Numair Ahmed
Erik Contreras
Hector Herrera
Alvin Lieu

Reconfigurable Desktop Grinding Machine using Modular Robotics

Sponsored by Dr. Barbara Linke

Faculty Mentor: Professor Linke
FINAL REPORT

EME-185A/B

PROFESSOR STEPHEN ROBINSON
PROFESSOR CRISTINA DAVIS
TEACHING ASSISTANT MICHAEL SCHIRLE

MAY 29, 2015

TABLE OF CONTENTS

Abstract

This motor-module prototype establishes the groundwork for a reconfigurable desktop machine that would allow users to adapt the machine to a workpiece. Potential uses of the machine include grinding, sanding, detailing and printing parts in a repeatable manner. Additionally, its small size and low cost make it ideal for hobby and educational purposes. The design of the project primarily focuses on accurate tracking and programmability of the machine's end-effector (or tool), and robustness to withstand stresses during operation. As such, the modular linkages that comprise the machine feature high-torque motors to counteract vibrations and improve stability.

The design uses the compact 211 NEMA 11 stepper motor joined with a custom-built 1:150 ratio gearbox, and the Arduino Mini 05 microcontroller coupled with two stepper motor drivers, all contained in a 3D-printed PLA housing module. Through testing, the module was shown to have achieved very high precision. However, the module still requires improvements, mainly better transfer of power through the gearbox by implementing stronger gears. Other developments for the future include implementation of an efficient battery pack that will allow untethered operation, wireless communication via bluetooth, position tracking, closed-loop control and additional degrees of freedom (DOF).

FINAL LAYOUT DRAWINGS OF SYSTEM AND SUBSYSTEMS

Complete Motor Module (refer to Figure 31 for complete assembly)

