so much time and work were put in at the beginning of the design process, Kyle embodies much of what was initially specified. That is not to say that there are no problems or things that would be done differently, if attempted again. However it may lead to why this Pro/Con comparison may seem a little overly optimistic and simplistic.

Pro	Con
Sized correctly	Heavier than ideal
Clean initial prototype	Coding not seamless
Easy to assembly, repair, clean	Program not yet effortless in use
Designed to carry 25lbs. Can	May want more carrying capacity
Low center of gravity, high stability	May be a trip hazard

Table 1: Pros and Cons

As shown in the table, while the size lends itself to multiple applications, easy movement, turning and storing, the initial prototype is heavier than ideal. The next iteration of the design process would involve steps to reduce the overall weight by at least one third. It is important to keep the weight as low as possible for Kyle because of what it may be tasked to carry, and how that load will affect the performance of the motors and battery.

While pleased with the visual and functional outcome of the first prototype of Kyle, the coding continues to be problematic. Several computer science and computer engineering students were consulted, and while the initial set up involved an Xbox Kinect, Raspberry Pi microcontroller and required the Python programming language, the group decided to go back to an Arduino microcontroller and the Arduino Sketch language, with which each member was more familiar, to ensure a working prototype was available by the semester's end. However, in future versions, Raspberry Pi and Kinect would open up a whole new world of possibilities and applications, and should be pursued. Also, the programming of iPhone and Android enabled apps was a new endeavor and produced less than ideal results. A hallmark to this concept is ease of use, so a deeper dive into the development of Kyle as an actual product would need both the internal coding and the cell phone user interface to be extensively redone.

While it is able to carry its own weight plus a small load, iterations beyond this proof-of-concept would need to focus on optimizing the ratios and allow for a higher max load, with a safety factor built in. Also, in considering safety and fail scenarios, its low center of gravity is great for navigating turns and stability, but one can't help but acknowledge that at less than one foot tall, it would make for a trip hazard, if no attachments were employed. Perhaps this can be handled in the coding; built obstacle avoidance into it so that any time it is on, it will actively avoid people walking towards it, if given no other commands, and make a beep noise/flash lights if it is cornered.

One the positive side, the final design is simple to assemble, repair and clean. There could be a self-assembly option for consumers, driving down the end user's cost even further. Replacement parts could be made easy available, and repair guides/videos could show exactly how to get Kyle back up and operating, ASAP.

Effectiveness of the Prototype

The Kyle prototype fits its specifications and this group is generally pleased with its development. As far as its effectiveness, however, lets' consider two paths; whether it can carry the load, and whether it can follow the target.

The motors were tested in the DfX lab and have the RPM and torque to carry around its body and a small load, with no issues. The battery seems completely sufficient for the task, currently, though were the model scaled up; it would need to have both the motors and battery scaled up in conjunction. Given the end-of-semester rush and delay in getting machined parts, extensive testing was not a possibility. Ideally, testing to failure with the battery and the motors would have been conducted. However informal tests in the lab have been positive; it can carry itself and a load. It is this group's opinion that this robotic follower will be effective in its assigned carrying task, but without failure testing to quantify it, how much and how long are unknown.

At the time of writing this paper, testing and iterative changes to the programming are still underway. While the group is optimistic, the current goal is to make sure Kyle can follow a leader without being distracted by other objects, moving or stationary. The development of sophisticated testing procedures, let alone actual testing and analysis, is just not the stage where this project is, today. Therefore, evaluation of the programmatic effectiveness is still to be determined, by future testing and prototype development.

Cost

While the budget for Capstone was not limiting, what was learned during the course shows the need for low or lower cost options for people with mobility issues or rehabilitation needs. As mentioned earlier in this paper, the demographic was widened to try to lower the stigma of using an assistive device, but the core target user is still envisioned as someone with a more immediate need for help, and less personal mobility. Many vulnerable populations in Florida and the US are also people on fixed incomes. Therefore a priority is making a product with a long life cycle that is easy to repair, resilient with technology changes (iPhone/Android apps, etc.) and using the most cost effective materials possible.

Table 2 shows Kyle's Bill of Materials. It is important to note that this includes only the parts that were used in the final assembly, and does not include the two donated windshield wiper motors that were recycled from a past group's project. It also does not include the Xbox Kinect, or the Raspberry Pi. If all the materials used in development were listed, the figure jumps to over well over $600. However, the items listed below are a good recipe of what the basic follower would need.

Mechanical Components					
Item No.	Part number	Description	Quantity	Unit price	Total
1	89155K27	Al Base plate 1/4"	1	$91.85	$91.85
2	2439T44	Wheel	2	$14.83	$29.66
3	2724T42	Casters	2	$7.96	$15.92
4	8674K165	Nylon Block	1	$37.29	$37.29
5	6546K21	1"x1"x36" tubing	2	$13.56	$27.12
6	13135A59	Bracket	24	$0.33	$7.92
7	8505K12	Acrylic 12" x 24" x 1/8" (Black)	3	$13.46	$40.38
8	89015K275	Motor Mount	1	$11.74	$11.74
9	3176T36	Vibration damper	2	$8.85	$17.70
11	91294A188	FlatHead Cap Screw 8mm (100)	1	$4.59	$4.59
12	92949A589	ButtonheadCapScrew 1 3/4" (10)	1	$4.95	$4.95
13	92095A227	ButtonheadCapScrew 14mm (25)	1	$9.95	$9.95
14	94500A226	ButtonheadCapScrew 8mm (100)	1	$7.42	$7.42
15	94895A030	Hex Nut (100)	1	$4.91	$4.91
Electrical components					
16		IR Sensor (20cm-150cm)	2	$15.95	$31.90
17		Arduino	1	$24.95	$24.95
18		12V 12Amp rechargable battery	1	$27.41	$27.41
19		breadboard	1	$5.00	$5.00
20		BTS7960 Motor Driver	2	$22.60	$45.20
Total					$400.66

Table 2: Bill of Materials

The quarter inch Aluminum plate ($91.85) was the most expensive part used in the project. This would be the first place to start reducing cost/weight of the end product, if Capstone was a two semester course. Different wheels and wider sourced fasteners and electrical components would also help in reducing cost. The goal would be to reduce the cost to less than $400 for the end user.

Recommendations

Throughout this paper, a general list has been developing of what to do next.

If a second prototype was slated for development, here are the main points for improvement discussed so far, organized around three main categories:

- *General*
  - Reduce cost
  - Reduce weight
  - Publish repair guides
- *Programming*
  - Post open source code
  - Work on the user interface
  - Streamline the code
- *Safety*
  - Round the edges
  - Add lights/beep noise to warn it is trapped or cornered
  - Add passive obstacle avoidance to avoid trip hazard

However, there are three additional recommendations this group would like to advance in regards to Kyle's development (**Figure 8**).

Figure 8: Additional recommendations

- *Lightweight Materials*

  There are many strong, lightweight materials on the market and in development. More research here would make this proposed robotic follower a more likely candidate for actual development. While it may end as a tradeoff between cost and function, this is definitely an area for further focus.

- *Additional Sensors*

  Several types and groupings of sensors were used in the Kyle prototype. However further exploration into other infrared, ultrasonic and optical devices, combinations and arrays, would definitely be recommended.

- *Incorporate User Feedback*

  Finally, this group would recommend letting a variety of users take home and interact with the device for week long sessions, gathering their feedback and sincerely applying that data into the next iteration.

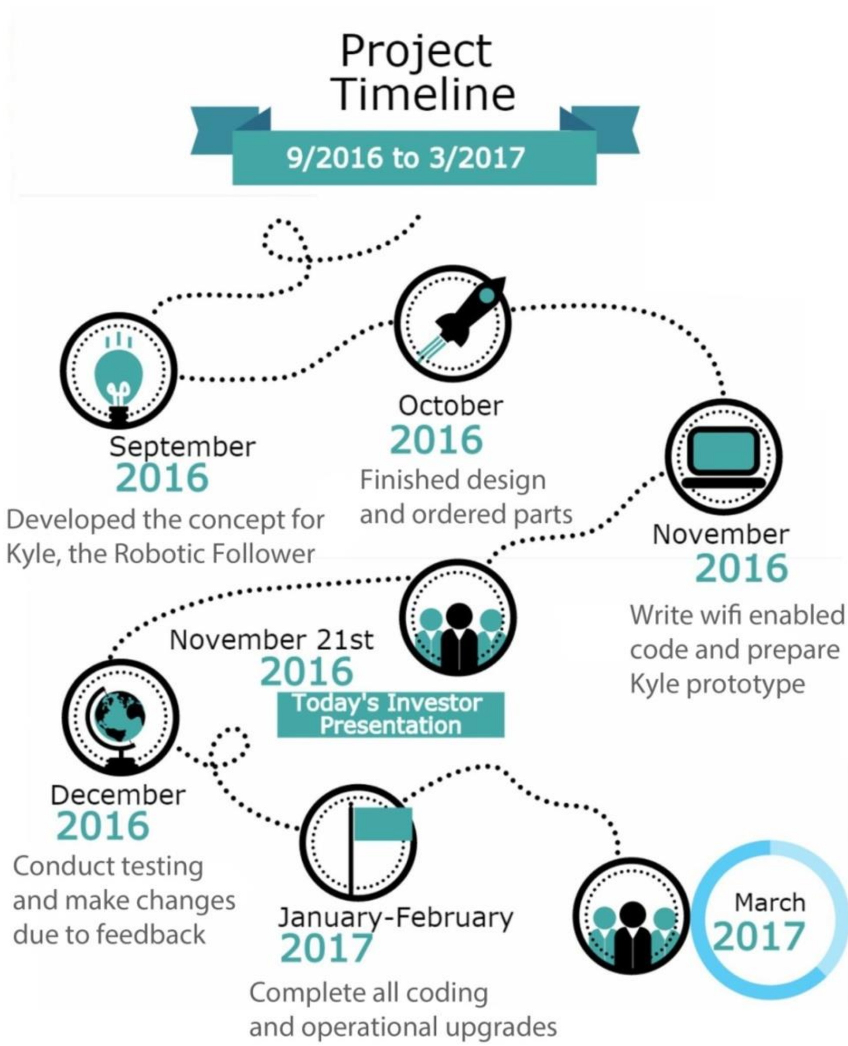


Figure 9: Proposed timeline, if Capstone were two semesters

In Figure 9 (above), the same graphic that was part of the "sales" presentation delivered on November 21st is listed. While this group won't be pursuing the next steps formally, the general feasibility of it is not discouraging. Kyle is a unique idea among the similar solutions on the market because it is modular. It may be programmed to move like a Roomba (especially if modified with specific cleaning attachments) or follow a target, and the option of attachments expand its capabilities and usefulness exponentially.

Acknowledgments

"Team For," made up by Timm Bischoff, Steven Krygier, Stephanie Legg and Kat Liegl, put a lot of effort into the conception and development of this robotic follower over the course of this semester, however this project would not have been possible without the help and advice from the computer engineering students in the computer lab on the first floor of the Kopp building; Anthony, Karl, Kyle and Wei Ye (a mechanical graduate student).

This team is also indebted to Dr. Don Dekker and Dr. Steven Sundarro for their guidance and weekly supervision; always ready to listen to concerns and counter them with resources and options.

We'd like to express sincere thanks for the support, resources and encouragement in completing this project, and therefore the last class towards graduation.

References

Sharp IR Sensor Data Sheet: (9 page user guide)

http://www.robotstorehk.com/gp2y0a02_e.PDF

Arduino: Source for reference and template sketches to operate the microcontroller

http://www.arduino .com

Motor

http://www.ebay.com/itm/Jazzy-Power-Wheelchair-Gearboxes-Motors-Large-Rc-Lawnmower-Robotics-Hubs-/222254572607

BTS7690 Motor Driver Operation: (28 page user guide)

http://wordpress.bonairetec.com/?p=75

BTS7960 Data Sheet:

http://www.robotpower.com/downloads/BTS7960_v1.1_2004-12-07.pdf

Actuators

https://www.firgelliauto.com/collections/linear-actuators

Appendices

□PNY PowerPack AD5200

Make the Most of Your Mobile Lifestyle.

PNY's PowerPack AD5200 offers a sleek and innovative solution in rechargeable batteries. Know when it's time to recharge with the convenient digital display, which numerically displays the PowerPack's battery level status. Integrated LED light allows the PowerPack to double as a flashlight for emergency situations. The portable, aluminum design is small enough to keep with you at all times, yet durable enough to withstand a busy lifestyle, so you can stay charged and stay in touch. Whether in your travel bag or your purse, the AD5200 is the perfect accessory for your mobile device. Sleek and innovative, **the AD5200 PowerPack delivers enough juice for up to 3 full-charges***. PNY's PowerPack AD5200 meets the needs of even the most demanding mobile lifestyle. Stay connected on the go with PNY!

The PowerPack comes with an included Micro-USB cable – which easily recharges the PowerPack and also supports charging via three universal USB ports to Android, Windows, and BlackBerry smartphones, tablets, and USB powered devices. The PowerPack also accepts an Apple charging cable, for charging your iPhone, iPod, or iPad.

With a PowerPack from PNY, you can take power on the go and keep your device charged at all times – no matter where you are.