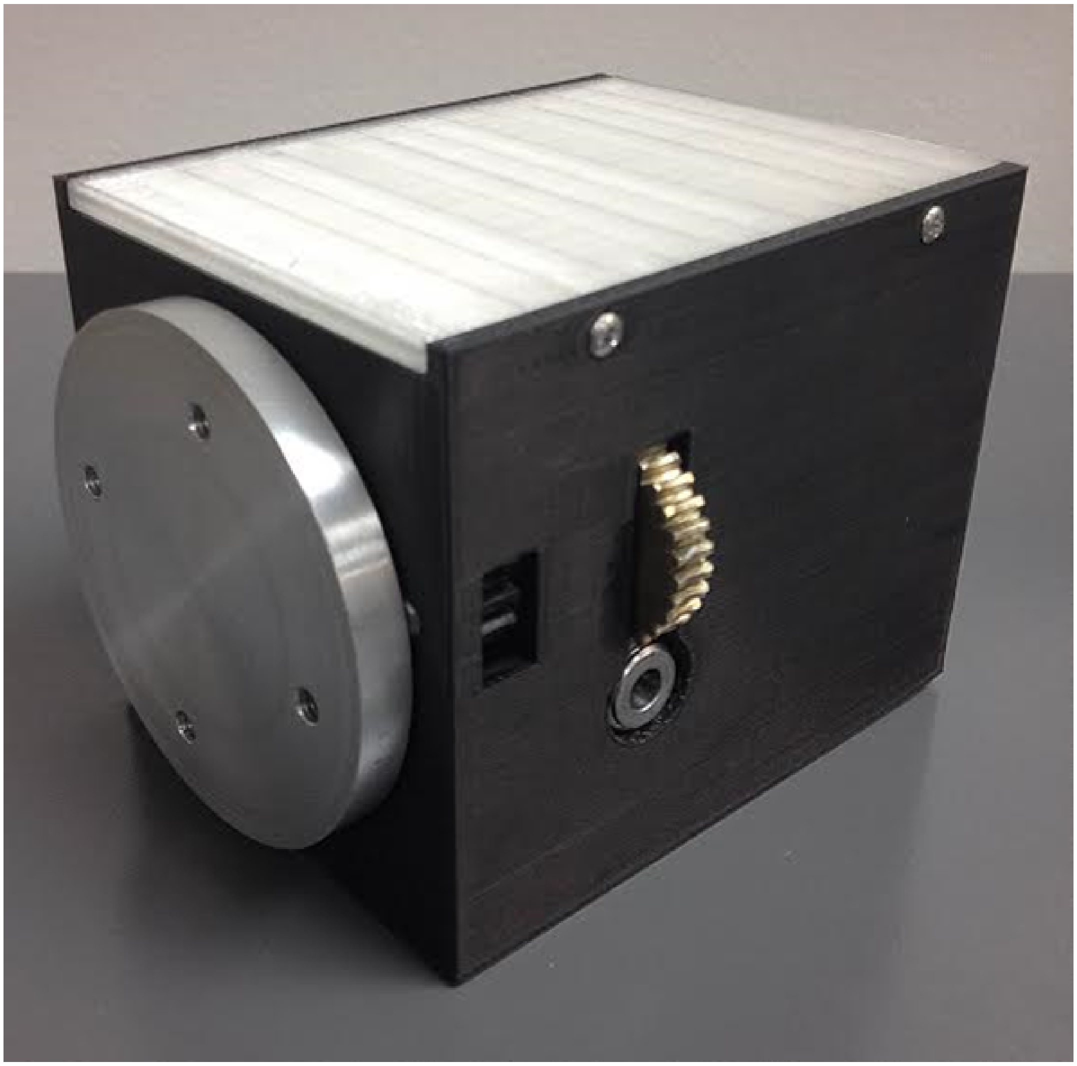


Figure 1: *Assembled motor module unit*

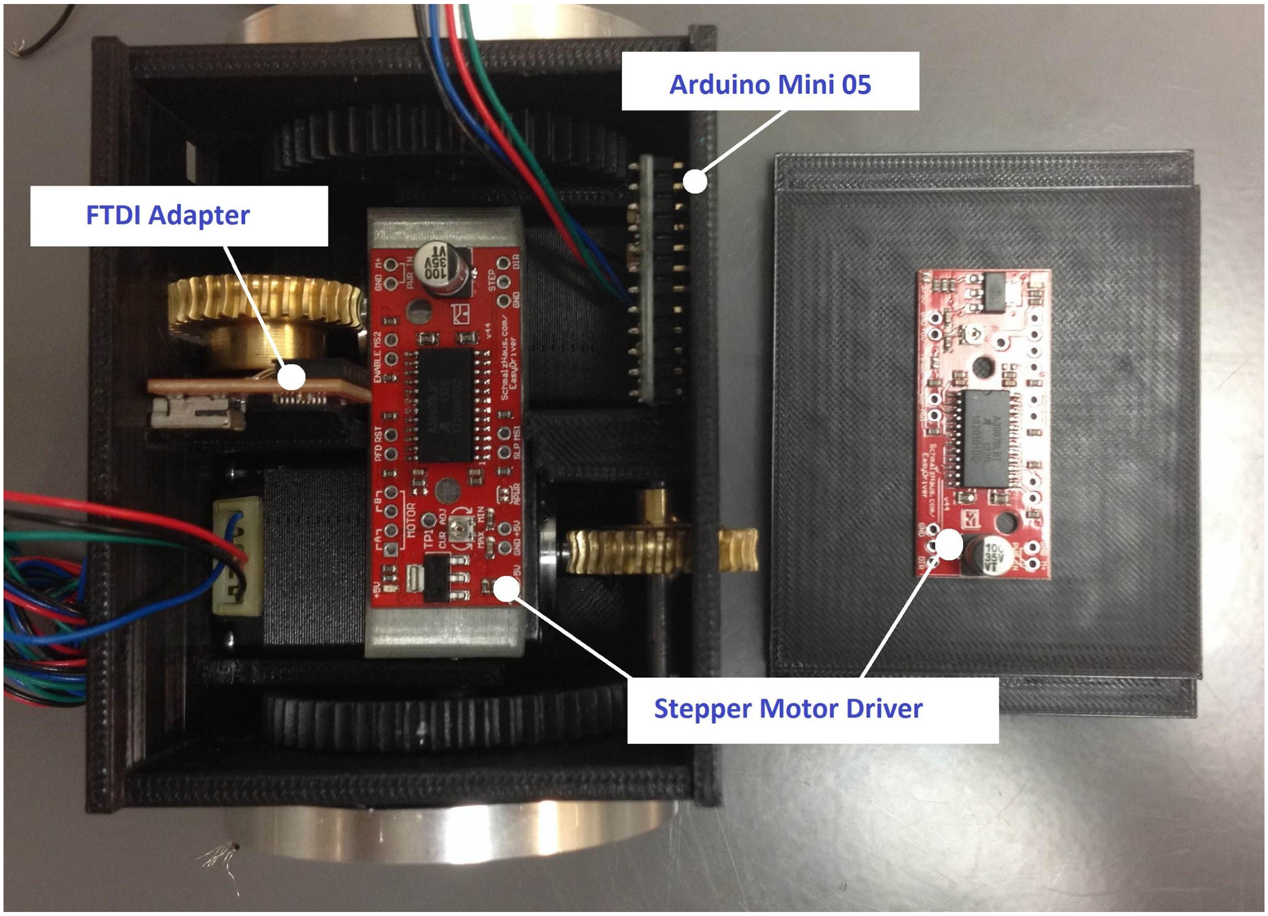


Figure 2: *Internal components of housing*

Gearbox

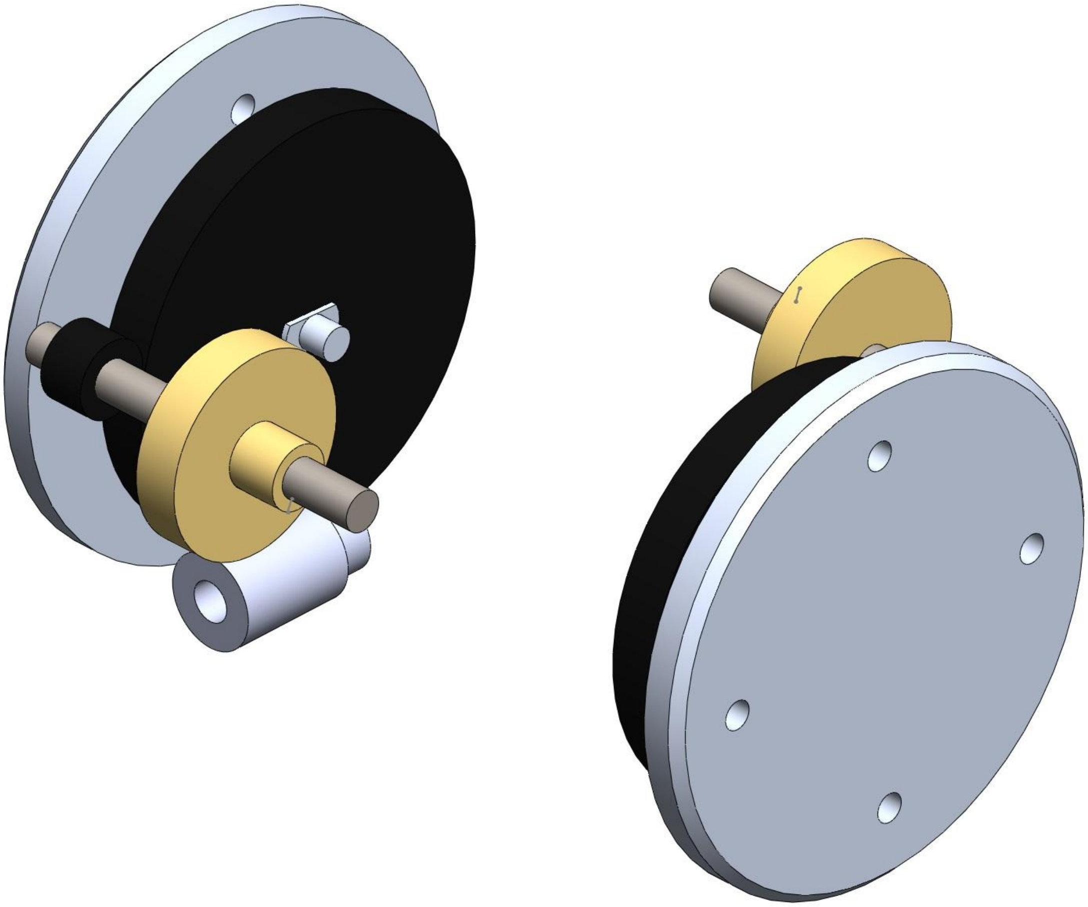


Figure 3: *Gearbox subassembly*

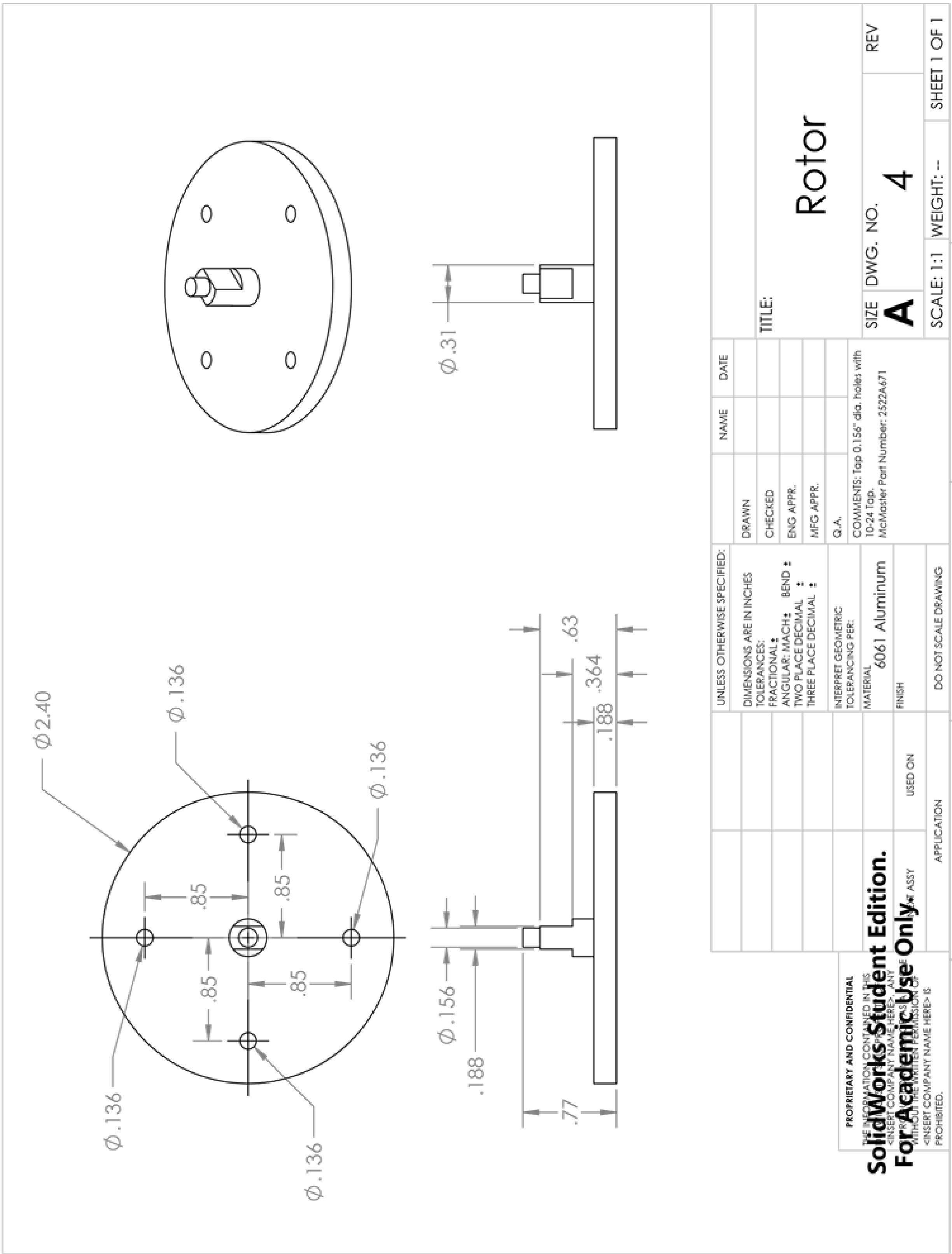


Figure 4: Working drawing of *aluminum rotor*

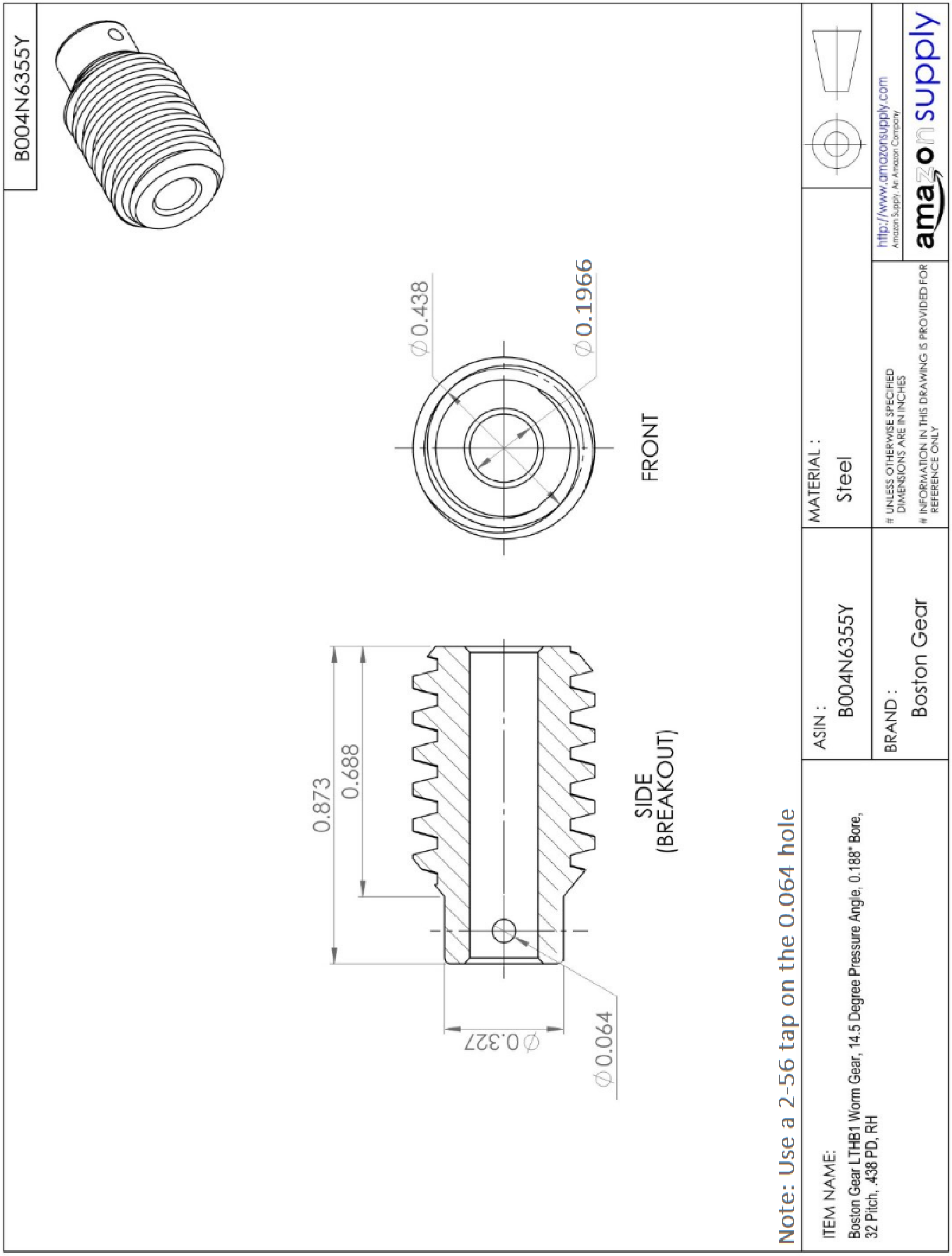

B004N6355Y
Ø0.438
Ø0.1966
FRONT
0.873
0.688
SIDE
(BREAKOUT)
Ø0.327
Ø0.064
Note: Use a 2-56 tap on the 0.064 hole
ITEM NAME:
Boston Gear LTHB1 Worm Gear, 14.5 Degree Pressure Angle, 0.188" Bore, 32 Pitch, .438 PD, RH
ASIN :
B004N6355Y
BRAND :
Boston Gear
MATERIAL :
Steel
# UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
# INFORMATION IN THIS DRAWING IS PROVIDED FOR REFERENCE ONLY
http://www.amazonsupply.com
Amazon Supply. An Amazon Company
amazon supply

Figure 5: *Working drawing of steel worm*

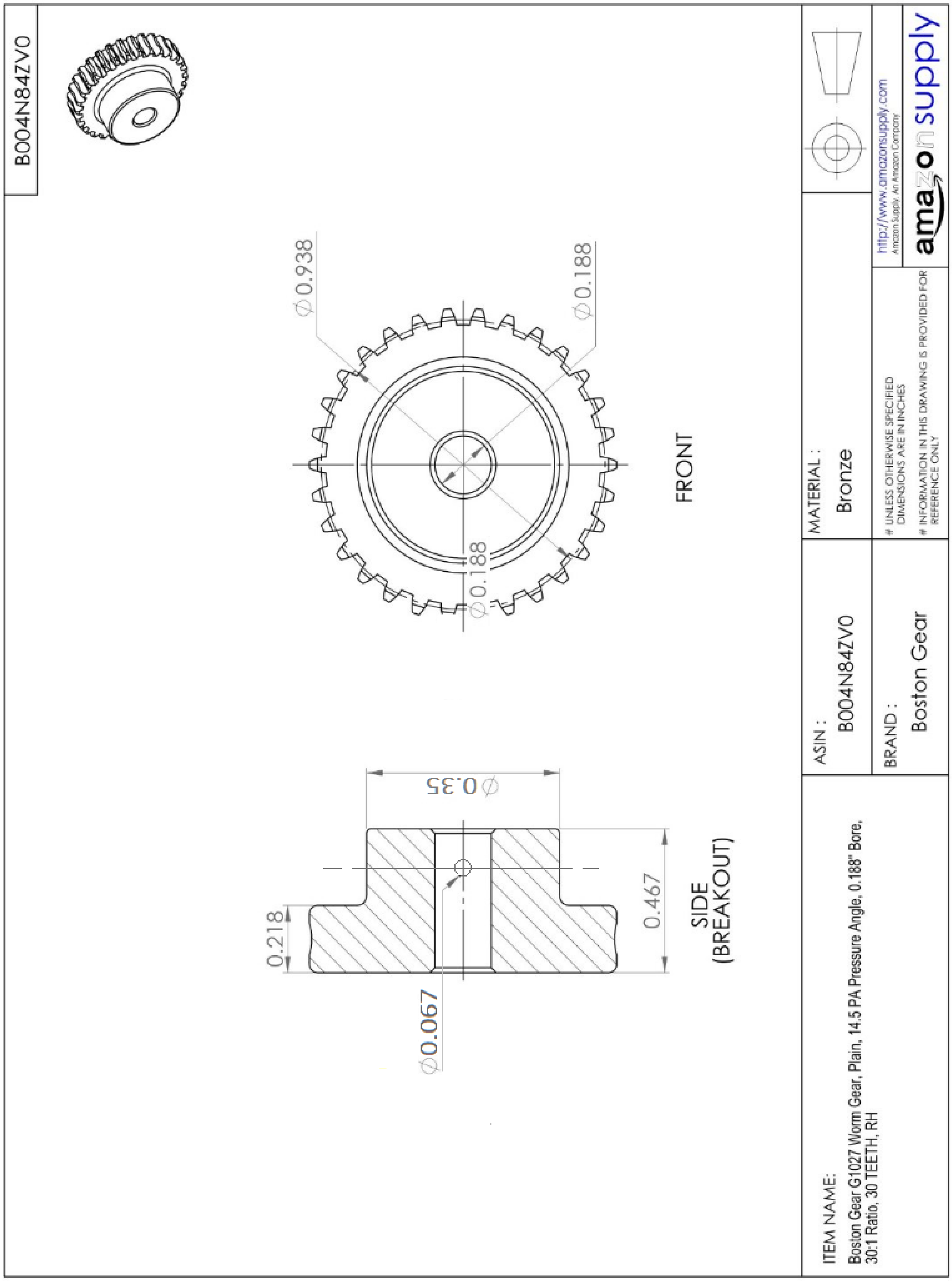

B004N84ZV0
Ø0.938
Ø0.188
Ø0.188
FRONT
Ø0.35
0.218
0.467
Ø0.067
SIDE
(BREAKOUT)
ITEM NAME:
Boston Gear G1027 Worm Gear, Plain, 14.5 PA Pressure Angle, 0.188" Bore, 30:1 Ratio, 30 TEETH, RH
ASIN :
B004N84ZV0
BRAND :
Boston Gear
MATERIAL :
Bronze
# UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
# INFORMATION IN THIS DRAWING IS PROVIDED FOR REFERENCE ONLY
http://www.amazonsupply.com
Amazon Supply, An Amazon Company
amazon supply

Figure 6: *Working drawing of brass worm gear*

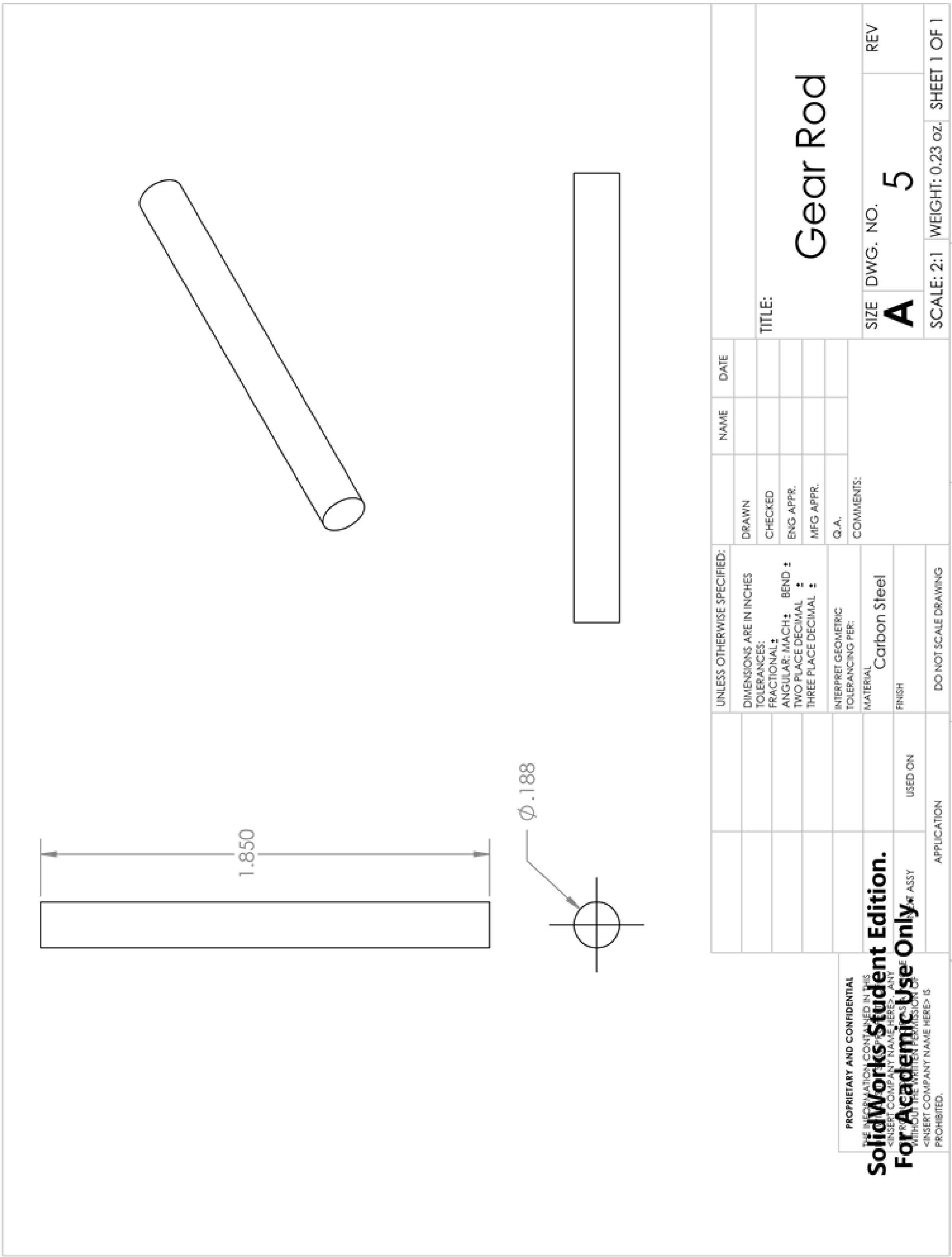

1.850
Ø.188
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
Carbon Steel
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Gear Rod
SIZE
A
DWG. NO.
5
REV
SCALE: 2:1
WEIGHT: 0.23 oz.
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
SolidWorks Student Edition. For Academic Use Only.
1
2
3
4
5