*Varies by device

□□□Specifications

Capacity	5200mAh
Input	2 Amp (Micro-USB)
Outputs	2.4 Amp
Battery Level Indicator	LED
Cables Included	Micro-USB Cable
Number of Charges	Up To 3x Charge*
Works With	Android, Apple, Blackberry, Windows
Color	Silver

- See more at: https://www.pny.com/PowerPackAD5200?sku=P-B-5200-24-S03-RB#sthash.pRsaH7sr.dpuf

Appendix 1: Battery pack specifications

SHARP GP2Y0A710K0F

GP2Y0A710K0F

Distance Measuring Sensor Unit
Measuring distance: 100 to 550 cm
Analog output type

■Description

GP2Y0A710K0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.
The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.
This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Long distance type
   Distance measuring range : 100 to 550 cm
2. Analog output type
3. Package size : 58×17.6×22.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

■Agency approvals/Compliance

1. Compliant with RoHS directive(2002/95/EC)

■Applications

1. Projector (for auto focus)
2. Robot cleaner
3. Auto-switch for illumination, etc.
4. Human body detector
5. Amusement equipment
   (Robot, Arcade game machine)

■Absolute Maximum Ratings

(T_a=25℃, V_{CC}=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to V_{CC}+0.3	V
Operating temperature	T_{opr}	-10 to +60	℃
Storage temperature	T_{stg}	-40 to +70	℃

■Electro-optical Characteristics

(T_a=25℃, V_{CC}=5V)

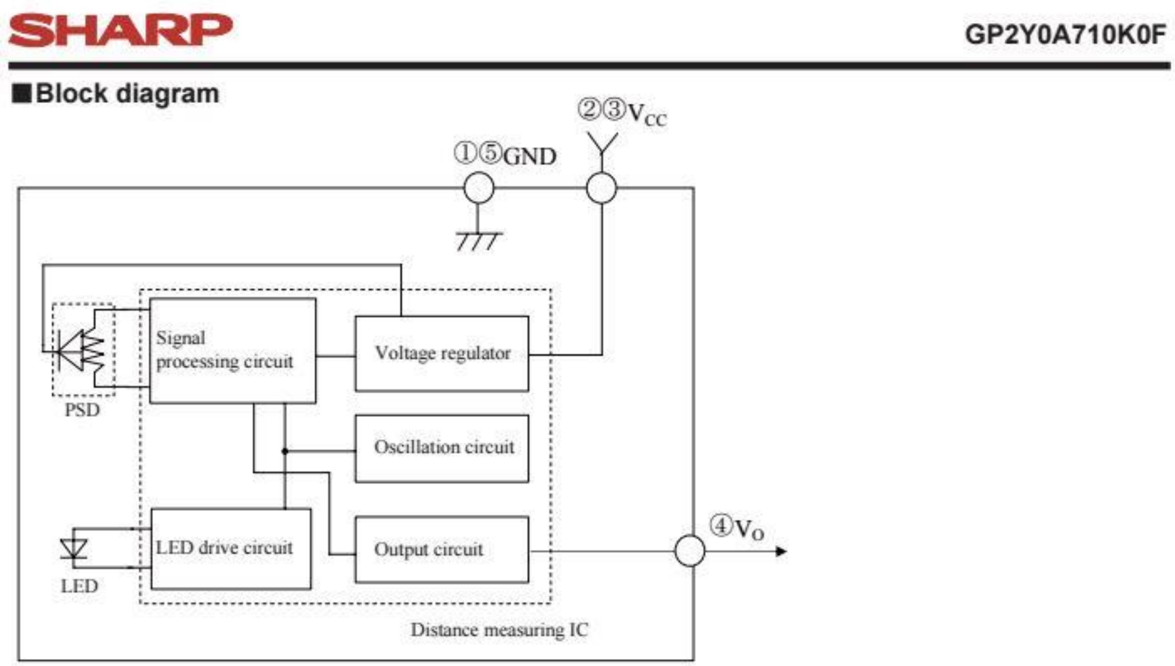
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I_{CC}	L=150cm (Note 1)	—	30	50	mA
Distance measuring	ΔL	(Note 1)	100	—	550	cm
Output voltage	V_O	L=100cm (Note 1)	2.3	2.5	2.7	V
Output voltage differential	ΔV_{O1}	Output voltage difference between L=100cm and L=200cm (Note 1)	0.5	0.7	0.9	V
	ΔV_{O2}	Output voltage difference (L=100cm→200cm)/ Output voltage differentce (L=200cm→550cm)(Note 1,2)	1.25	1.55	1.85	V

* L : Distance to reflective object
Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27・white face, reflectance; 90%)
Note 2 : The value at 550 cm is the average of 20 times distance measuring.

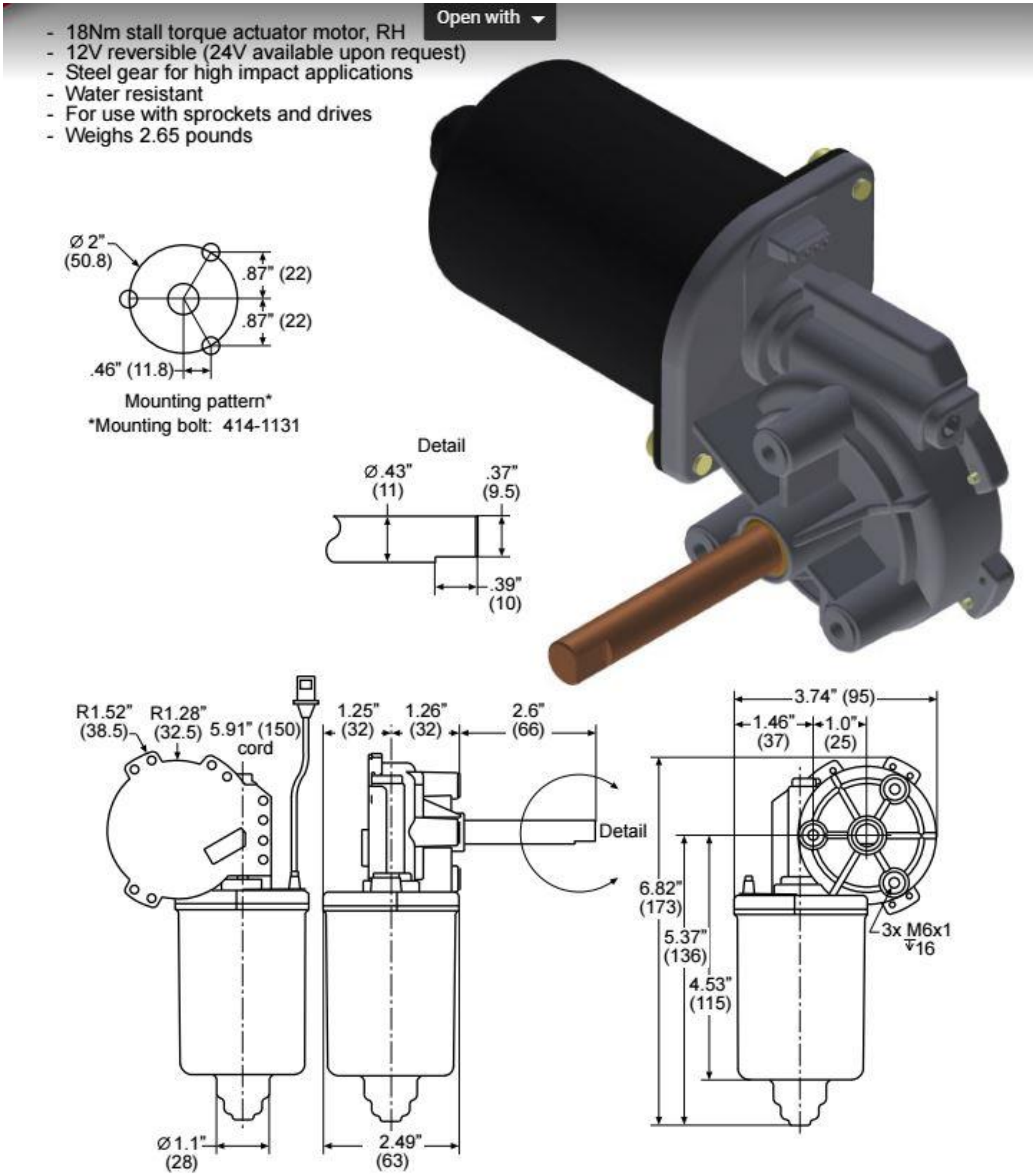
■Recommended operating conditions

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{CC}		4.5 to 5.5	V

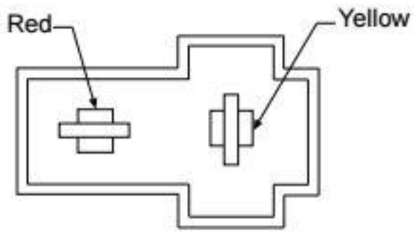
Appendix 2 : IR Sensor specifications



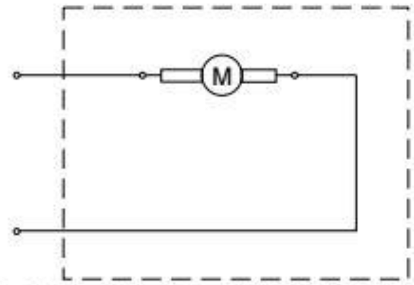
Appendix 3: IR Sensor Block Diagram



Appendix 4: 218-2003 Motor



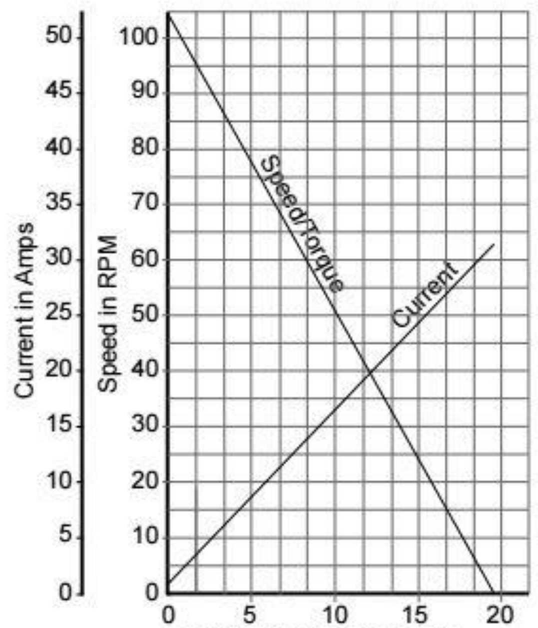
Terminal housing: 317-1057
Terminal: 317-1054
Mate terminal housing: 317-1056
Mate terminal: 317-1055



Red (+), yellow (-) = CW preferred rotation
Yellow (+), red (-) = CCW

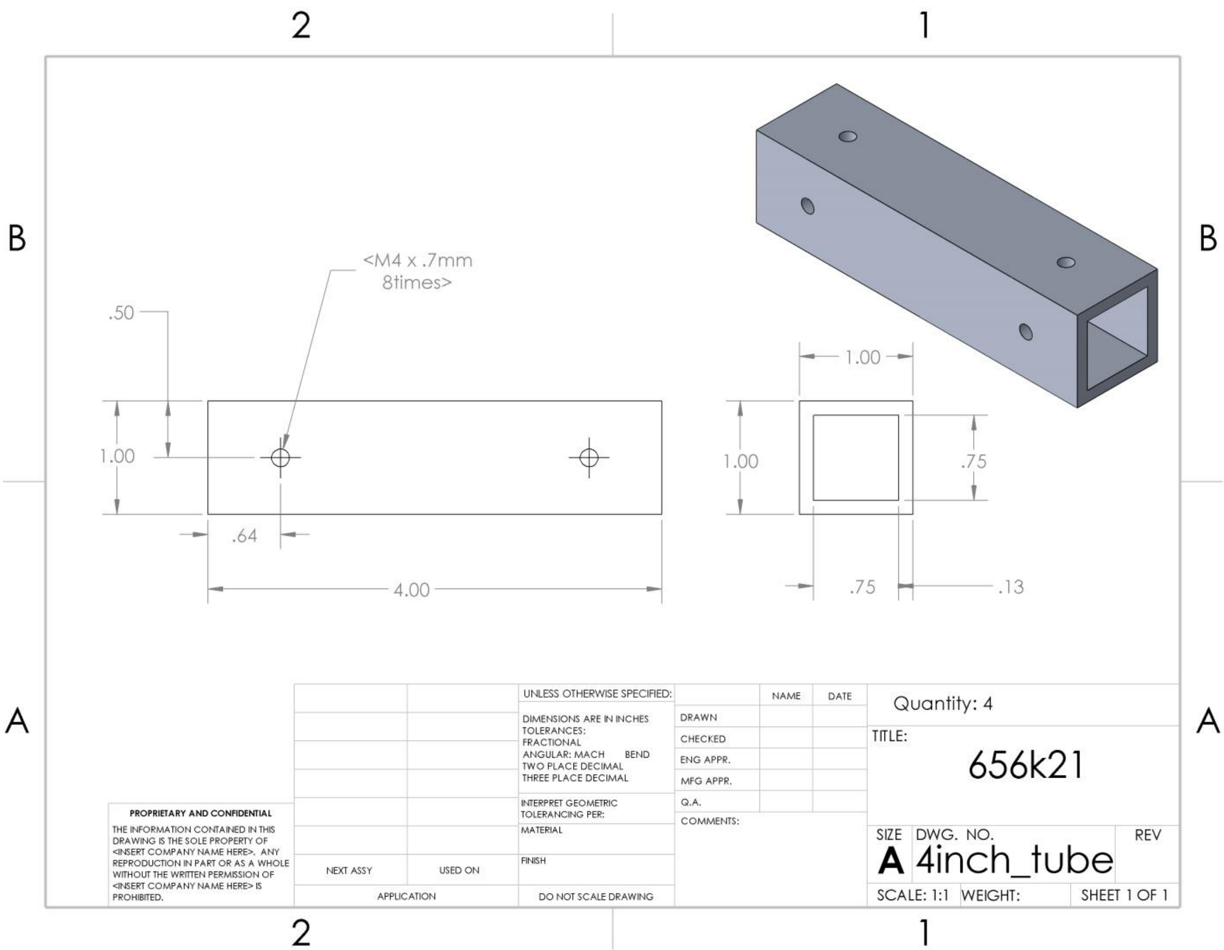
Clockwise Motor Shaft Rotation		
Data Point	Data Type	Value Range
No Load	Current (A)	1.7 - 1.4
	Speed (rpm)	116.2 - 95.1
Stall Load	Torque (Nm)	21.3 - 17.5
	Current (A)	36.0 - 29.4
Peak Power	Power (W)	60.0 - 49.1
	Torque (Nm)	11.1 - 9.1
Nominal (Peak Efficiency)	Power (W)	36.5 nominal
	Speed (rpm)	82.1 nominal
	Current (A)	7.0 nominal
	Torque (Nm)	4.3 nominal

Counter-Clockwise Motor Shaft Rotation		
Data Point	Data Type	Value Range
No Load	Current (A)	1.6 - 1.3
	Speed (rpm)	113.5 - 92.9
Stall Load	Torque (Nm)	20.6 - 16.9
	Current (A)	35.7 - 29.2
Peak Power	Power (W)	55.5 - 45.4
	Torque (Nm)	10.5 - 8.6
Nominal (Peak Efficiency)	Power (W)	34.3 nominal
	Speed (rpm)	79.6 nominal
	Current (A)	7.4 nominal
	Torque (Nm)	4.2 nominal

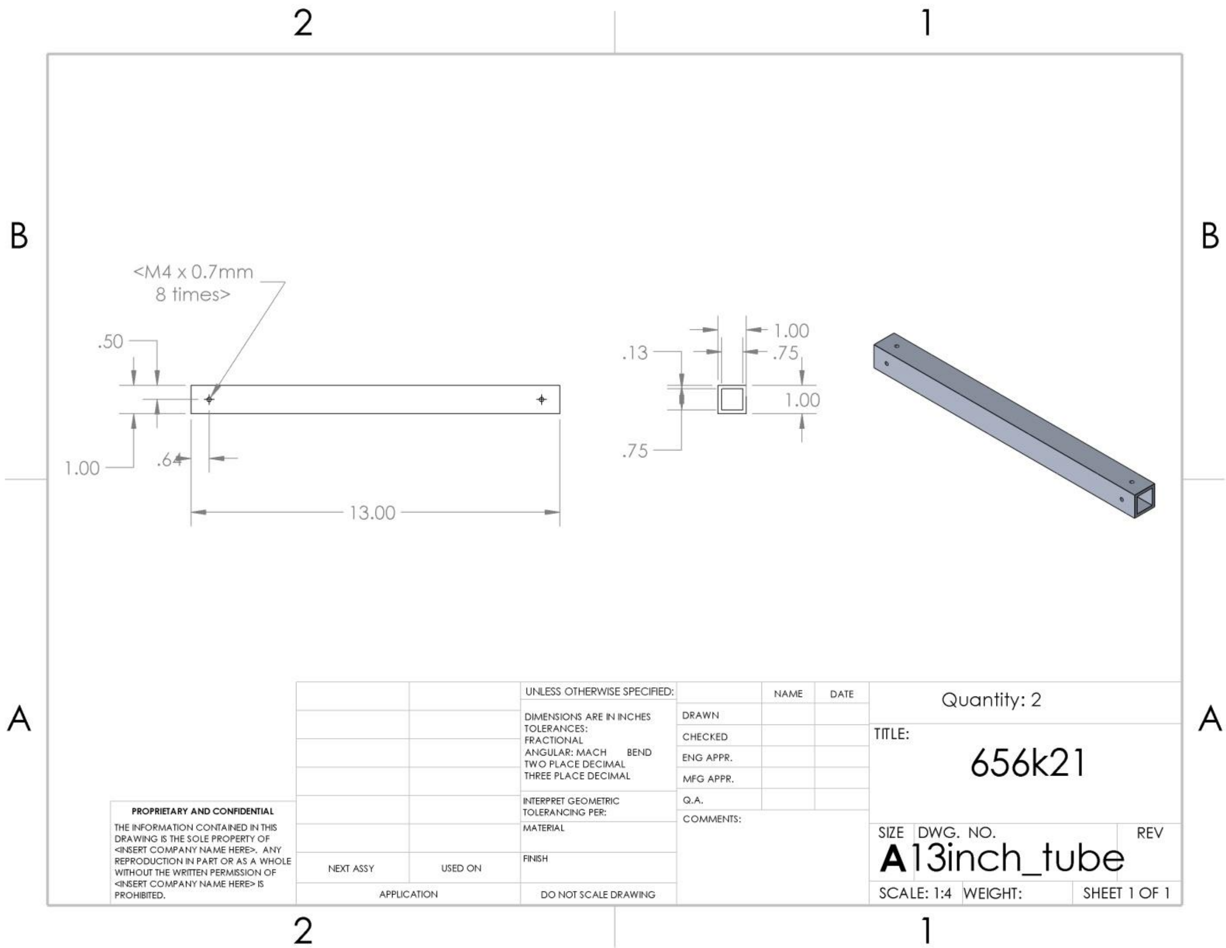


Appendix 5: 218-2003 Motor Specifications

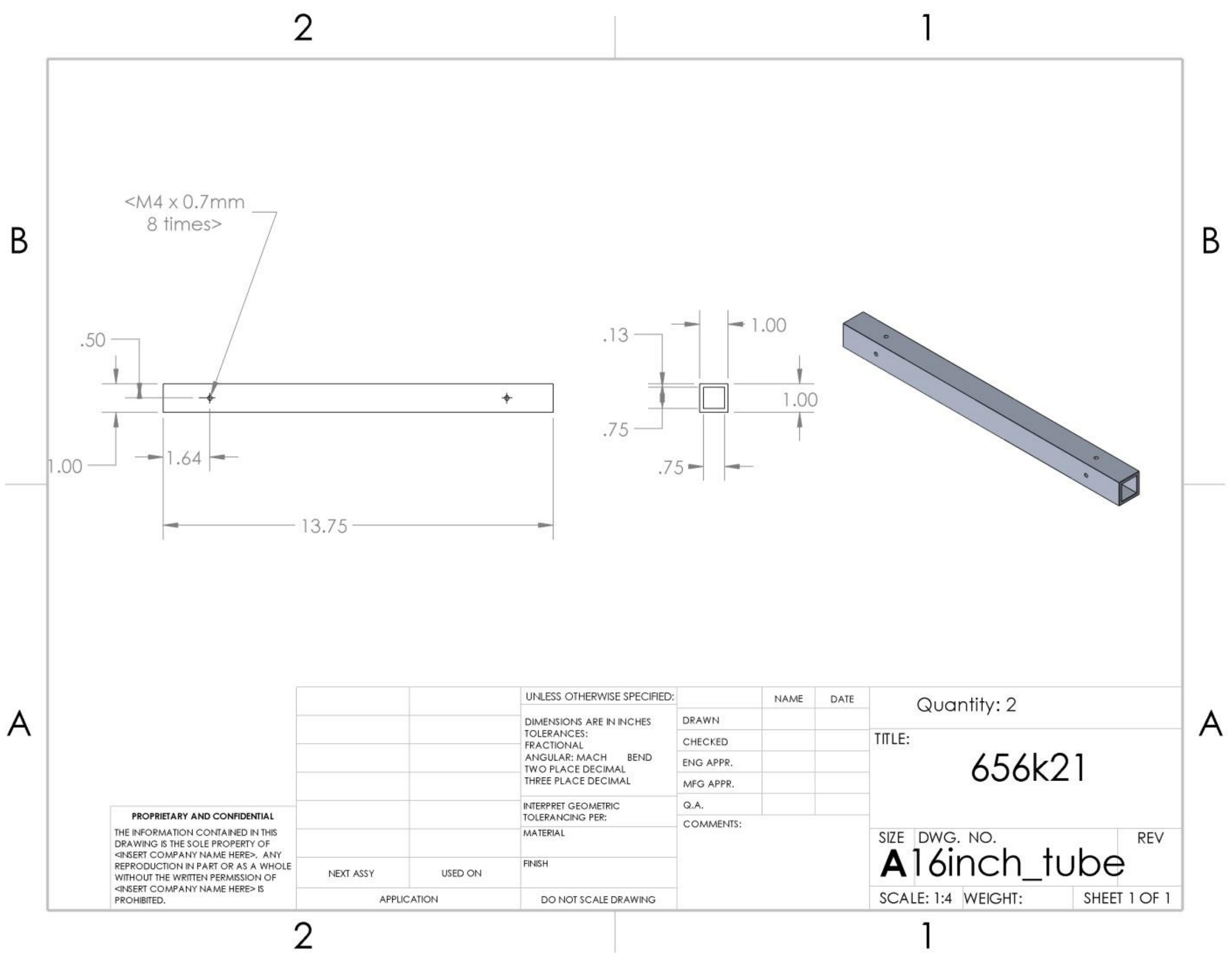
Drawings



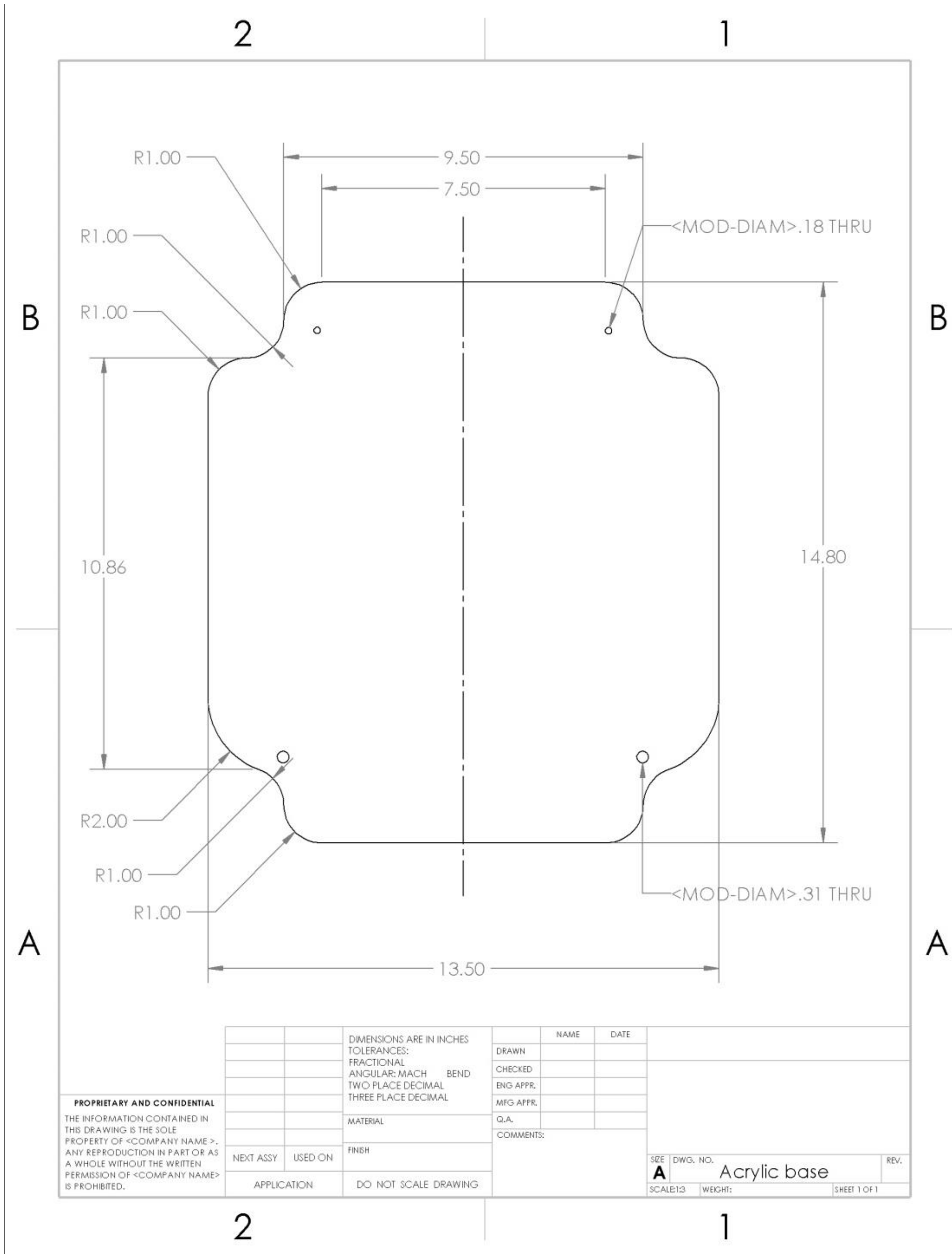
CAD Drawings 1: 4 inch tube



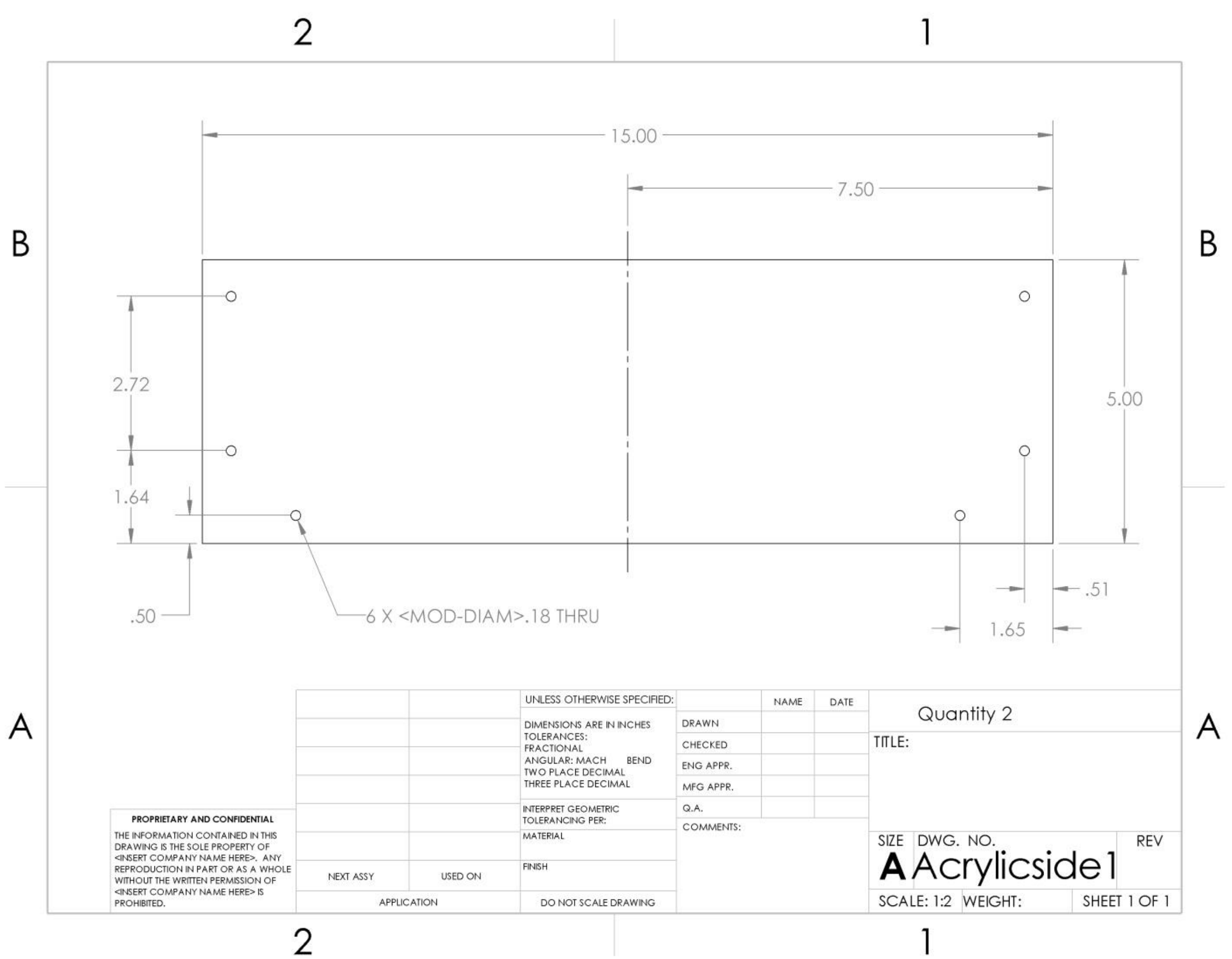
CAD Drawings 2: 13 inch tube



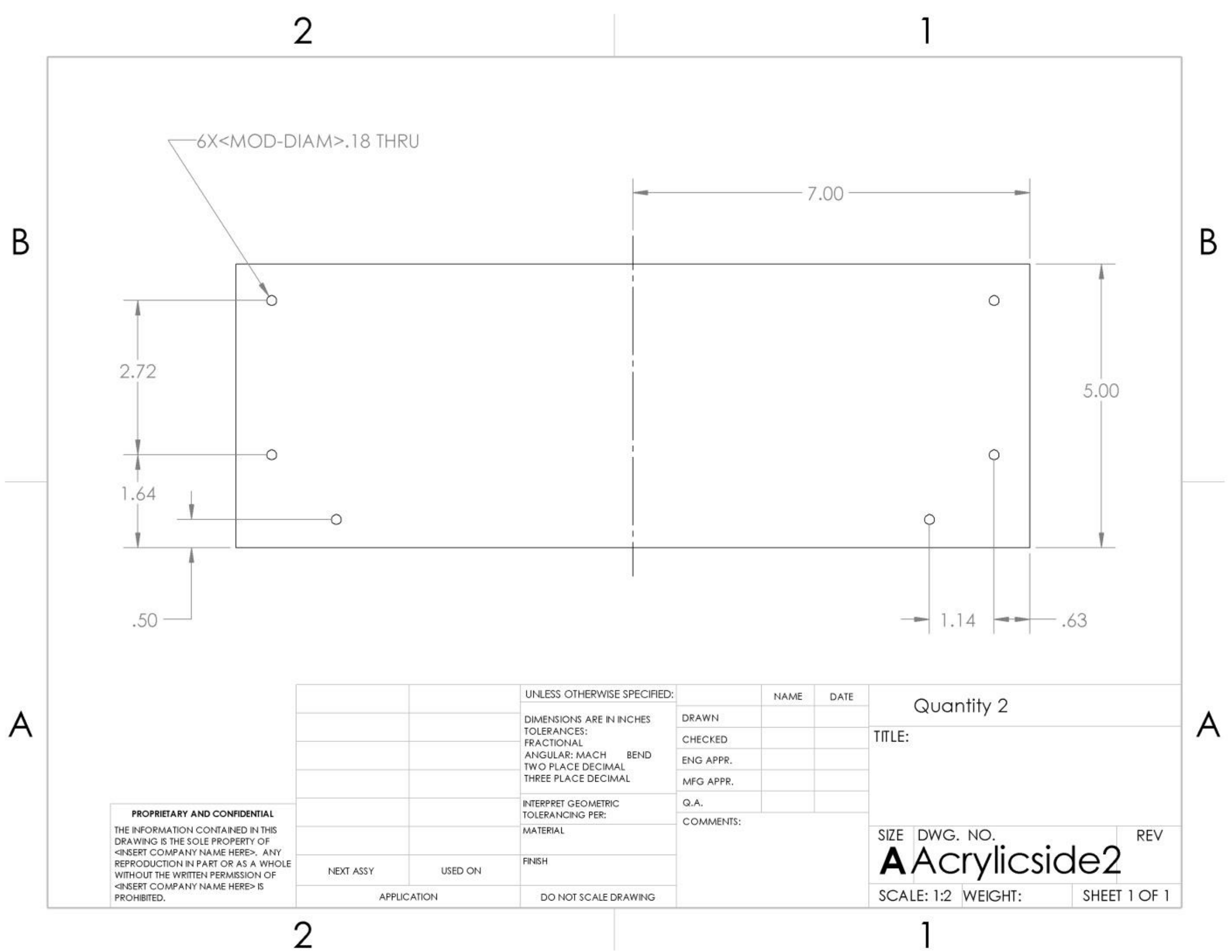
CAD Drawings 3: 16 inch tube



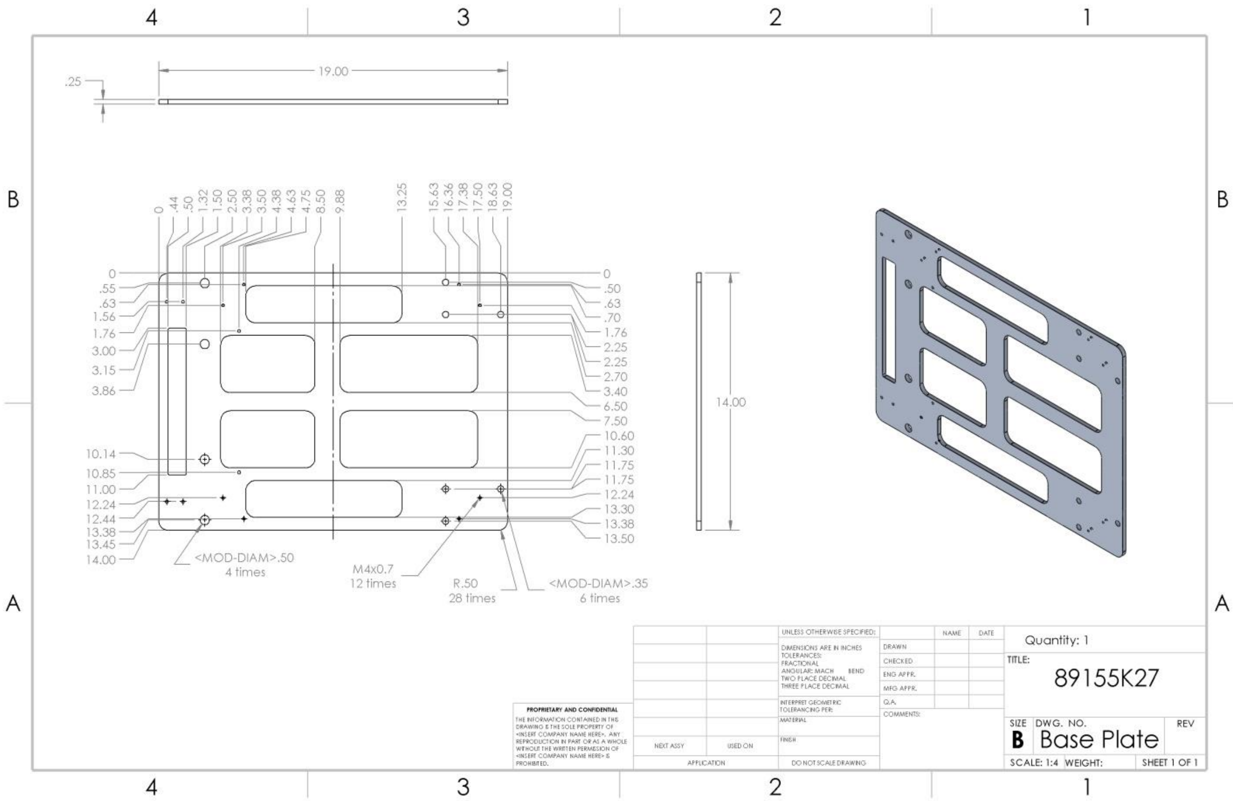
CAD Drawings 4: Acrylic Base



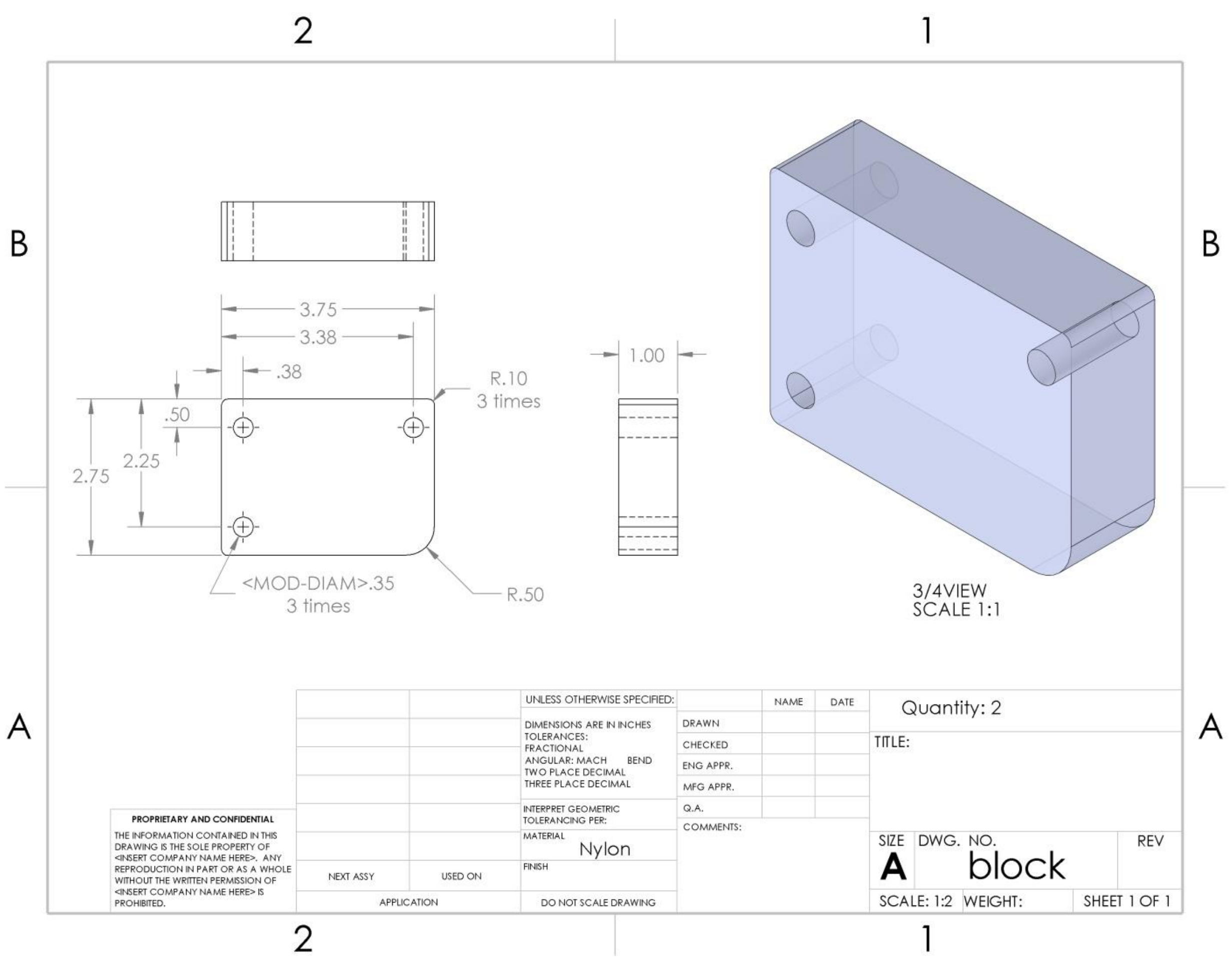
CAD Drawings 5: Acrylic Side 1



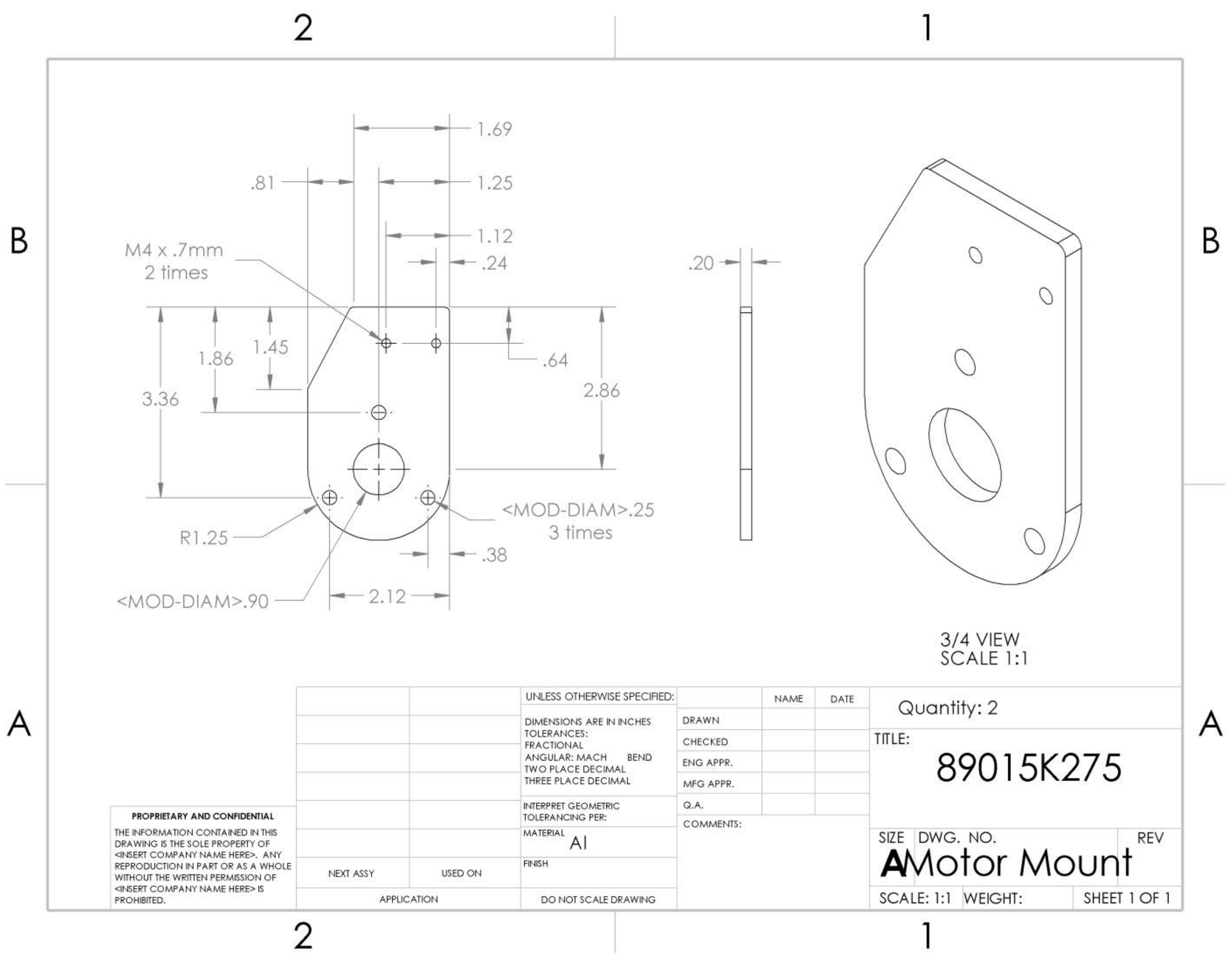
CAD Drawings 6: Acrylic Side 2



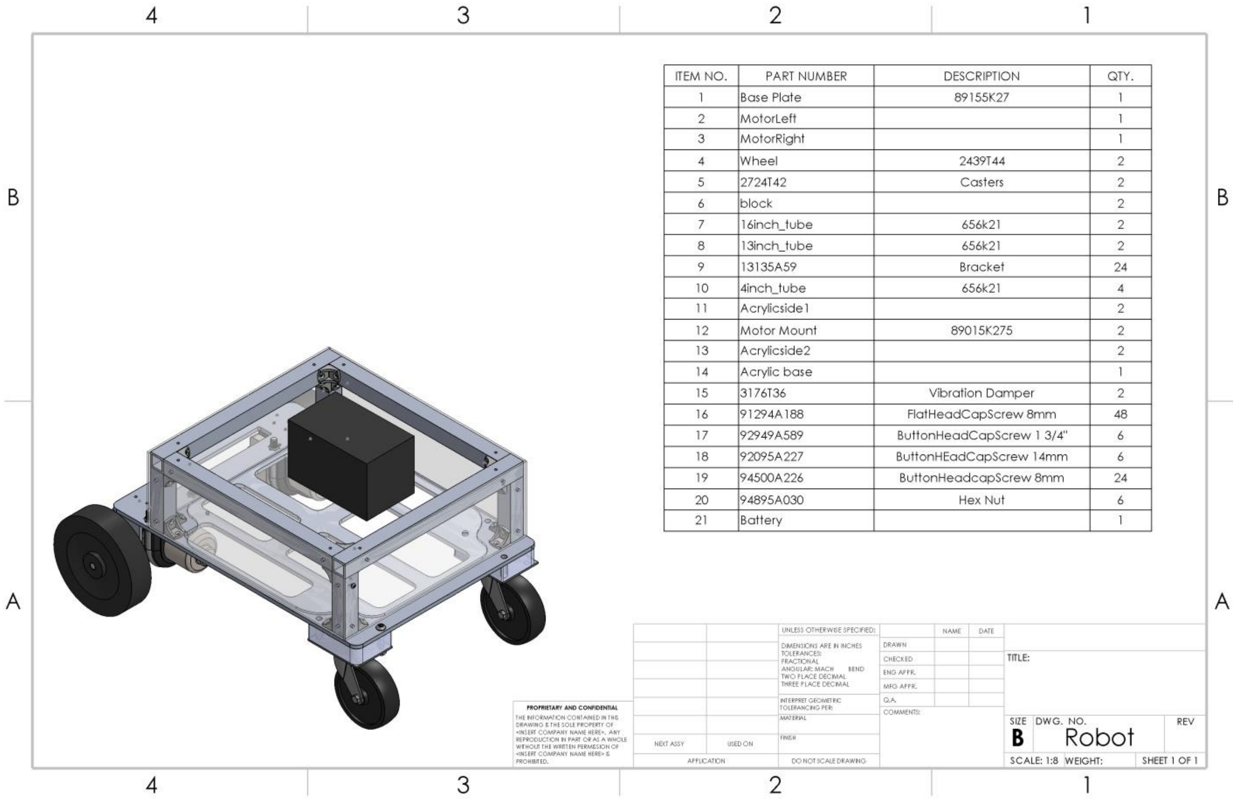
CAD Drawings 7: ¼" Aluminum Base Plate



CAD Drawings 8: Block



CAD Drawings 9: Motor Mount



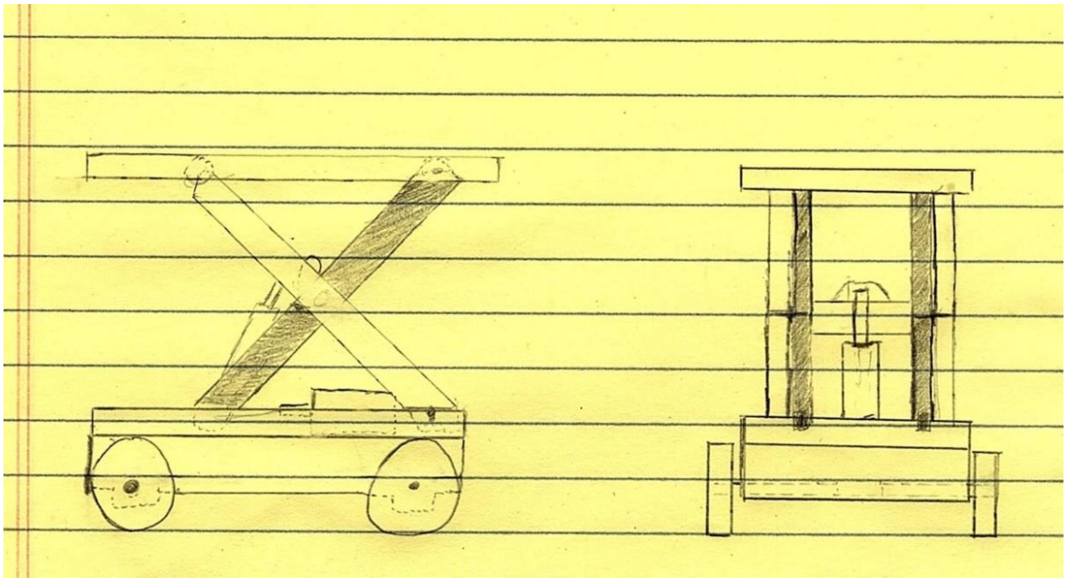
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Base Plate	89155K27	1
2	MotorLeft		1