Figure 7: *Working drawing of steel gear rod*

Dual-motor subassembly

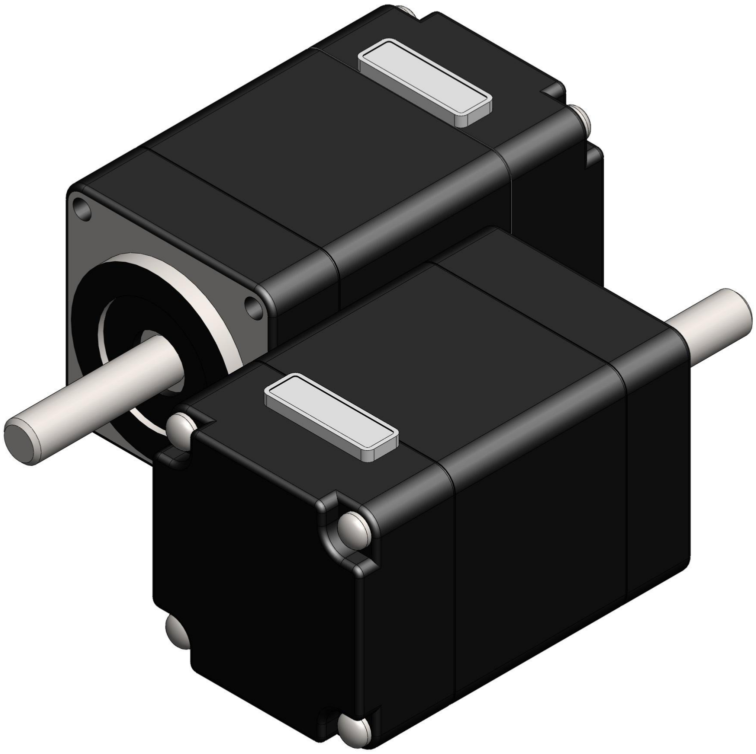


Figure 8: *Two NEMA 11, 211 stepper motors*

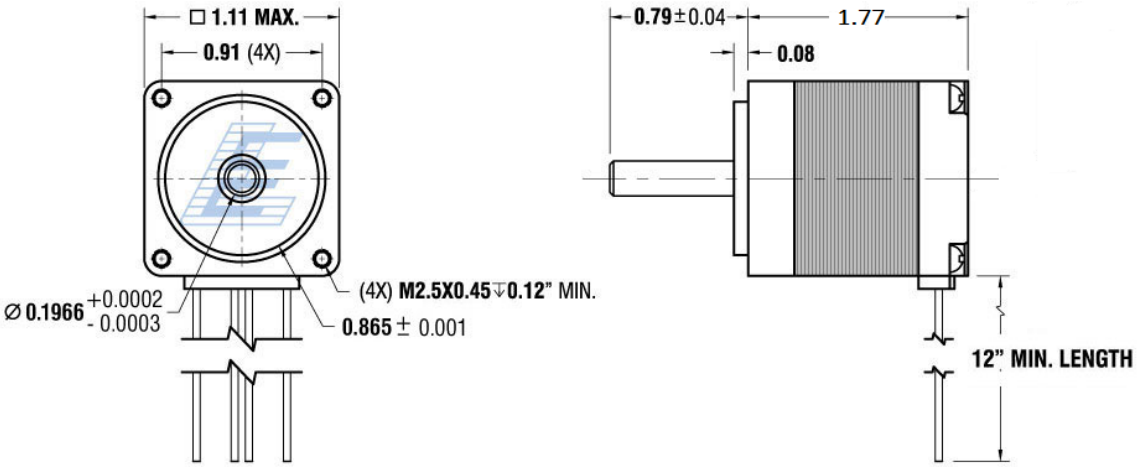


Figure 9: *Dimensions of the Nema 11 Stepper motors*

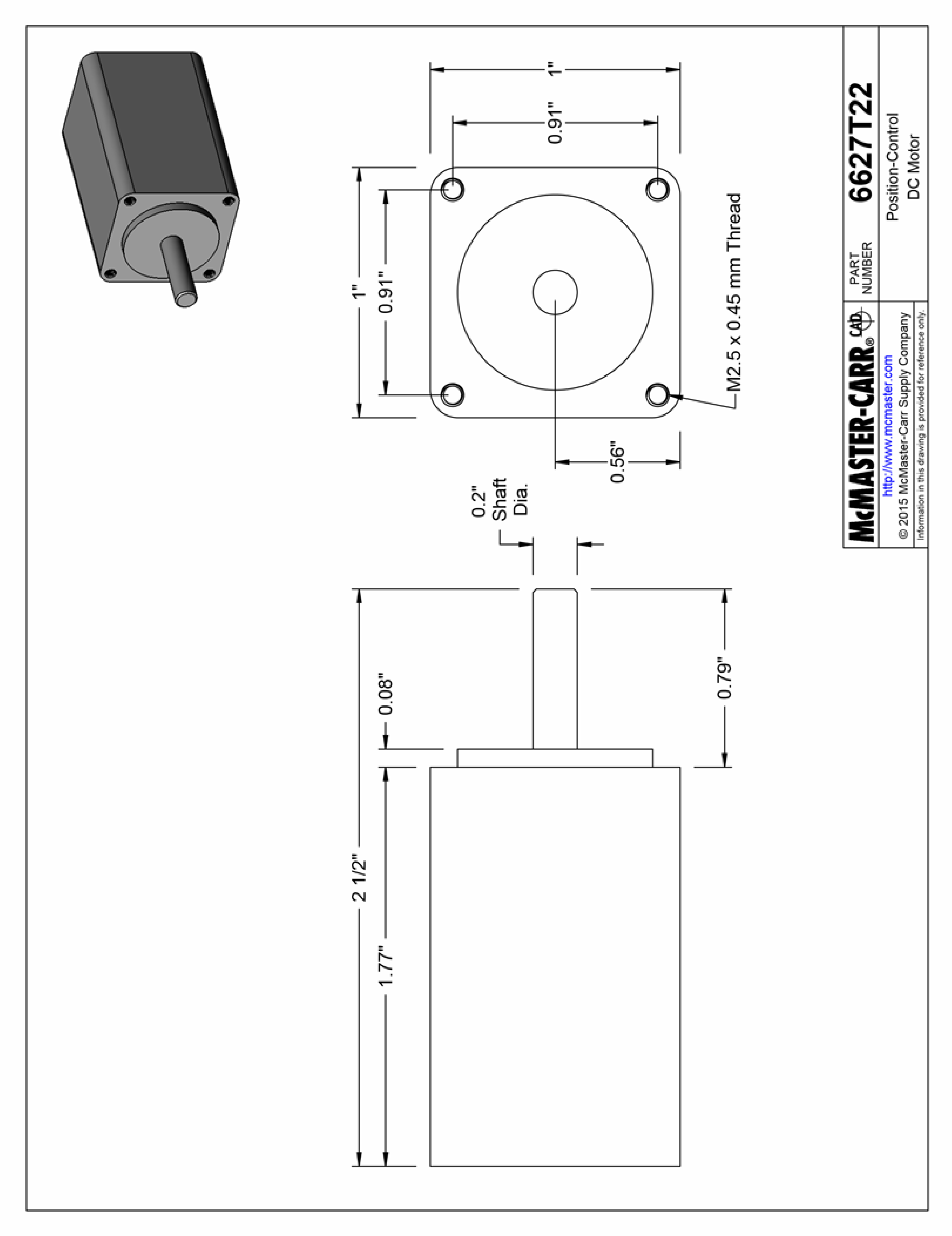


Figure 10: *Working drawing of Nema 11 Stepper Motors*

PLA Housing

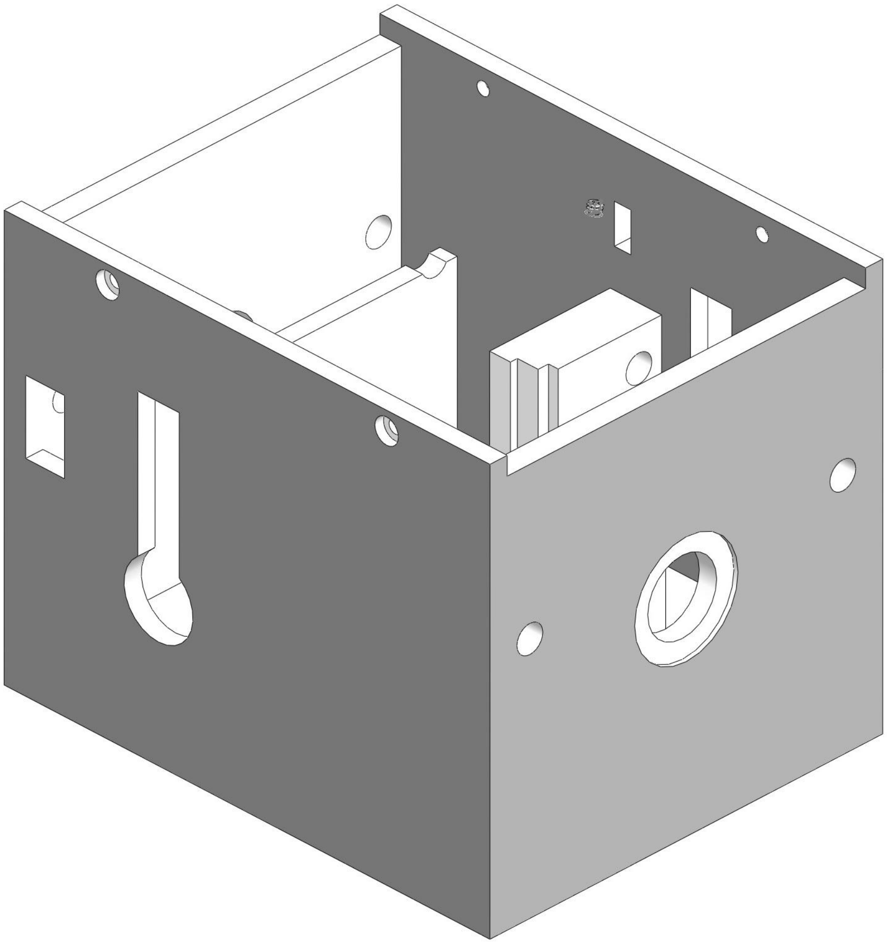


Figure 11: *PLA housing subassembly*

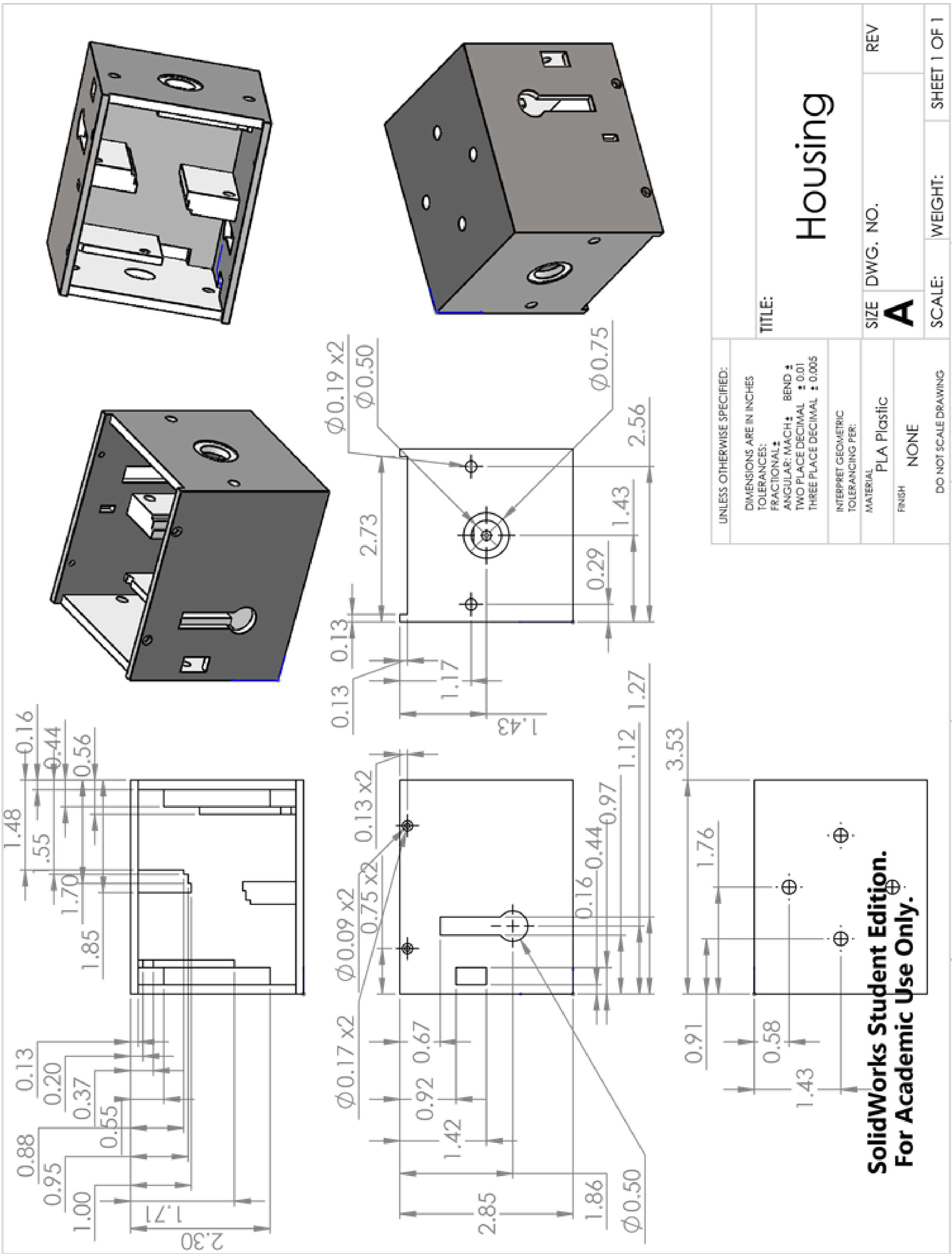


Figure 12: *Working drawing of PLA plastic housing*

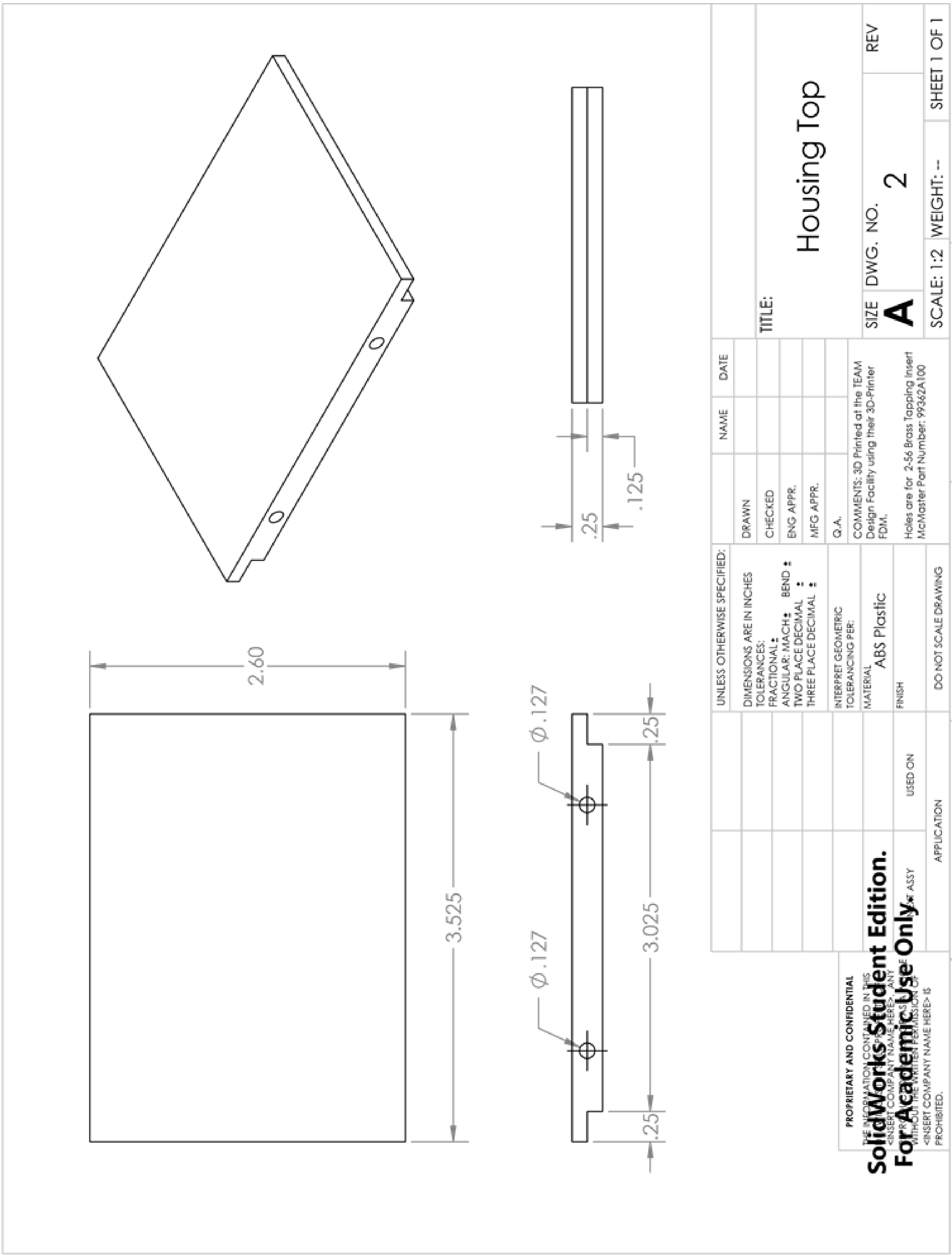


Figure 13: *Working drawing of PLA Housing Lid*

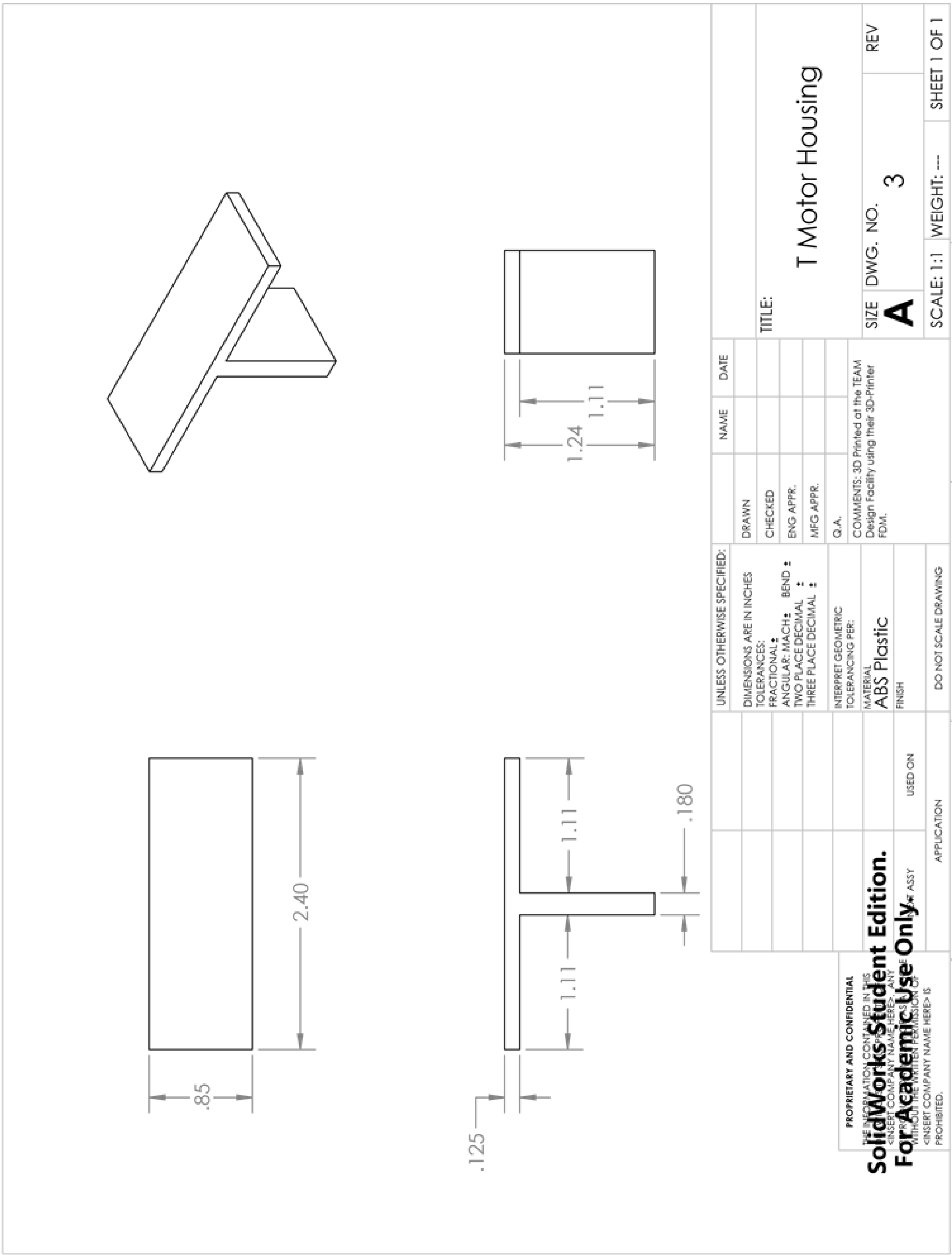
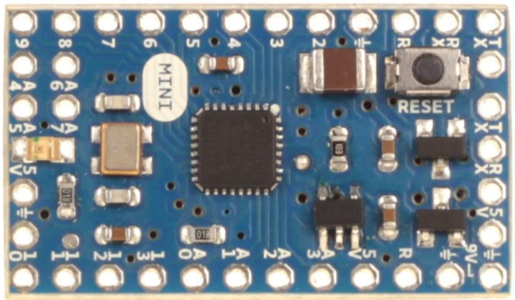

.85
2.40
.125
1.11
1.11
.180
1.24
1.11
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
ABS Plastic
FINISH
DO NOT SCALE DRAWING
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS: 3D Printed at the TEAM Design Facility using their 3D-Printer FDM.
NAME
DATE
TITLE:
T Motor Housing
SIZE
A
DWG. NO.
3
REV
SCALE: 1:1
WEIGHT: ---
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS [illegible] <INSERT COMPANY NAME HERE>. ANY [illegible] WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
SolidWorks Student Edition.
For Academic Use Only.
1
2
3
4
5

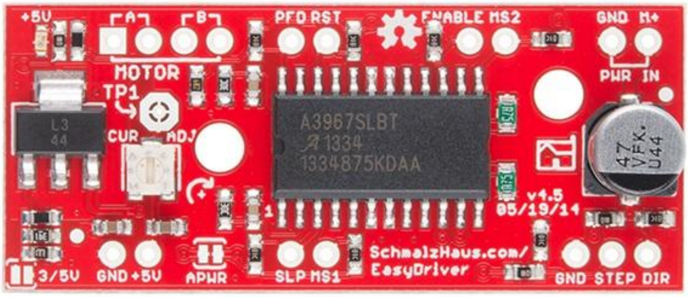
Figure 14: *Working drawing of T-Motor housing constraint*

Electronics Architecture

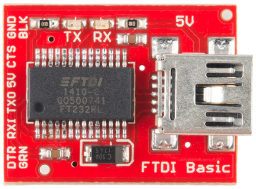
Main logic components:



Arduino Mini 05 microcontroller



EasyDriver Motor Driver



Sparkfun FTDI to Serial Adapter

FINAL BILL OF MATERIALS

The following table outlines the bill of materials for a single motor module. This project had an budget constraint of $500 per module. A total of 2 modules were constructed with each module amounting to $499.67. The team successfully stayed within budget for a total of $999.34.