3	MotorRight		1
4	Wheel	2439T44	2
5	2724T42	Casters	2
6	block		2
7	16inch_tube	656k21	2
8	13inch_tube	656k21	2
9	13135A59	Bracket	24
10	4inch_tube	656k21	4
11	Acrylicside1		2
12	Motor Mount	89015K275	2
13	Acrylicside2		2
14	Acrylic base		1
15	3176T36	Vibration Damper	2
16	91294A188	FlatHeadCapScrew 8mm	48
17	92949A589	ButtonHeadCapScrew 1 3/4"	6
18	92095A227	ButtonHEadCapScrew 14mm	6
19	94500A226	ButtonHeadcapScrew 8mm	24
20	94895A030	Hex Nut	6
21	Battery		1

CAD Drawings 10: Assembled Robot (without electrical components)

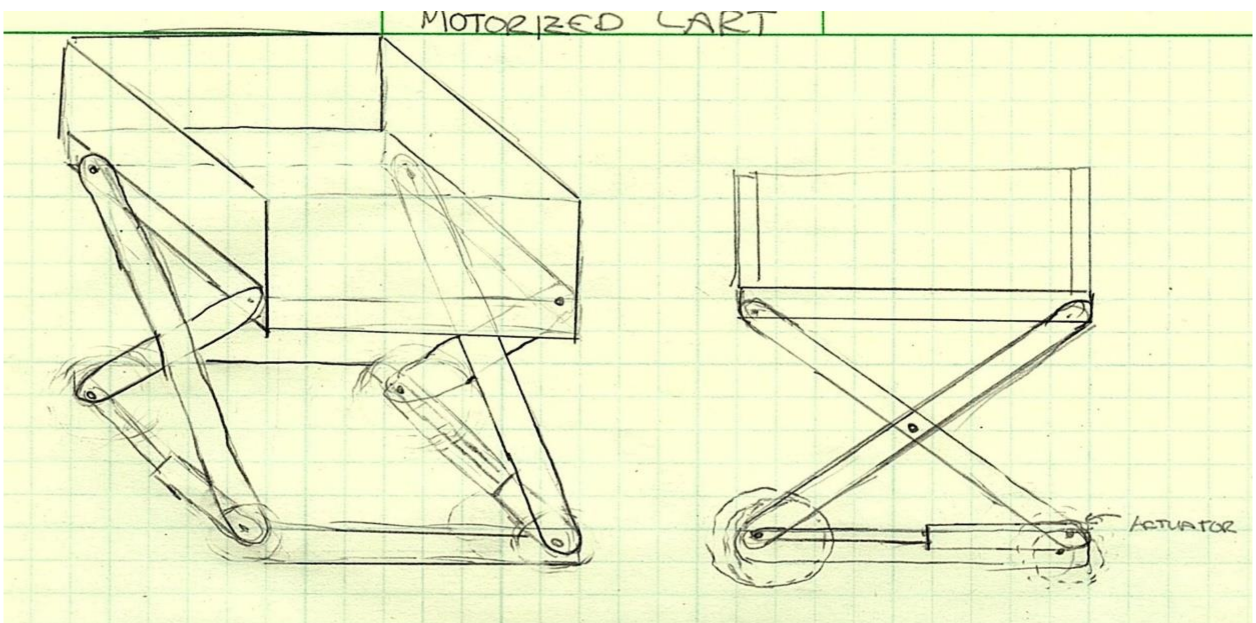
Bill of Materials

Mechanical Components					
Item No.	Part number	Description	Quantity	Unit price	Total
1	89155K27	Al Base plate 1/4"	1	$91.85	$91.85
2	2439T44	Wheel	2	$14.83	$29.66
3	2724T42	Casters	2	$7.96	$15.92
4	8674K165	Nylon Block	1	$37.29	$37.29
5	6546K21	1"x1"x36" tubing	2	$13.56	$27.12