Table 1: *Bill of Materials for both manufacturing stock and purchased material*

Bill of Materials						
Component	**Provider**	**Part Number**	**Material - Part Description**	**Cost per Unit**	**Qty.**	**Total Cost**
Multipurpose 6061 Aluminum Rod, 1/2 ft	Mcmaster Carr	8974K77	2 1/2" Diameter, for rotors	$30.33	1	$30.33
1045 Medium Carbon Steel High-Strength Rod	Mcmaster Carr	8279T19	3/16" Diameter; for gear rods	$13.97	1	$13.97
10-24 Tap	Mcmaster Carr	2522A671	For rotor and base threads	$4.96	1	$4.96
2-56 Tap	Mcmaster Carr	2522A663	General Purpose Tap for set screw	$9.72	1	$9.72
Type 316 Stainless Steel Pan Head Phillips Machine Screw	Mcmaster Carr	91735A240	10-24 Thread, 3/8" Length	$7.68	1	$7.68
Type 316 Stainless Steel Pan Head Phillips Machine Screw	Mcmaster Carr	91735A013	2-56 Thread, 5/16" Length	$11.02	1	$11.02
Type 316 Stainless Steel Cup Point Set Screw	Mcmaster Carr	92313A010	2-56 Thread, 1/16" Long; Pack of 100	$9.80	1	$9.80
Brass Tapping Insert for Plastics, 10-24	McMaster Carr	99362A500	Threaded Insert x10	$10.02	1	$10.02
Brass Tapping Insert for Plastics, 2-56	Mcmaster Carr	99362A100	Threaded Insert x10	$7.78	1	$7.78
Mini High-Precision Stainless Steel Ball Bearing - ABEC-5	McMaster Carr	57155K338	Flange Shield, Extended Inner Ring, 5/16" ID, 1/2" OD	$8.72	2	$17.44
EasyDriver Stepper Motor Driver	Amazon	B004G4XR60	Stepper Motor Driver	$8.00	2	$16.00
Arduino Mini 05	Amazon	B00HUB2NXO	Microcontroller	$12.95	1	$12.95
FTDI Adapter	Amazon	DEV-097168	FTDI to Serial Adapter	$16.95	2	$33.90
Boston Gear, Worm	Amazon	B004N6355	14.5 Degree Pressure Angle,	$18.97	2	$37.94

		Y	0.188" Bore, 32 Pitch			
Boston Gear, Worm Gear	Amazon	B004N84ZV0	4.5 PA Pressure Angle, 0.188" Bore, 30:1 Ratio, 30 Teeth	$27.80	2	$55.60
Mini Step (DC Stepper Motor)	Lin Engineering	211-18-01	NEMA 11, 13.7 oz-in Holding Torque	$45.90	2	$91.80
Durable Plastic Gears	Jameco	1810072	Nylon Gear, 50 teeth and 10 teeth cogs	$7.95	2	$15.90
Motor Module Housing	TEAM Prototyping Facility	N/A	3D Printed Housing (PLA plastic)	$94.00	1	$94.00
*Note: Cost for two modules is twice the total amount listed on this table.			**Shipping & Handling + Tax**			
			Mcmaster-Carr		$7.56	
			Amazon (Prime)		$2.09	
			Lin Engineering		$5.43	
			Jameco		$3.78	
				TOTAL	**$499.67**	

SUMMARY REPORT

The primary feature of this machine product and its most novel feature is its modularity, or mechanical compatibility with like-modules. With multiple modules, a user can configure them into several different machine tools to meet their machining needs. The module linkage system offers a tool loading factor of safety (FOS) of 1.45.

This motor module features: 1) high angular precision with step size of 0.012 degrees and an uncertainty of 0.006 degrees, 2) high torque output of 255 oz in, 3) small form factor of 2.3x2.85x3.5 inches, and 4) implementation of an Arduino microcontroller for logic controls. The precision featured in this product allow for highly accurate placement of an end-effector, such as grinders or mill cutters, that may be attached in a machine tool set-up. This precision facilitates fabrication of small details and features into a user’s workpiece. Coupled with high torque output, the modules can maintain structural stability under the high reactionary stresses of machining. The module’s small form factor significantly reduces the size of required workspace more common of traditional machine tools.

This motor module uses the Arduino Mini 05 microcontroller. With an extensive support community and freely available Arduino IDE, this microcontroller allows for easy programming of the motor module, even for novice software developers. This feature thus allows for continued improvement of the product over time, and facilitates the addition of new components, should they be desired.

These modules were constructed from several materials. The module’s housing is made of 3D-printed PLA plastic. The use of plastic prevents the housing from causing any wireless signal interference with electronic components that will be implemented in future designs. 3D printing the product allowed the creation of very small internal features for securing the electronic components, as exhibited by Figure 2 and Figure 12. The rotors that allow attachment of linkages to the module were made from aluminum via traditional lathe and mill machining (Figure 4). Other components of the module were made in similar fashion. To assemble, the components were placed inside the plastic housing and either screwed together or fitted. The Nylon gears were secured to the rod via knurling and epoxy. The wires were soldered together and chips isolated with electrical tape. More in-depth details of the manufacturing and assembly of the module can be found in Appendix II and Appendix III.

To operate, a user will input a specified angle command. This command is inputted to the module’s microcontroller. The microcontroller interacts with the motor driver to feed the dual stepper motors a PWM signal. These PWM signal commands will run the stepper motor, which will then turn the worm screw attached to the motor’s shaft. Through the gear box assembly of Figure 3, the motor module is able to output a torque much higher than the motor’s output torque. Detailed steps on operating the module’s software controls are included in Appendix II.

The gearbox, which features a worm drive, provides high rigidity against reactionary forces in a small space. This worm drive allows the motor to drive the gear assembly while disallowing torque to be back-driven on the motor due to the torque applied from external loading of the rotors. This feature prevents damage to the motor should too high a load be applied. Additionally, the large 1:150 gear ratio of the gearbox significantly reduces the angular speed of the rotor, in reference to the motor inputted angular velocity. Because the module will be used mainly for small machining, high speed was not a priority of the design.

This design still has room for improvement, including: 1) the plastic gears, 2) 3D Housing, 3) Motor Drivers, 4) implementing of position sensors.

The plastic gears wore very quickly during operation. During precision testing without a load, the module proved to be highly precise (See Testing section in Appendix I). Under load however, the plastic gears began to flex and eventually degraded at the keyhole. This wearing reduced the precision. Steel gears, and set screws could be implemented into the design for more durability and precision. The decision to use plastic gears was mainly due to budget constraints.

The 3D printed housing, while: 1) easy to manufacture, 2) lightweight and 3) Bluetooth compatible, had a few shortcomings in printout accuracy, causing misalignment of holes and warping of the base. These issues caused the gearbox to lock and required additional machining. An alternative solution is to make the current housing from an aluminum U-channel and a 3D printed PLA top (see Figure 31 in Appendix V). This design would provide rigidity and accuracy of the shaft holes.

The initial design accounted for the use of two stepper motors to rotate two output rotors facing opposite each other on either side of the motor module. The two steppers require the use of two motor drivers for power and logic control. At the current stage of development, complete use of both motors has not been established due to flaws in the the code logic for simultaneously controlling two EasyDriver stepper motor drivers. At this stage, complete robust control is possible for one EasyDriver at a time, and thus one side of the motor module. To remedy this issue, further research into motor drivers and the transistor chips used is necessary. The complete understanding of transistors used in motor drivers will aid in proper programming logic for future development. Even with just half of the module being operational, the module exhibits the desired features of precision and rigidity described earlier.

And while the product already features high precision, a position sensor could be implemented for even more precision. Even more advantageous would be the ability to integrate closed-loop control. Variation between intended angle and output angle, even if small, can affect the machinability of the end product. Position tracking can account for the flexing or vibration of the modules under machining. Moreover, closed-loop control could open the door for more advanced machining tasks.

Other suggestions for future development, include

- implementation of a battery pack
- wireless communication via bluetooth
- graphic user interface (GUI)
- larger form factor with larger motors

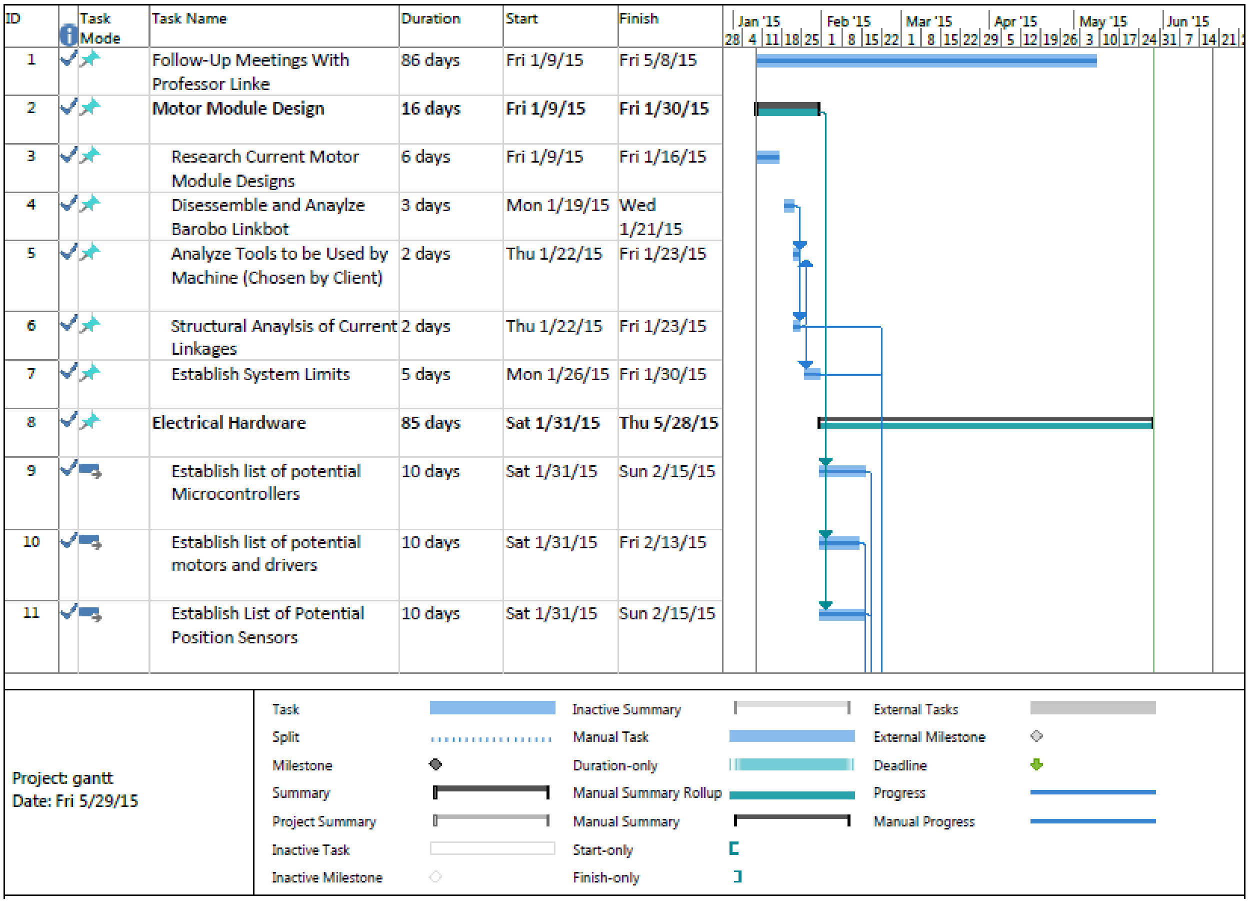
The implementation of a battery pack will allow for a more self contained product, and better service the need for a small and portable machine tool. Wireless communication would grant more cohesive functionality of several modules. This feature would truly complement the modular vision of the product. As with the position sensor, the ability to communicate with several modules would draw on much more advanced tasks to be performed. The possibility of implementing multiple cross-communicating modules creates the possibility for tool pathing and tracking of machine tool space. A self contained product without the need for external wires will increase modularity.

Furthermore, the creation of a GUI would allow an end-user to easily interact with the product. As the intended consumer of this product is a machine hobbyist, a simple way to interact with the product is absolutely crucial for such a user. A GUI may also allow for more rapid prototyping tasks to be performed.

Finally, adopting this design to a larger form factor may also be necessary, given the desired features of this product. The current product consists of very closely packaged components, leaving very little room for additional components or modification, should they be desired. Hence, expansion of the module's dimensions may be necessary.

PROJECT MANAGEMENT INFORMATION

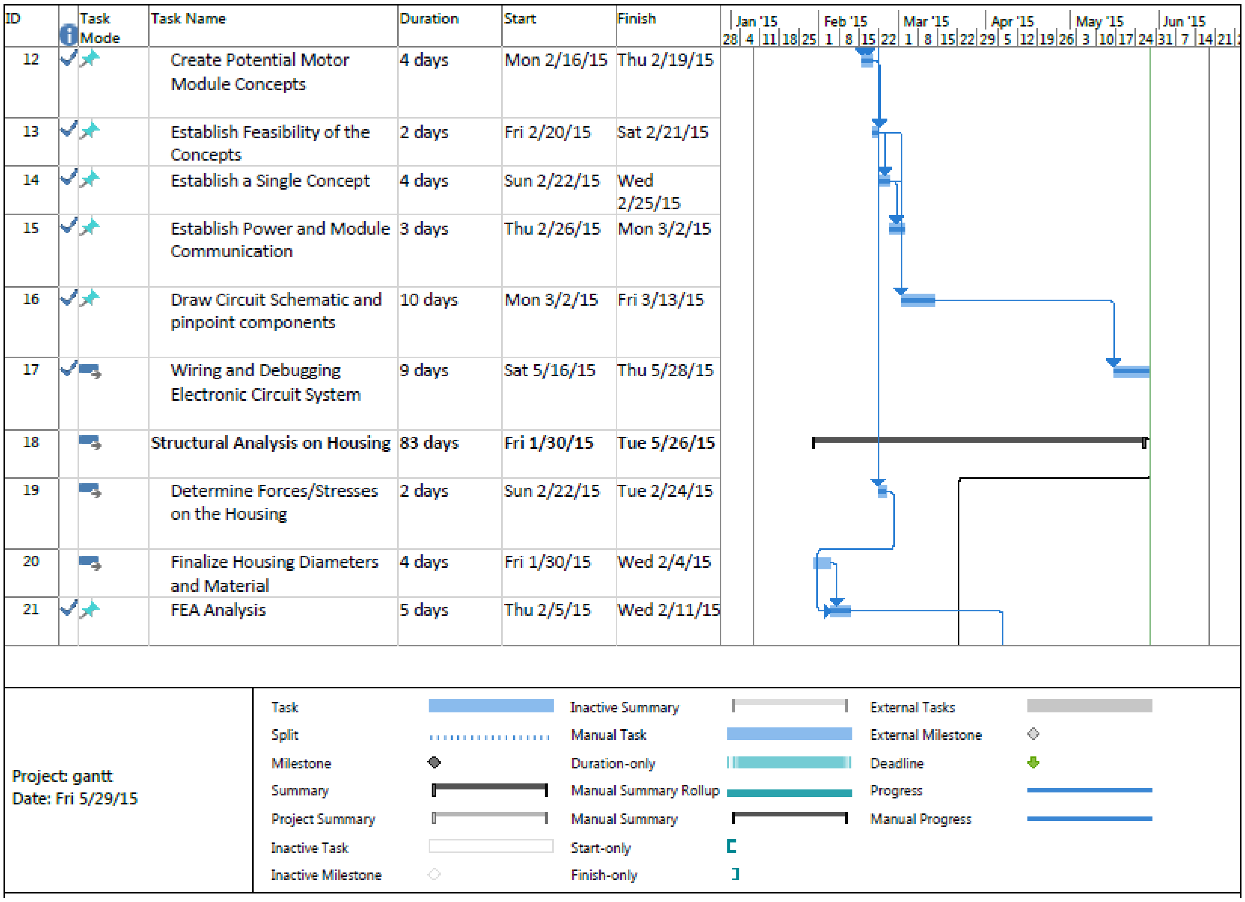
Electro-mechanical components were ordered and arrived on time, as per the Gantt chart. Some complications and subsequent delays had arisen during fabrication and assembly of module parts. Machinery at the EFL was not capable of manufacturing operations for initial material selection, mainly stainless steel. To overcome the issue, the team opted to simplify the geometry, and purchase off-the-shelf parts. Additionally, the decision to expand the usage of 6061 aluminum was made since the material contained sufficient material properties to still withstand the stresses of the module. Assembly also proved difficult due to the small form factor of the modules. The team, thus, made modifications to the housing and added internal hole features into the housing design to accommodate these components. The necessary redesign of the product caused further delay, shifting completion time for assembly and testing to later dates. Other offsets to the Gantt chart included technical difficulties with electronic setup, and debugging of logic code. T.A. Michael Schirle assisted with these issues, however, and the project was able to progress.