6	13135A59	Bracket	24	$0.33	$7.92
7	8505K12	Acrylic 12" x 24" x 1/8" (Black)	3	$13.46	$40.38
8	89015K275	Motor Mount	1	$11.74	$11.74
9	3176T36	Vibration damper	2	$8.85	$17.70
11	91294A188	FlatHead Cap Screw 8mm (100)	1	$4.59	$4.59
12	92949A589	ButtonheadCapScrew 1 3/4" (10)	1	$4.95	$4.95
13	92095A227	ButtonheadCapScrew 14mm (25)	1	$9.95	$9.95
14	94500A226	ButtonheadCapScrew 8mm (100)	1	$7.42	$7.42
15	94895A030	Hex Nut (100)	1	$4.91	$4.91
Electrical components					
16		IR Sensor (20cm-150cm)	2	$15.95	$31.90
17		Arduino	1	$24.95	$24.95
18		12V 12Amp rechargable battery	1	$27.41	$27.41
19		breadboard	1	$5.00	$5.00
20		BTS7960 Motor Driver	2	$22.60	$45.20
Total					$400.66

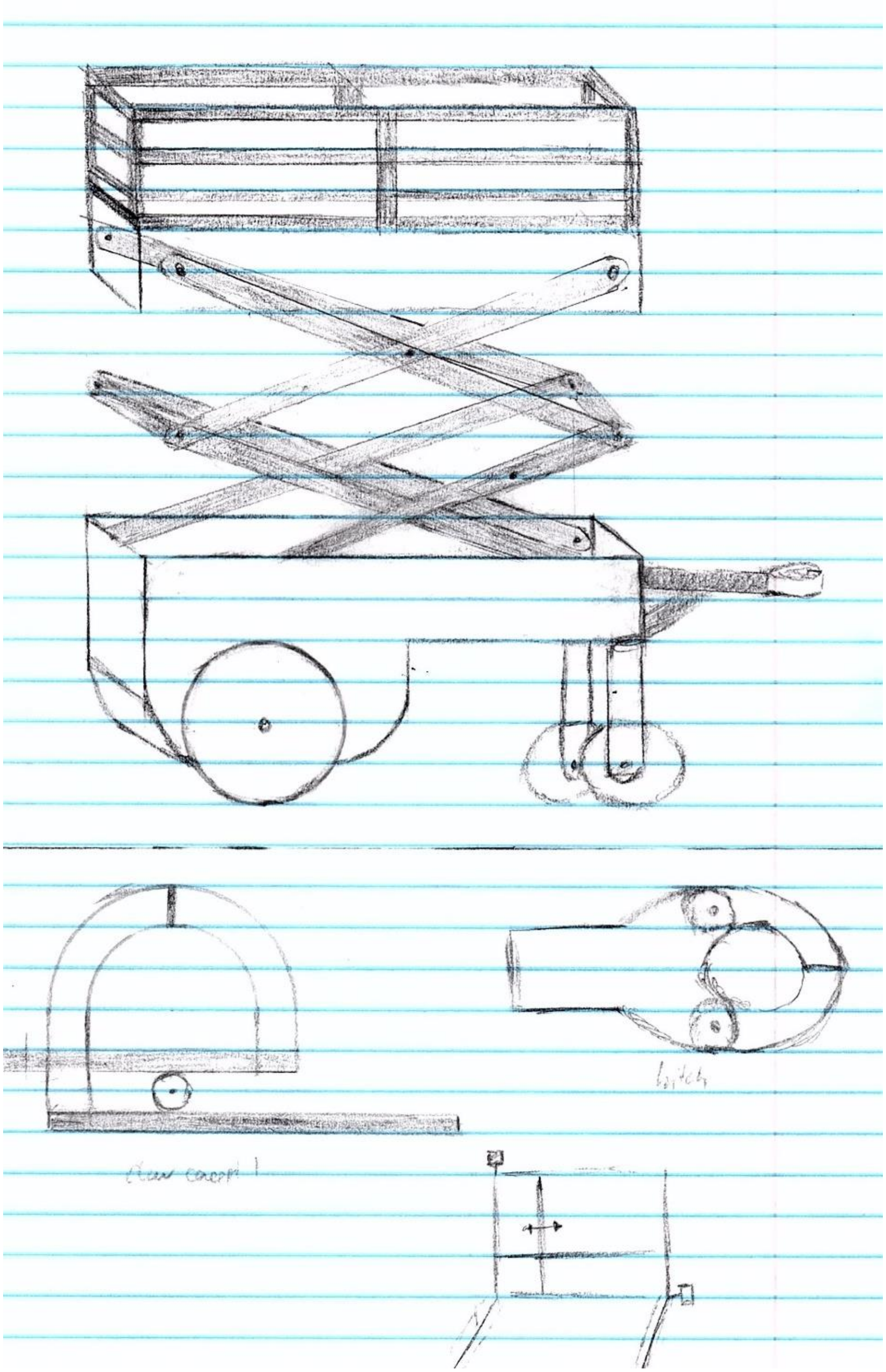
Alternatives



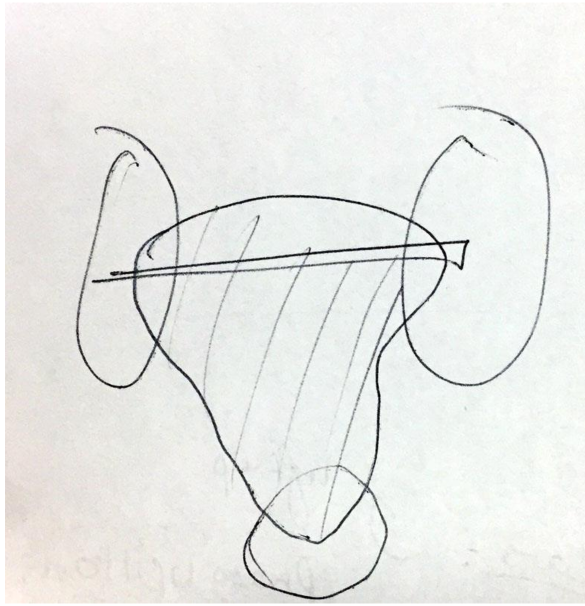
Alternatives 1: Motorized scissor lift, with folding table