ID	ℹ	Task Mode	Task Name	Duration	Start	Finish
1	✓		Follow-Up Meetings With Professor Linke	86 days	Fri 1/9/15	Fri 5/8/15
2	✓		**Motor Module Design**	**16 days**	**Fri 1/9/15**	**Fri 1/30/15**
3	✓		Research Current Motor Module Designs	6 days	Fri 1/9/15	Fri 1/16/15
4	✓		Disessemble and Anaylze Barobo Linkbot	3 days	Mon 1/19/15	Wed 1/21/15
5	✓		Analyze Tools to be Used by Machine (Chosen by Client)	2 days	Thu 1/22/15	Fri 1/23/15
6	✓		Structural Anaylsis of Current Linkages	2 days	Thu 1/22/15	Fri 1/23/15
7	✓		Establish System Limits	5 days	Mon 1/26/15	Fri 1/30/15
8	✓		**Electrical Hardware**	**85 days**	**Sat 1/31/15**	**Thu 5/28/15**
9	✓		Establish list of potential Microcontrollers	10 days	Sat 1/31/15	Sun 2/15/15
10	✓		Establish list of potential motors and drivers	10 days	Sat 1/31/15	Fri 2/13/15
11	✓		Establish List of Potential Position Sensors	10 days	Sat 1/31/15	Sun 2/15/15

Project: gantt
Date: Fri 5/29/15

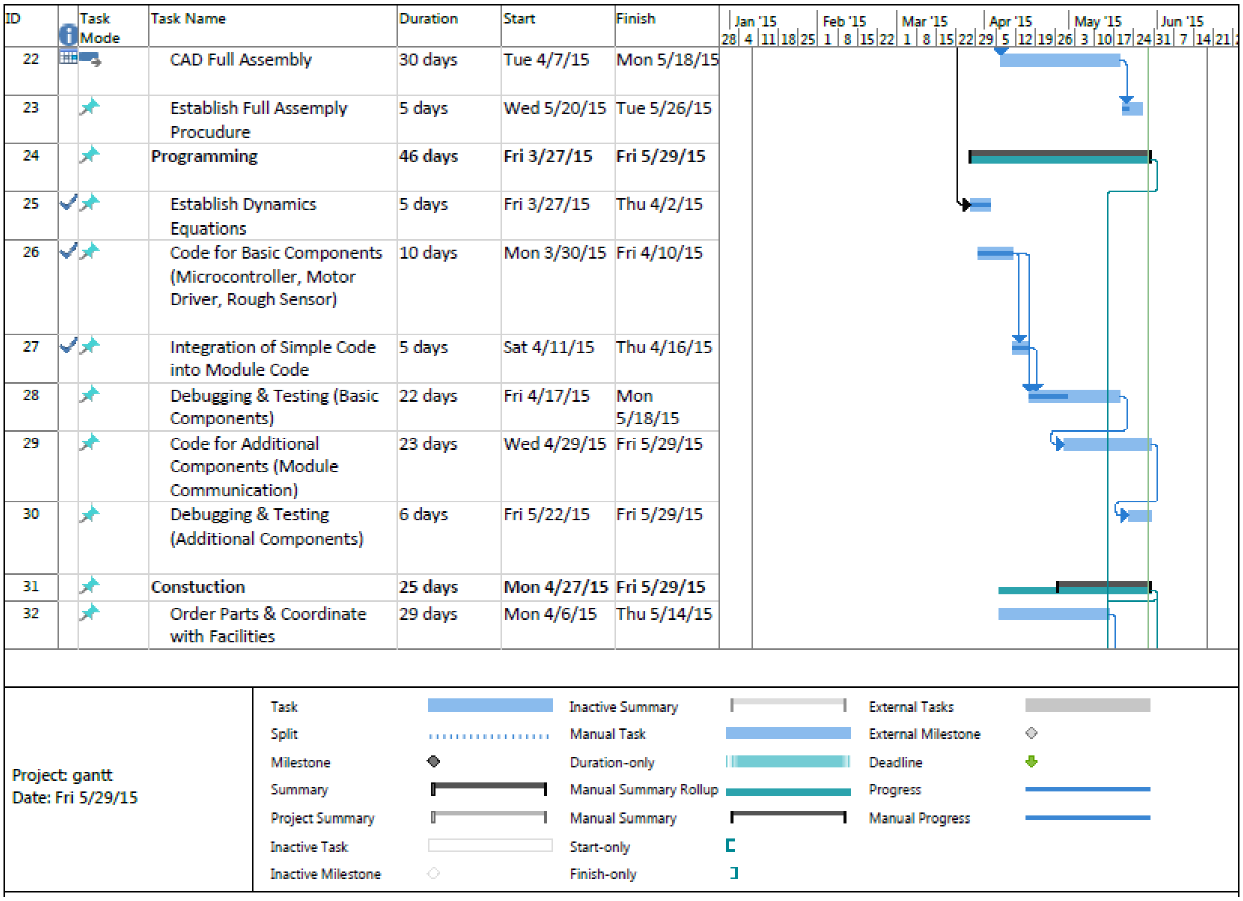
Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	◇
Milestone	◆	Duration-only		Deadline	⇩
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only	[		
Inactive Milestone	◇	Finish-only	]		



ID	ℹ	Task Mode	Task Name	Duration	Start	Finish
12	✓	📌	Create Potential Motor Module Concepts	4 days	Mon 2/16/15	Thu 2/19/15
13	✓	📌	Establish Feasibility of the Concepts	2 days	Fri 2/20/15	Sat 2/21/15
14	✓	📌	Establish a Single Concept	4 days	Sun 2/22/15	Wed 2/25/15
15	✓	📌	Establish Power and Module Communication	3 days	Thu 2/26/15	Mon 3/2/15
16	✓	📌	Draw Circuit Schematic and pinpoint components	10 days	Mon 3/2/15	Fri 3/13/15
17	✓	➡	Wiring and Debugging Electronic Circuit System	9 days	Sat 5/16/15	Thu 5/28/15
18		➡	**Structural Analysis on Housing**	**83 days**	**Fri 1/30/15**	**Tue 5/26/15**
19		➡	Determine Forces/Stresses on the Housing	2 days	Sun 2/22/15	Tue 2/24/15
20		➡	Finalize Housing Diameters and Material	4 days	Fri 1/30/15	Wed 2/4/15
21	✓	📌	FEA Analysis	5 days	Thu 2/5/15	Wed 2/11/15

Project: gantt
Date: Fri 5/29/15

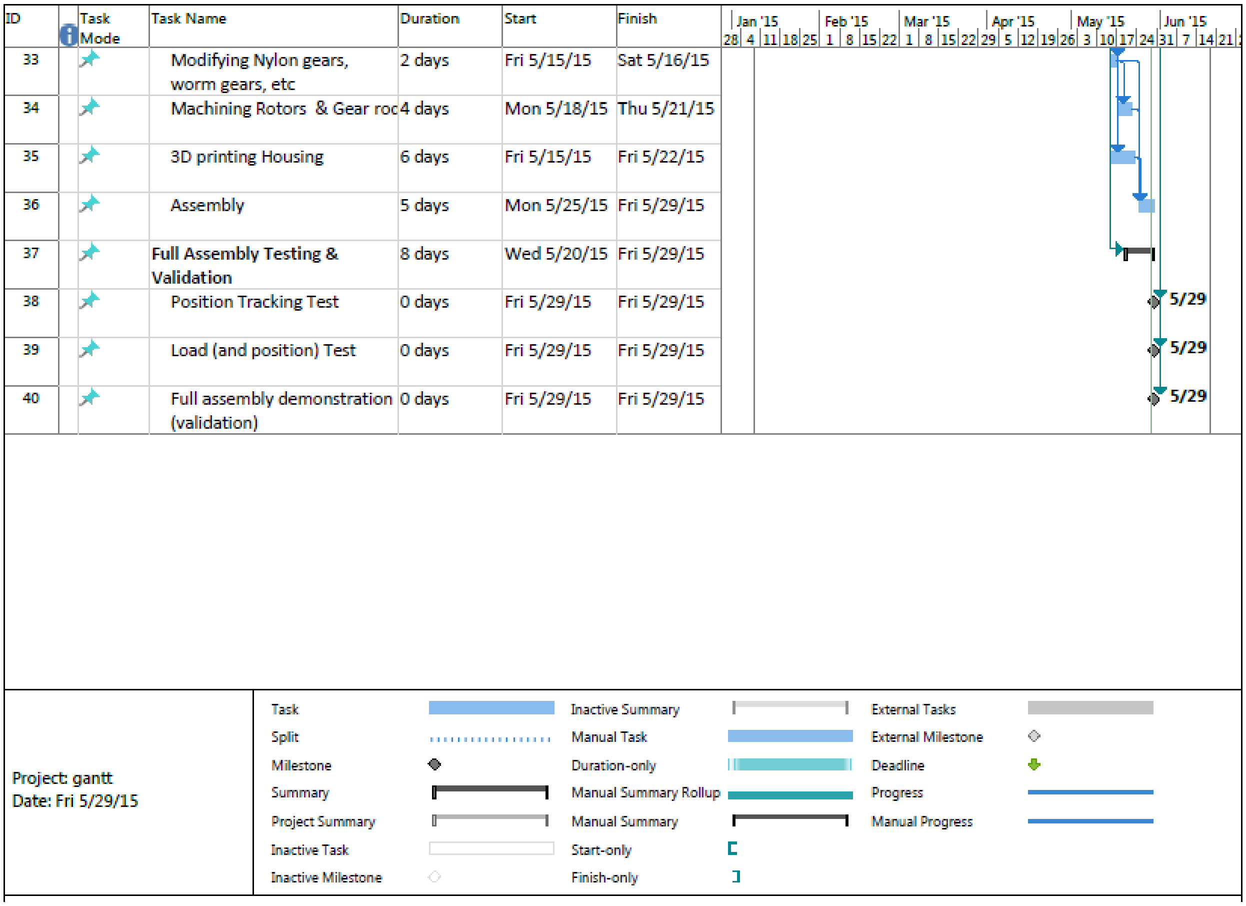
Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈┈┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⇩
Summary	▬	Manual Summary Rollup	▬	Progress	─
Project Summary	▬	Manual Summary	▬	Manual Progress	─
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		



ID	ℹ	Task Mode	Task Name	Duration	Start	Finish
22			CAD Full Assembly	30 days	Tue 4/7/15	Mon 5/18/15
23		📌	Establish Full Assemply Procudure	5 days	Wed 5/20/15	Tue 5/26/15
24		📌	**Programming**	**46 days**	**Fri 3/27/15**	**Fri 5/29/15**
25	✓	📌	Establish Dynamics Equations	5 days	Fri 3/27/15	Thu 4/2/15
26	✓	📌	Code for Basic Components (Microcontroller, Motor Driver, Rough Sensor)	10 days	Mon 3/30/15	Fri 4/10/15
27	✓	📌	Integration of Simple Code into Module Code	5 days	Sat 4/11/15	Thu 4/16/15
28		📌	Debugging & Testing (Basic Components)	22 days	Fri 4/17/15	Mon 5/18/15
29		📌	Code for Additional Components (Module Communication)	23 days	Wed 4/29/15	Fri 5/29/15
30		📌	Debugging & Testing (Additional Components)	6 days	Fri 5/22/15	Fri 5/29/15
31		📌	**Constuction**	**25 days**	**Mon 4/27/15**	**Fri 5/29/15**
32		📌	Order Parts & Coordinate with Facilities	29 days	Mon 4/6/15	Thu 5/14/15

Project: gantt
Date: Fri 5/29/15

Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⇩
Summary	▬	Manual Summary Rollup	▬	Progress	─
Project Summary	▬	Manual Summary	▬	Manual Progress	─
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		



ID	Task Mode	Task Name	Duration	Start	Finish
33	📌	Modifying Nylon gears, worm gears, etc	2 days	Fri 5/15/15	Sat 5/16/15
34	📌	Machining Rotors & Gear roc	4 days	Mon 5/18/15	Thu 5/21/15
35	📌	3D printing Housing	6 days	Fri 5/15/15	Fri 5/22/15
36	📌	Assembly	5 days	Mon 5/25/15	Fri 5/29/15
37	📌	**Full Assembly Testing & Validation**	8 days	Wed 5/20/15	Fri 5/29/15
38	📌	Position Tracking Test	0 days	Fri 5/29/15	Fri 5/29/15
39	📌	Load (and position) Test	0 days	Fri 5/29/15	Fri 5/29/15
40	📌	Full assembly demonstration (validation)	0 days	Fri 5/29/15	Fri 5/29/15

Project: gantt
Date: Fri 5/29/15

Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⇩
Summary	▬	Manual Summary Rollup	▬	Progress	▬
Project Summary	▬	Manual Summary	▬	Manual Progress	▬
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		

BIBLIOGRAPHY

1. M. Ansari, A. Amir, M. Hoque, "Microcontroller Based Robotic Arm," *International Conference on Electrical Engineering and Information & Communication Technology*, 2014.