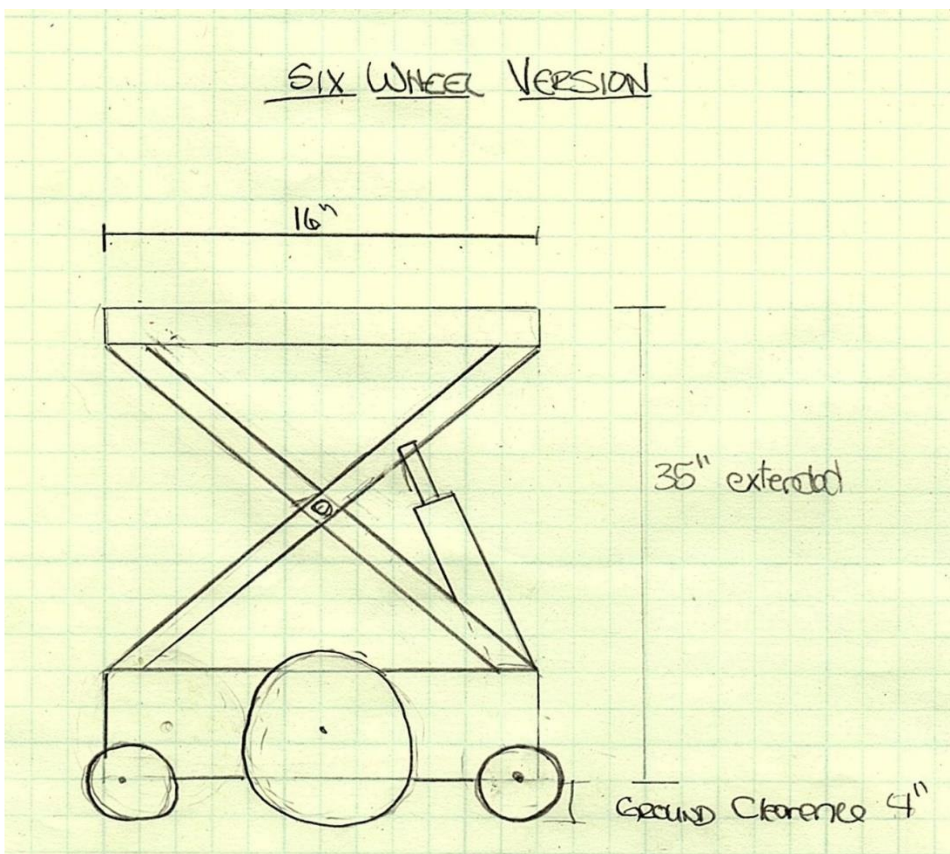
Alternatives 2: Motorized scissor lift, basket, low to the ground



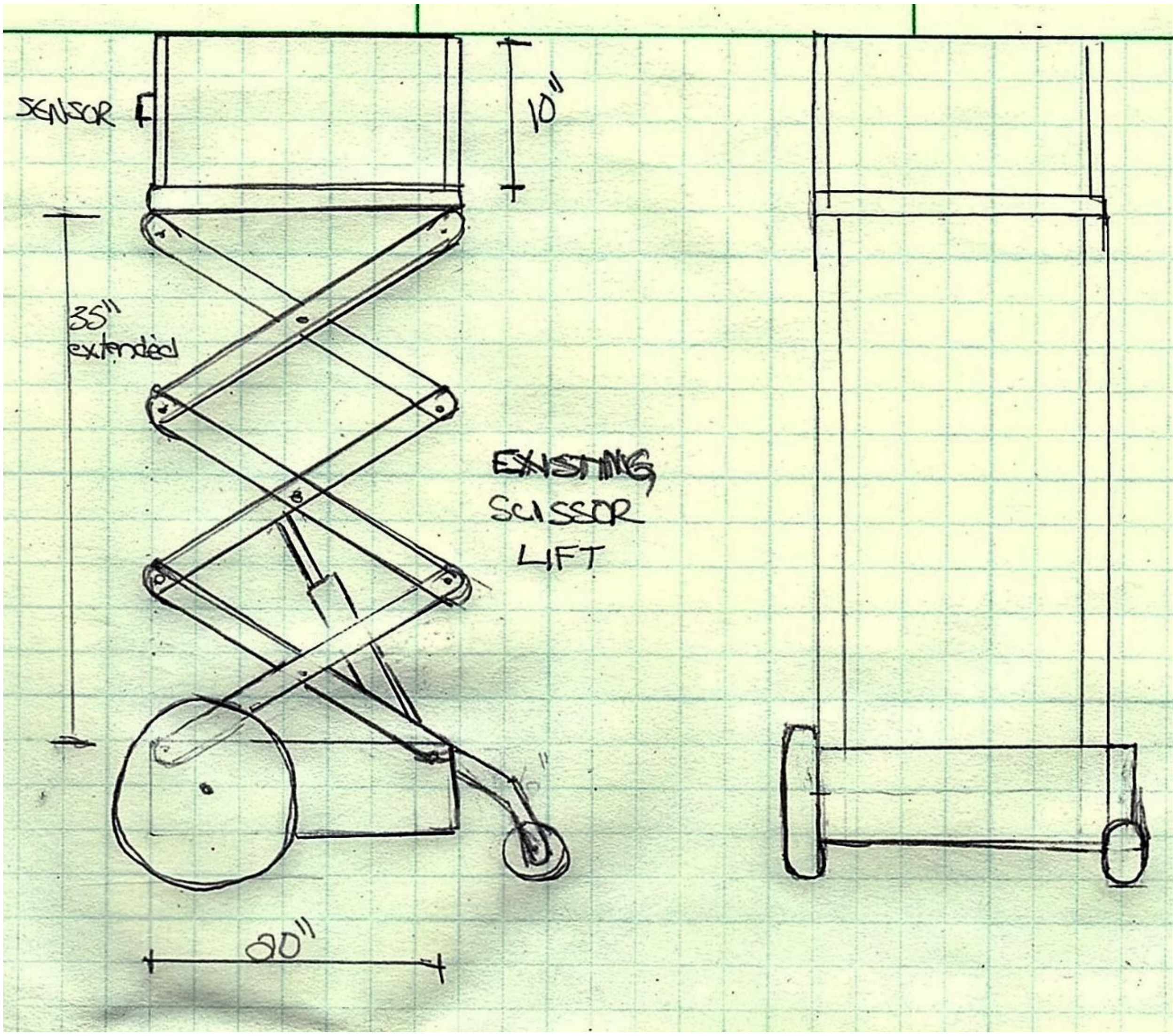
Alternatives 3: Motorized cart with lift and basket, and hitch



Alternatives 4: 3 Wheel base



Alternatives 5: 6 Wheel base



Alternatives 6: Motorized cart with former Capstone group's scissor lift

Decision Matrix

Criteria\Alternatives	**Weight (%)**	**Target (1-10)**	**Final Kyle Design (1-10)**
Safe, Reliable, Accurate	20	7	7
Human factors, Pro Social, Useful	20	7	7
Low weight, Efficient, Battery life	20	7	6
Affordable, Modular, Available	15	7	8
Simple, Clean lines, Elegant	15	7	8
Sustainable, Long life cycle, Easy to repair	10	7	8
Total	**100**	**70**	**72**

Table 3: Alternatives Criteria Analysis: 10 is the best choice

While all the ideas included, and a few others were considered, the current Kyle prototype design was the only one who met or exceeded 70% on each of the targets from the Parameters and Specifications chart, Figure 1.

Calculations

Rolling Resistance

C_{rr} = Coefficient of Rolling Resistance

N = Normal Force

F_{rr} = Rolling Resistance Force

$F_{rr} = C_{rr} * N$

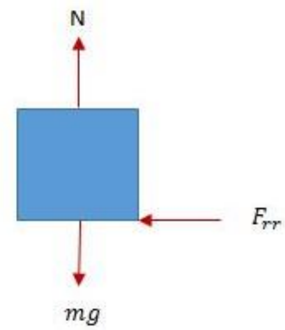
$F_{rr} = 0.003 * (36.24kg) * (9.81\frac{m}{s^2})$

$\boldsymbol{F_{rr} = 1.068\,N}$

$T = F_{rr} * d$

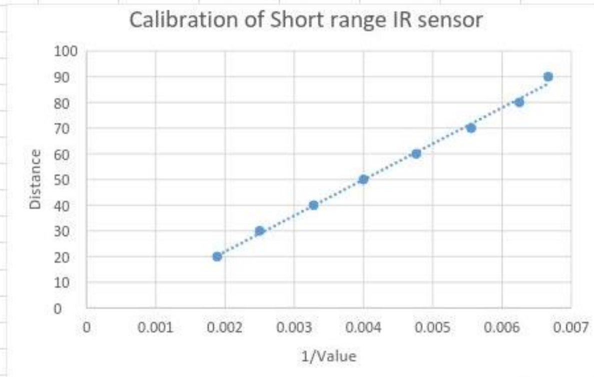
$T = 1.068N_{\square} * 0.1524m$

T = 0.1627 N*m



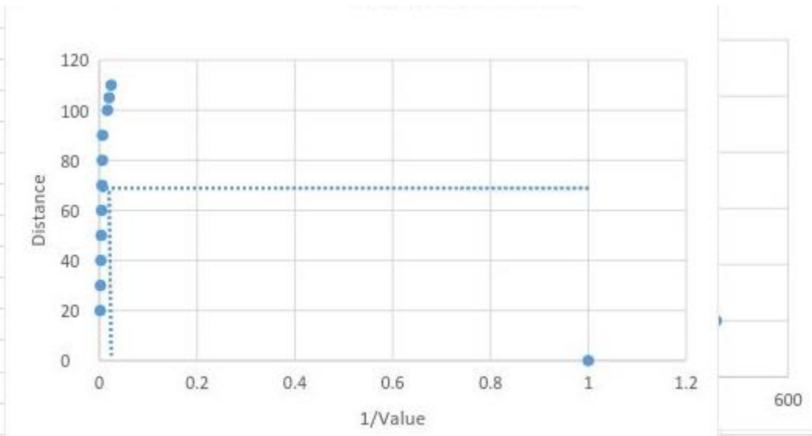
Calculation 1: Rolling Resistance

1/V	V	Distance(cm)
0.00188679	530	20
0.0025	400	30
0.00327868	305	40
0.004	250	50
0.00476190	210	60
0.00555555	180	70
0.00625	160	80
0.00666666	150	90
	60	
	50	
	41	



Calculation 2: Calibration of the Short Range IR sensor

./v	Voltage	Distance
.00188679	530	20
0.0025	400	30
.00327868	305	40
0.004	250	50
.00476190	210	60
.00555555	180	70
0.00625	160	80
.00666666	150	90
.01666666	60	100
0.02	50	105
.02439024	41	110



Calculation 3: Calibration of the Long Range Sensor

Auxiliary

File Edit Sketch Tools Help

kartcode

```
#define IR1 A0
#define IR2 A1
#define Dir1L 7
#define Dir1R 8
#define pwm1L 6
#define pwm1R 5
#define Dir2L 2
#define Dir2R 4
#define pwm2L 9
#define pwm2R 3


void setup(){
  Serial.begin(9600);

  pinMode(IR1,INPUT);
  pinMode(IR2,INPUT);
  pinMode(Dir1R,OUTPUT);
  pinMode(Dir1L,OUTPUT);
  pinMode(Dir2L,OUTPUT);
  pinMode(Dir2R,OUTPUT);
  pinMode(pwm1L,OUTPUT);
  pinMode(pwm1R,OUTPUT);
  pinMode(pwm2L,OUTPUT);
  pinMode(pwm2R,OUTPUT);
}

void loop(){
  float IR1S;
  float IR2S;
  float IR1D;
  float IR2D;
  float IR1V;
  float IR2V;
  int Rup = 45;
  int Rlow = 35;
  int Rstop = 30;
  int fast = 255;
  int norm = 200;
  int slow = 100;

  IR1S = analogRead(IR1);
  IR2S = analogRead(IR2);
  IR1V = 5*IR1S/1024;
  IR2V = 5*IR2S/1024;
  IR1D = 13999*(1/IR1S)-6.0711;
  IR2D = 1.2789*(IR2V)-.5787;
  Serial.println("Volatge 1");
  Serial.println(IR1V);
  Serial.println("Distance 1");
  Serial.println(IR1D);
  Serial.println("Voltage 2");
  Serial.println(IR2V);
  Serial.println("Distance 2");
  Serial.println(IR2D);

  digitalWrite(Dir1L, HIGH);
  digitalWrite(Dir1R, HIGH);
  digitalWrite(Dir2L, HIGH);
  digitalWrite(Dir2R, HIGH);


//IR 1 Motor

  if  (IR1D>Rup) {
   Serial.println("speed up 1");
   analogWrite(pwm1R, fast);
   analogWrite(pwm1L, 255-fast);
  }

  if (IR1D <= Rstop) {
   Serial.println("stop 1");
   analogWrite(pwm1R, 0);
   analogWrite(pwm1L, 0);
  }

  if ( (IR1D >= Rlow) && (IR1D <= Rup) ) {
   Serial.println("norm 1");
   analogWrite(pwm1R, norm);
   analogWrite(pwm1L, 255-norm);
  }

  if ( (IR1D > Rstop) && (IR1D < Rlow) ) {
   Serial.println("slow 1");
   analogWrite(pwm1R, 255-slow);
   analogWrite(pwm1L, slow);
  }

//IR 2 Motor

  if  (IR2D>Rup) {
   Serial.println("speed up 2");
   analogWrite(pwm2R, fast);
   analogWrite(pwm2L, 255-fast);
  }

  if (IR2D <= Rstop) {
   Serial.println("stop 2");
   analogWrite(pwm2R, 0);
   analogWrite(pwm2L, 0);
  }

  if ( (IR2D >= Rlow) && (IR2D <= Rup) ) {
   Serial.println("norm 2");
   analogWrite(pwm2R, norm);
   analogWrite(pwm2L, 255-norm);
  }

  if ( (IR2D > Rstop) && (IR2D < Rlow) ) {
   Serial.println("slow 2");
   analogWrite(pwm2R, 255-slow);
   analogWrite(pwm2L, slow);
  }



  delay(200);
}
```