2. S. Joergensen, C. Schou, O. Madsen, "Developing modular manufacturing architectures," *5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, Vol. 5, pp. 55-60, 2014.

3. B. Linke, P. Harris, M. Zhang, "Development of Desktop Multipurpose Grinding Machine for Educational Purposes," 2014.

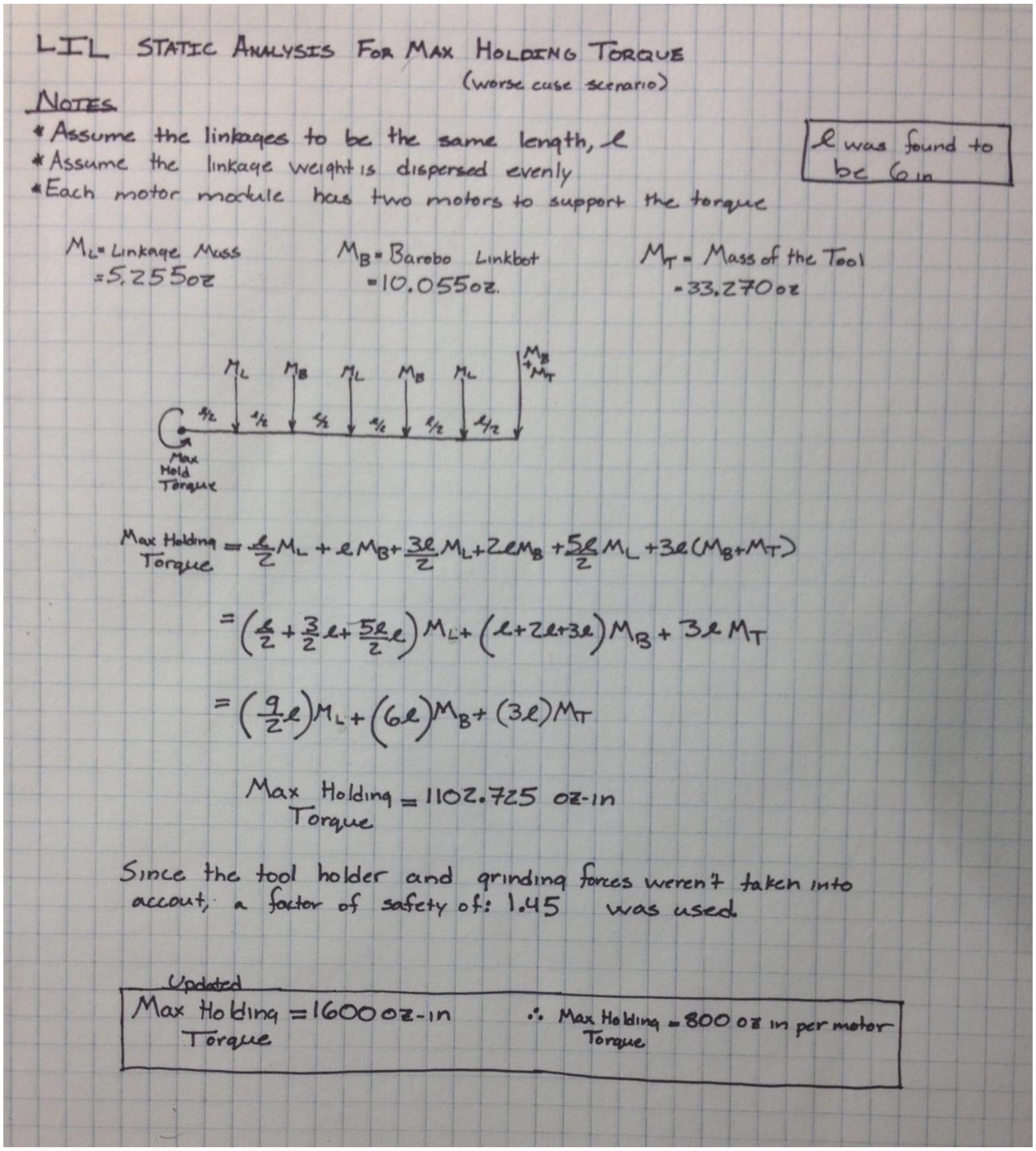
4. R. Budynas, J. Nisbett, 1st Initial. , *Shigley's Mechanical Engineering Design*, 10th ed., New York: McGraw Hill, 2015

5. R. Hibbeler, *Mechanics of Material*, 2nd ed., New York: McGraw Hill, 2014

Appendix I: DESIGN ANALYSIS

Torque Requirement for Loads

Provided that the system is a continuing project, a static analysis shows more linkages, a greater number of modules, and the heaviest tool available, in order to derive the maximum torque requirements for a base module. In total, 3 modules 3 linkages, and a tool were used in the calculation. These parameters are taken into account for all measurements following.



LIL STATIC ANALYSIS FOR MAX HOLDING TORQUE
(worse case scenario)

NOTES

- Assume the linkages to be the same length, ℓ
- Assume the linkage weight is dispersed evenly
- Each motor module has two motors to support the torque

ℓ was found to be 6 in

M_L = Linkage Mass = 5.255 oz

M_B = Barobo Linkbot = 10.055 oz.

M_T = Mass of the Tool = 33.270 oz

M_L
M_B
M_L
M_B
M_L
$M_B + M_T$
$\ell/2$
$\ell/2$
$\ell/2$
$\ell/2$
$\ell/2$
$\ell/2$
Max Hold Torque

$$\text{Max Holding Torque} = \frac{\ell}{2}M_L + \ell M_B + \frac{3\ell}{2}M_L + 2\ell M_B + \frac{5\ell}{2}M_L + 3\ell(M_B + M_T)$$

$$= \left(\frac{\ell}{2} + \frac{3}{2}\ell + \frac{5\ell}{2}\ell\right)M_L + \left(\ell + 2\ell + 3\ell\right)M_B + 3\ell M_T$$

$$= \left(\frac{9}{2}\ell\right)M_L + \left(6\ell\right)M_B + (3\ell)M_T$$

Max Holding Torque = 1102.725 oz-in

Since the tool holder and grinding forces weren't taken into accout, a factor of safety of: 1.45 was used.

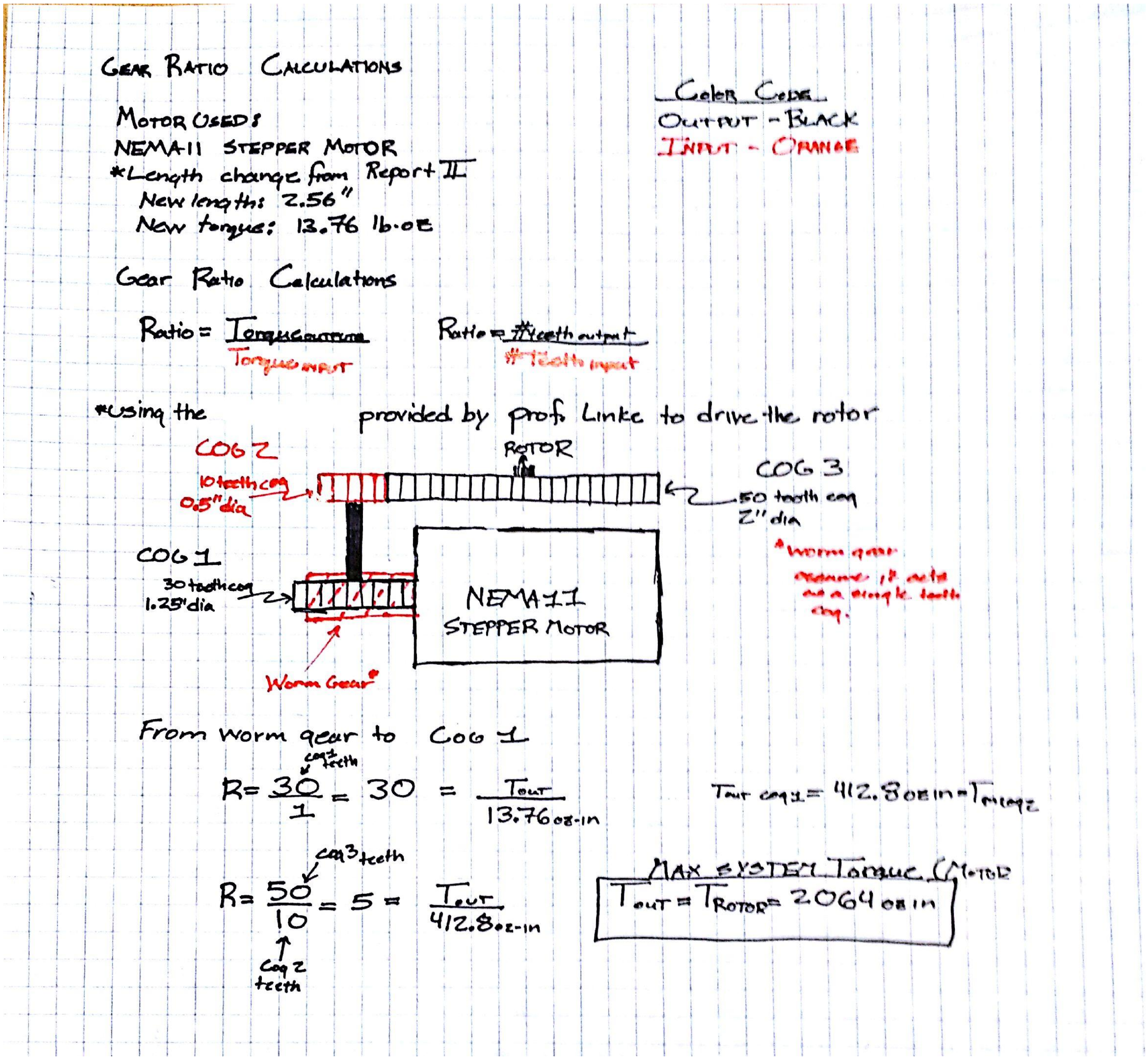
Updated

Max Holding Torque = 1600 oz-in ∴ Max Holding Torque = 800 oz in per motor

Figure 18: *Statics calculation for module torque requirements*

Gear Ratio

The gearbox comprises of 2 gear subcomponents: the worm drive and the regular spur gear set. The worm drive has a 30:1 gear ratio. From there, the worm gear drives and transfers power to the spur gear set up via a shaft. The connected spur gear set set features a 5:1 gear ratio. All together, the entire gear box provides a gear ratio of 150:1 (30/1 * 5/1). Figure 18 displays a more detailed analysis of the gear box calculations.



GEAR RATIO CALCULATIONS

Color Code
OUTPUT - BLACK
INPUT - ORANGE

MOTOR USED:
NEMA11 STEPPER MOTOR
*Length change from Report II
New length: 2.56"
New torque: 13.76 lb-oz

Gear Ratio Calculations

$$\text{Ratio} = \frac{\text{Torque}_{\text{output}}}{\text{Torque}_{\text{INPUT}}} \qquad \text{Ratio} = \frac{\#\text{teeth output}}{\#\text{Teeth input}}$$

*using the provided by prof. Linke to drive the rotor

From worm gear to Cog 1

$$R = \frac{\overset{\text{cog 1 teeth}}{30}}{1} = 30 = \frac{T_{OUT}}{13.76\,\text{oz-in}}$$

$T_{out\ cog 1} = 412.8\,\text{oz in} = T_{in\ cog 2}$

$$R = \frac{\overset{\text{cog 3 teeth}}{50}}{\underset{\text{Cog 2 teeth}}{10}} = 5 = \frac{T_{OUT}}{412.8\,\text{oz-in}}$$

MAX SYSTEM Torque (Motor

$$\boxed{T_{OUT} = T_{ROTOR} = 2064\,\text{oz in}}$$

Figure 19: *Gear box torque ratio calculations*

Precision

The stated precision of the stepper motors were 1.8°. With a gear ratio of 150 to 1, this would give the output rotors a precision of 0.01° (i.e.   1.8° / 150). The uncertainty is thus 0.006°.

Rotor Torsion

In general, a fixed geometry was applied to the rotor shaft where it connects to the internal gear box, and a distributed torque of 50 lb-in (or 800 lb-in, Figure 17) was applied to the screw holes where the linkage would be attached.

With these worst loading scenarios (see Figure 17), the rotor FEA showed significant stress concentration at the base of the rotor shaft. However, the minimum FOS still exceeded the desired FOS of 1.45. Hence, calculations concluded that the rotors would not fail.