Figure 10: Code for the follower

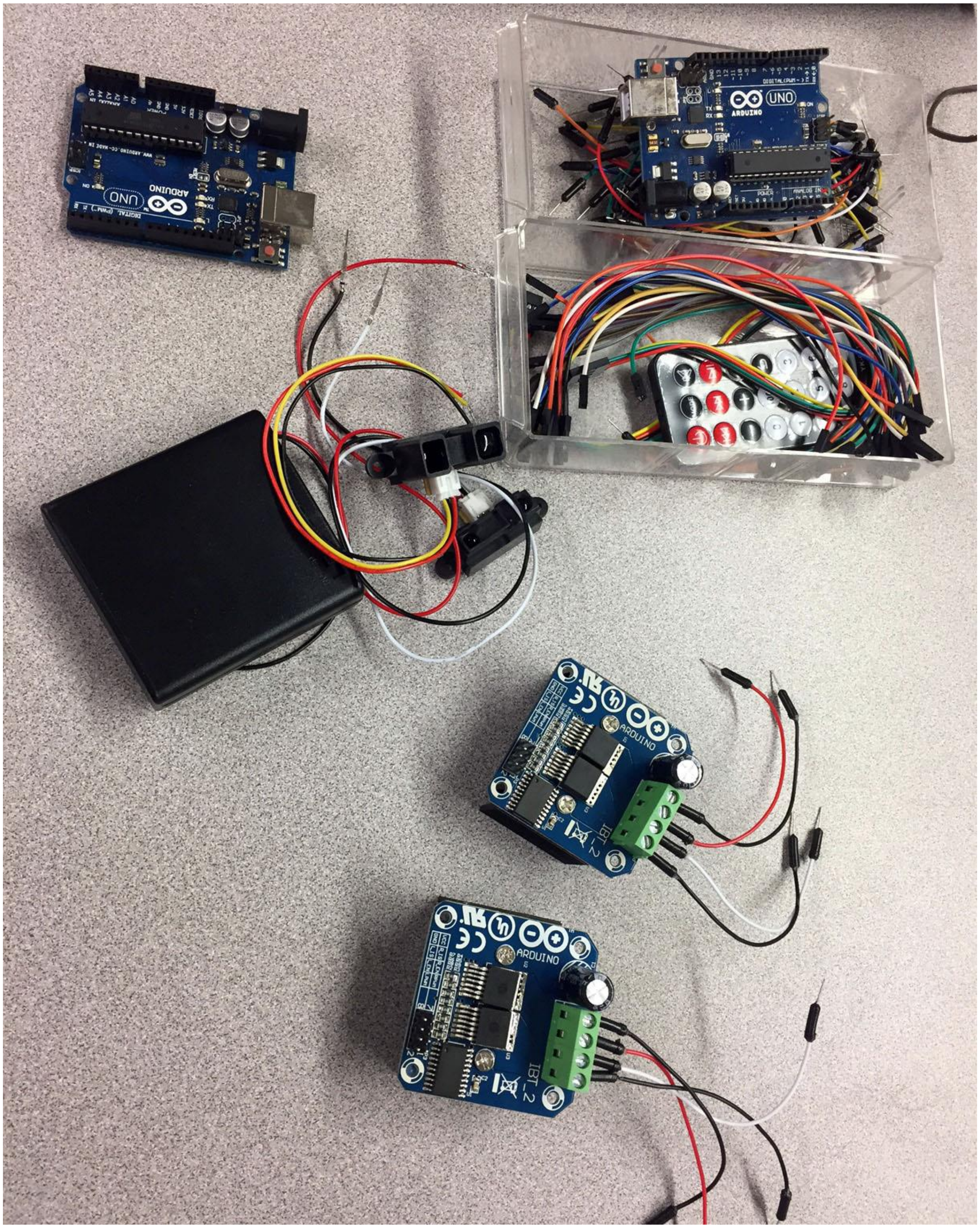


Figure 11: Project Assembly 1

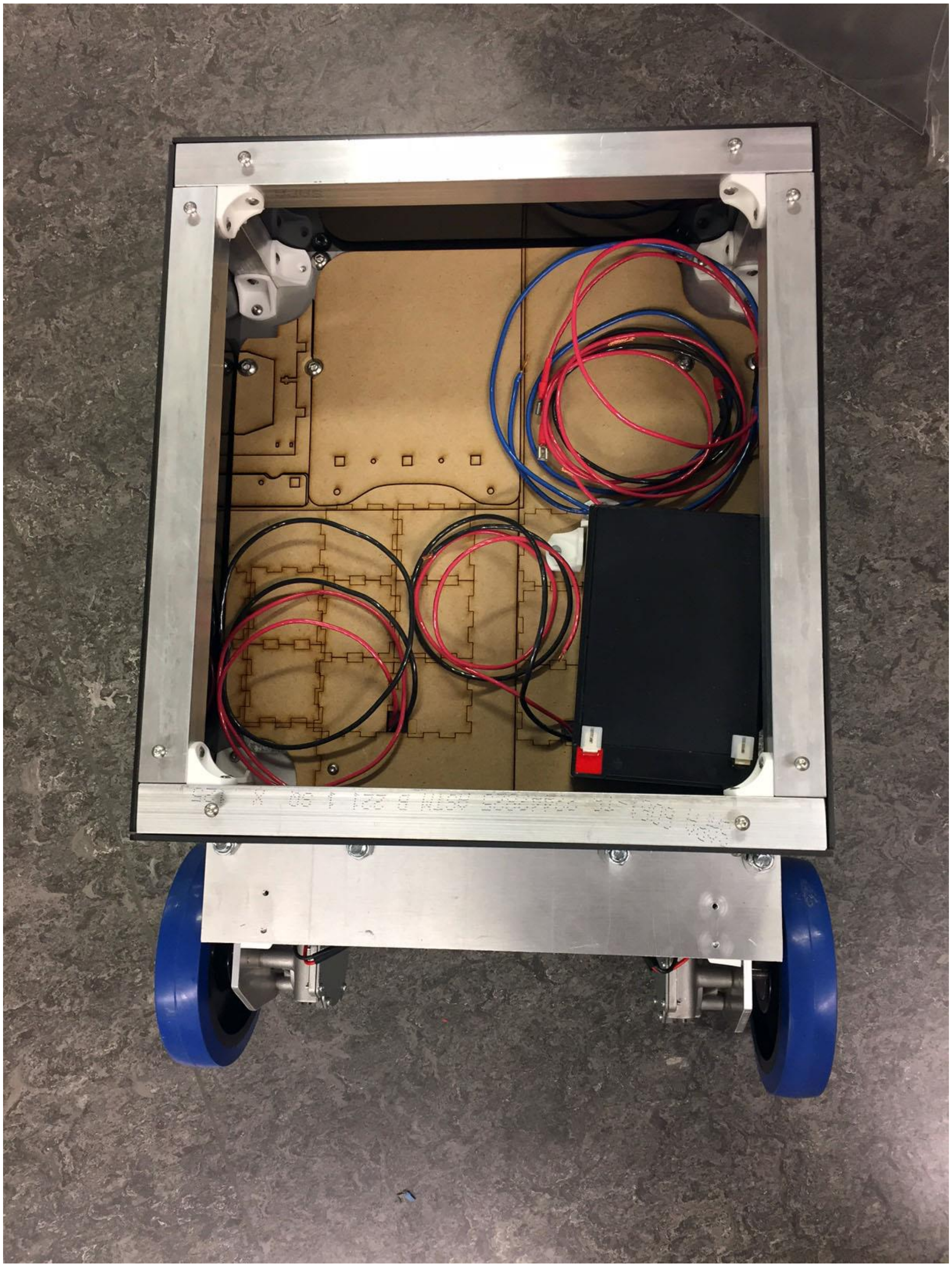


Figure 12: Project Assembly 2

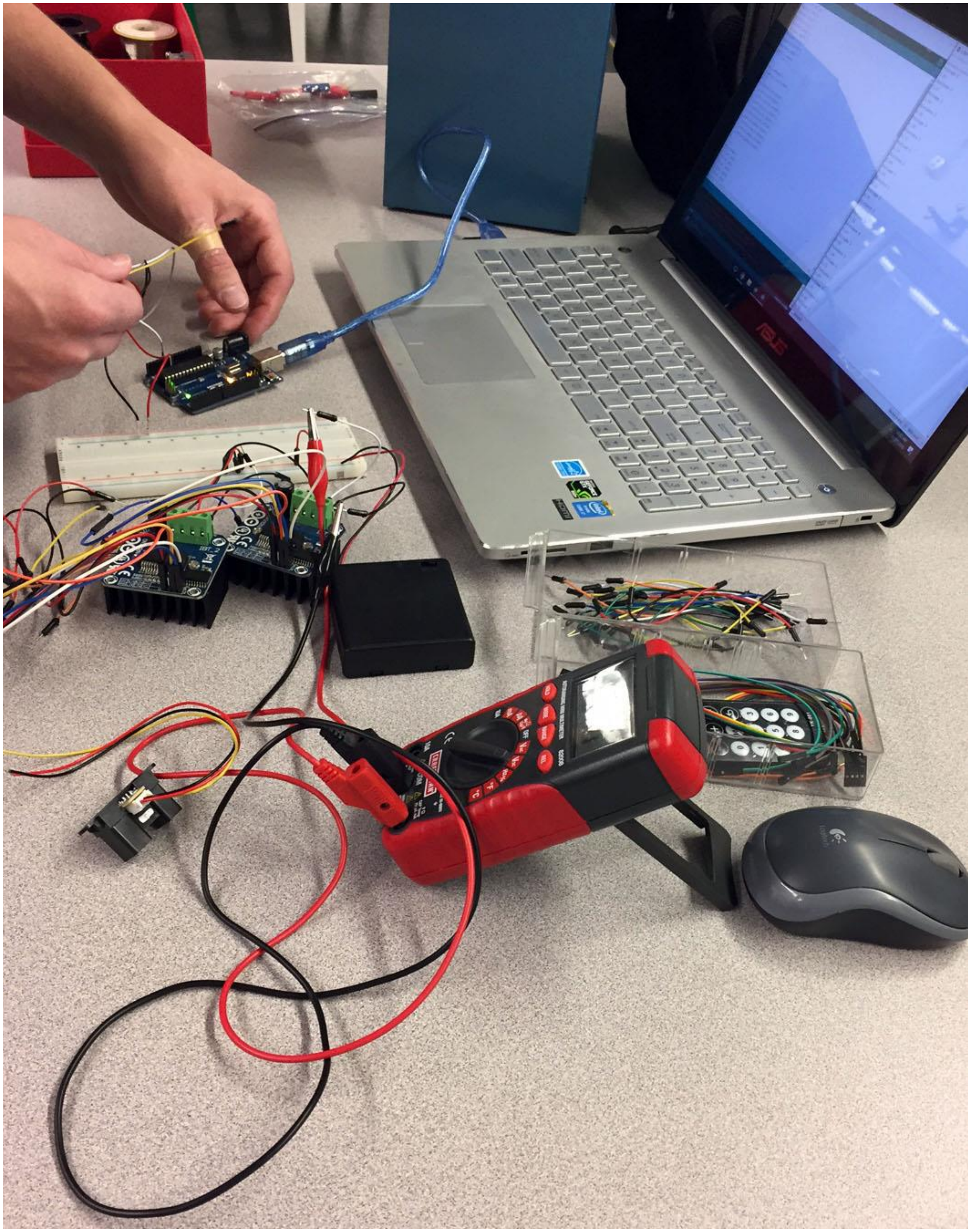


Figure 13: Project Assembly 3

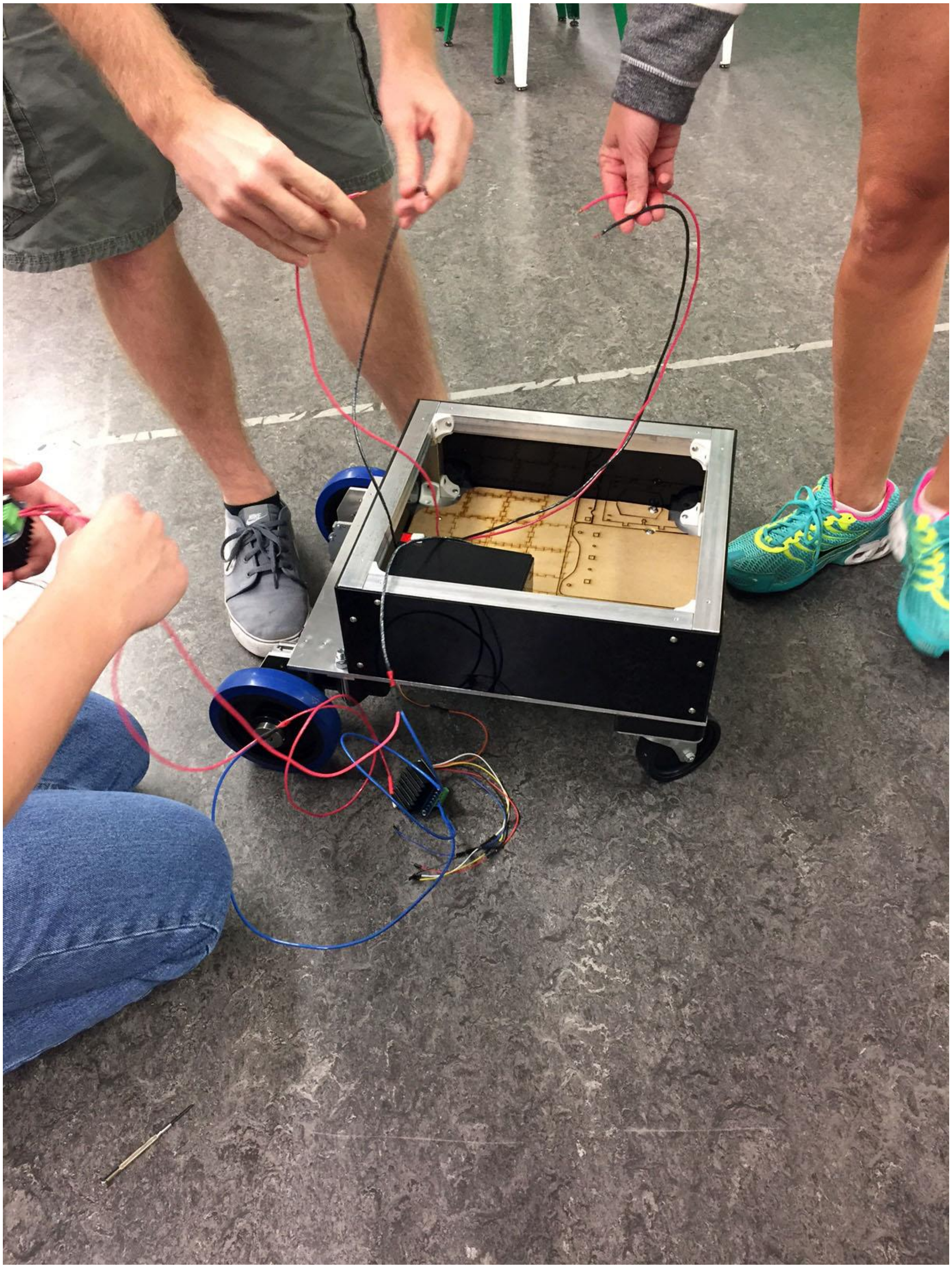


Figure 14: Project Assembly 4

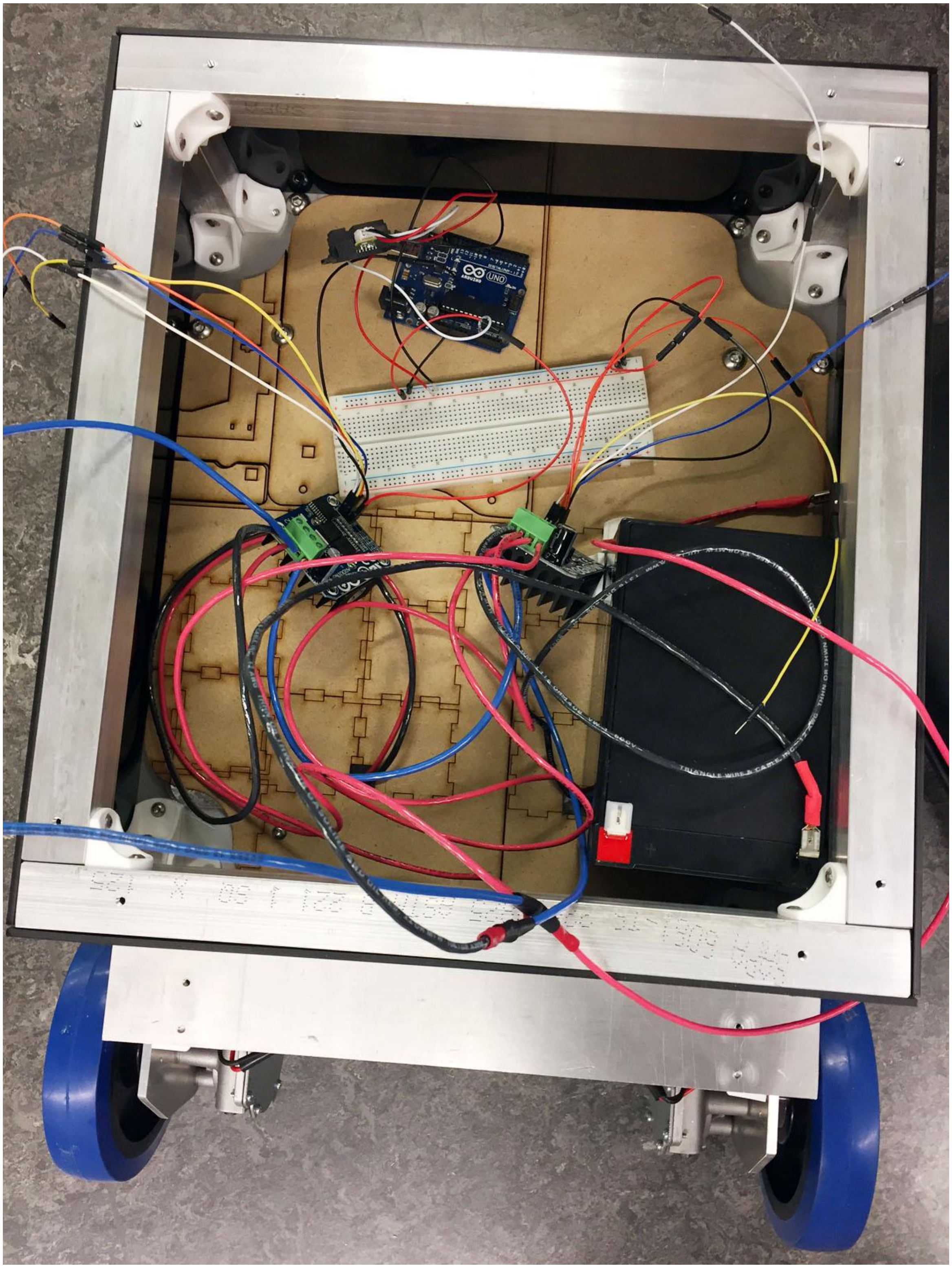


Figure 15: Project Assembly 5

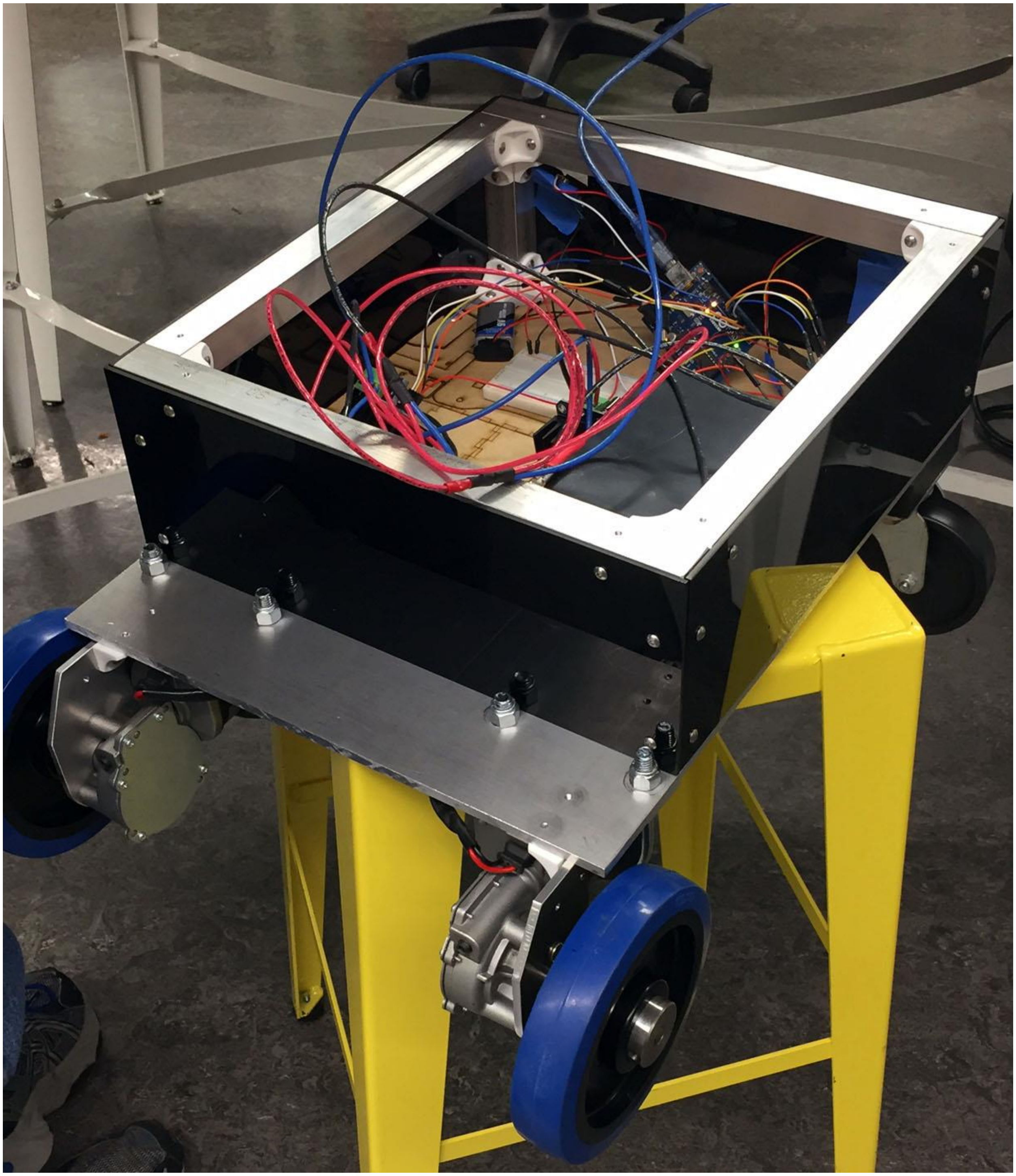


Figure 16: Project Assembly 6

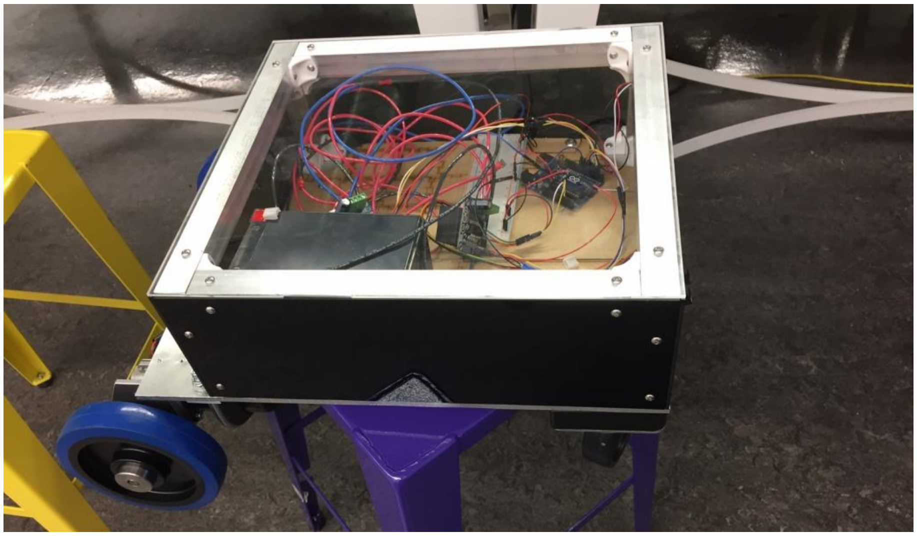


Figure 17: Project Assembly 7

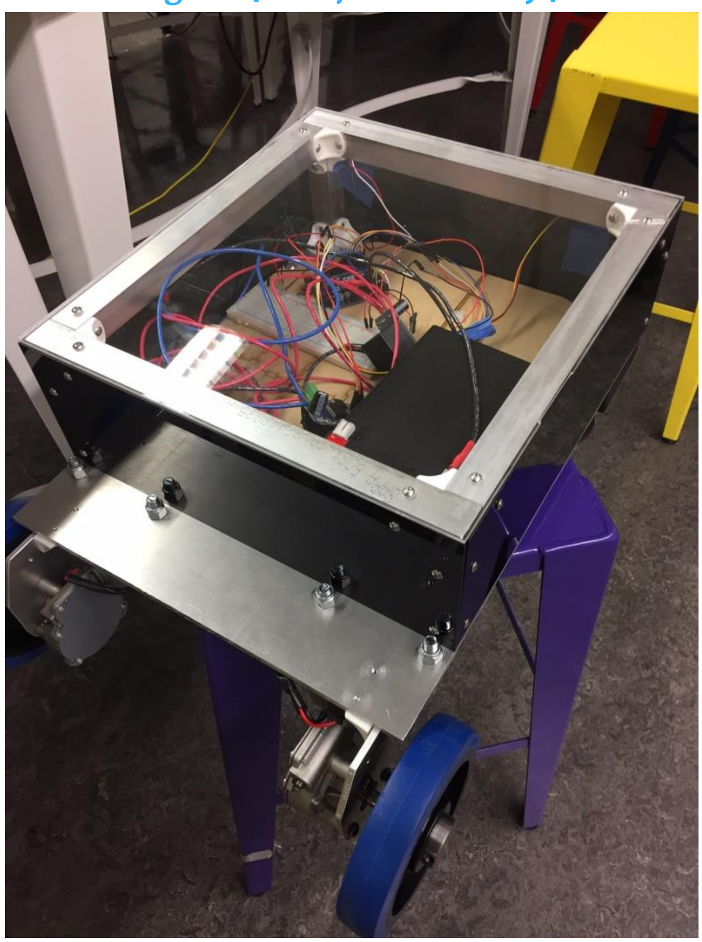


Figure 18: Project Assembly 8