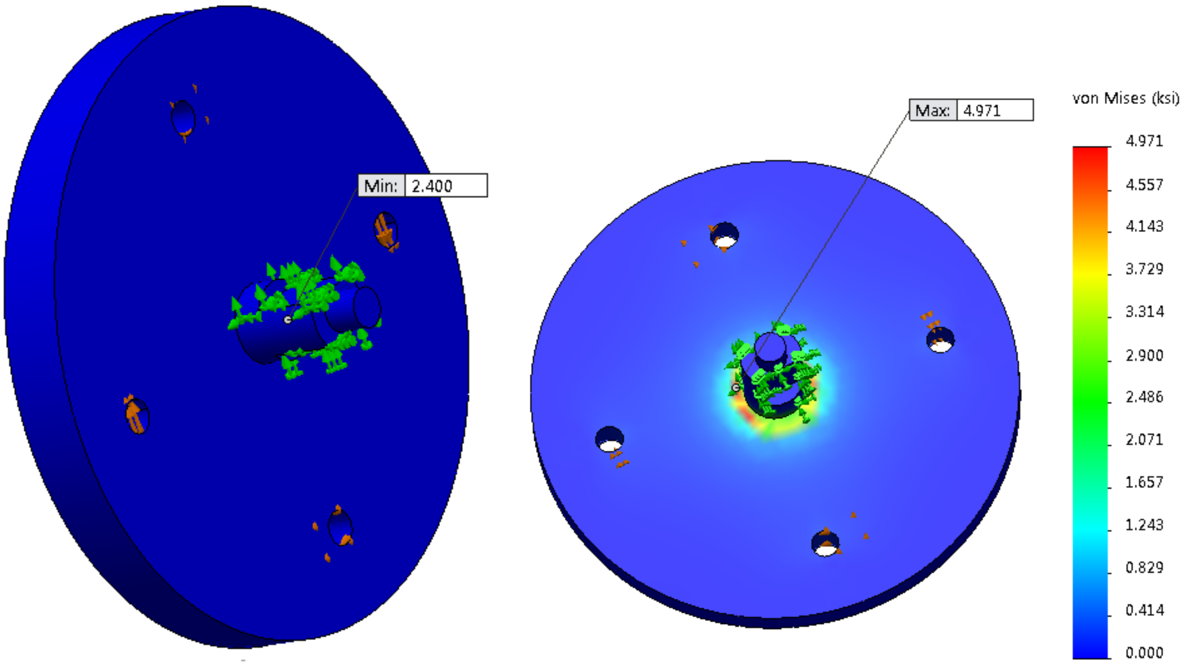


Figure 20:: *FEA analysis of rotors. Left is the minimum FOS. Right is the stress analysis*

Housing Stress Analysis

Applying the same constraints and loading as outlined in Figure 17, the housing FEA is shown in the figure below. An FOS calculation is not available for plastic materials in Solidworks, due to unavailability of a yield strength value. However, setting ultimate strength as the yield strength, minimum FOS was shown to be 60, much higher than design needs. In addition, the FEA below shows no significant stress concentration, with minimal concentration around the screw brackets.

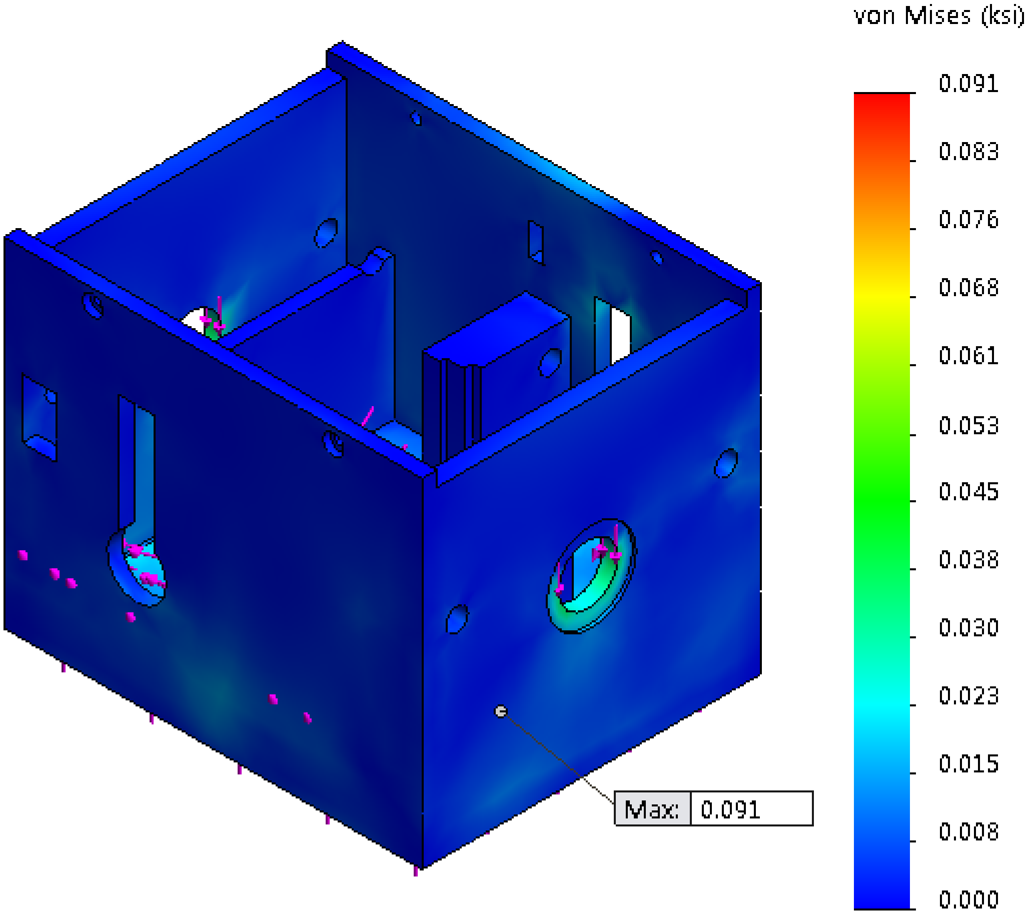


Figure 21: *Housing stress analysis*

Screws Stress Analysis

The screws hold the module-linkage joint in place, thus, they must endure the tension and shear forces from the weight of the linkages, modules, and tooling. Type 316 Stainless Steel Pan Head Phillips Machine Screw are selected based on the calculations below and SolidWorks FEA. The latter screws are incorporated in the entire linkage system. These screws are analyzed where they experience the most stress at the joints connecting the rotors to the linkages.

Maximum Load
Total weight: 3(5.255) + 3(10.055) + 33.270 = 79.2 oz-force
Force per screw: ~20 oz-force ⟶ 1.25 lb-force (for complete weight handled by tension -OR- shear)
Modified force per screw (tension, shear components): ~16 oz-force ⟶ 1 lb-force

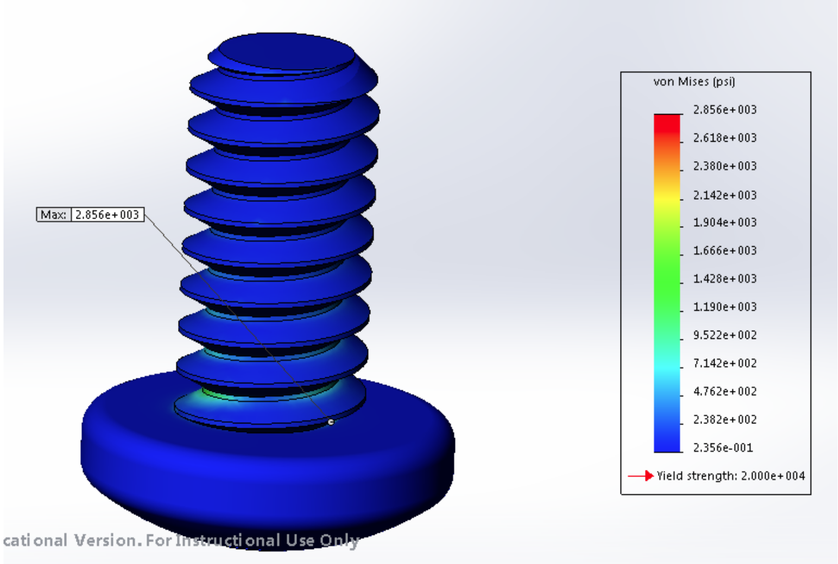


Figure 22: *Von Misses Stress distribution for screws*

Figure 21 illustrates the Von Misses stress distribution under 1 lb-force. The distribution notes a maximum stress of 2.856 kpsi at the base of the head of the screw which is below the yield strength of the screw, 20.00 kpsi. Indeed, these screws offer a factor of safety of 7.0 given its applications (Figure 22).

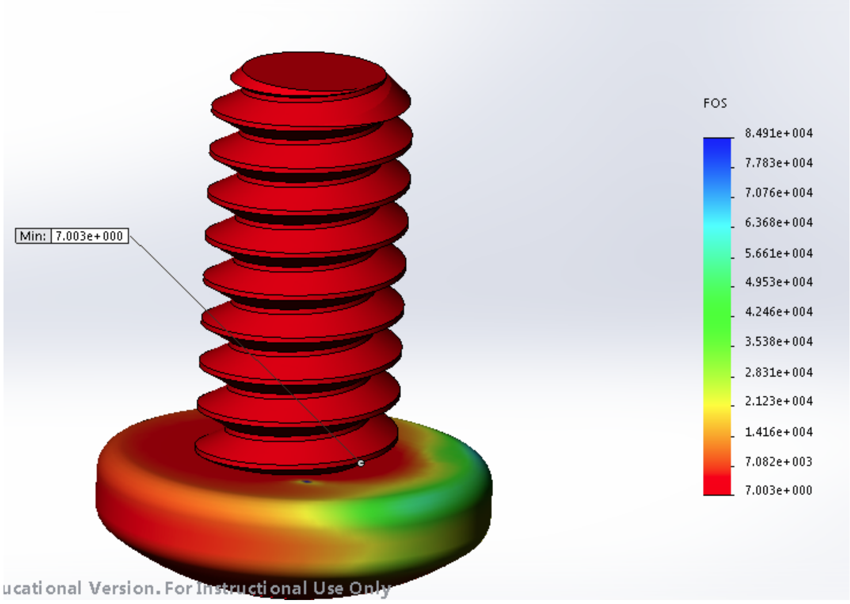


Figure 23: *FOS distribution for screws*

Testing

The motor module was tested for:

(1) Angular Precision and
(2) Loading Limit

Angular precision was tested with and without an applied load. The desired angle (dictated by a command signal) was tested against the angular displacement of the rotor to yield a percent error. Table 2 below shows that the percent error ranged from 0% to 1.48% on average, with commands for smaller angles creating the largest errors. With weight added to the motor module, it maintains the same precision trend as the no-load case. This trend holds until back-driving occurs. The source of this error most likely comes from flexing of the nylon gears. While metal gears would have been a better choice, budget and weight constraints made the nylon gears a better match.

Table 2: *Tabulated Data for No Load Precision Test*

Input Angle	Output Angle (AVG)	Percent Error
45°	44.33°	1.48%
90°	89.67°	0.37%
270°	271.17°	0.43%
360°	360°	0%

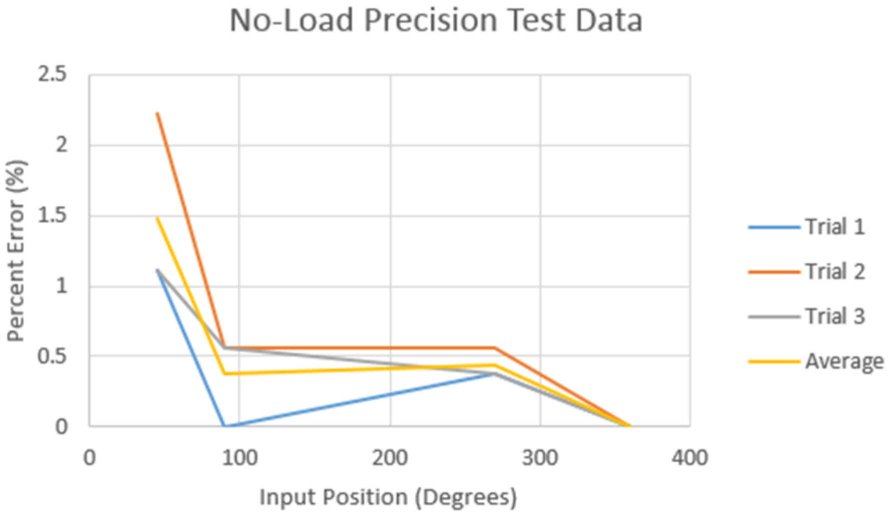


Figure 24: *Plot of No Load Precision Test Data*

The loading limit test was setup by adding weights to a cantilever attached to the motor module rotor until the module exhibits backdrive. It was found that backdrive occurred when 400 g was placed on an 8 inch linkage (50g). This give a holding torque of approximately 127 oz-in. This is better than the motor modules used in the previous year, however additional improvements can be made to get closer to the estimated 800 oz-in that the gearbox was designed for. Having the motor operate via micro-stepping greatly improved its performance.

Appendix II: USER MANUAL

Programming and Electronics

The motor module implements the use of an Arduino Mini ver 05 microcontroller board that does not come with a serial communication adapter (USB). The USB adapter is required in order to upload code from the user's laptop to the board. To circumvent this problem, the developer can use an Arduino UNO with its ATmega processor chip removed or an FTDI serial adapter to connect the USB wire. See circuit diagrams below.

All motor control code is developed in Arduino development environment and code is readily available. Contact project members via email (Numair Ahmed: mnahmed@ucdavis.edu or numairahmed1@gmail.com) for code zip file or copy & paste code from Appendix IV.

STOP

TIPS & WARNINGS:

- Ensure all wire connections are properly making contact where they should be - intermittant contact may damage motors and motor drivers.
- Make 24V power supply the LAST connection you make after ensuring correct connection of wires.
- Serial communication may take a minute or two to establish connection - please wait for user prompt to appear.

Circuit Setup with Arduino UNO for Serial Communication

Parts and Equipment:

- Arduino Mini 05 microcontroller
- Arduino UNO with ATmega chip removed
- USB cable for Arduino UNO
- 1 EasyDriver Motor Driver
- Breadboard
- 24V power source (adapter)
- Multimeter
- Jumper wires

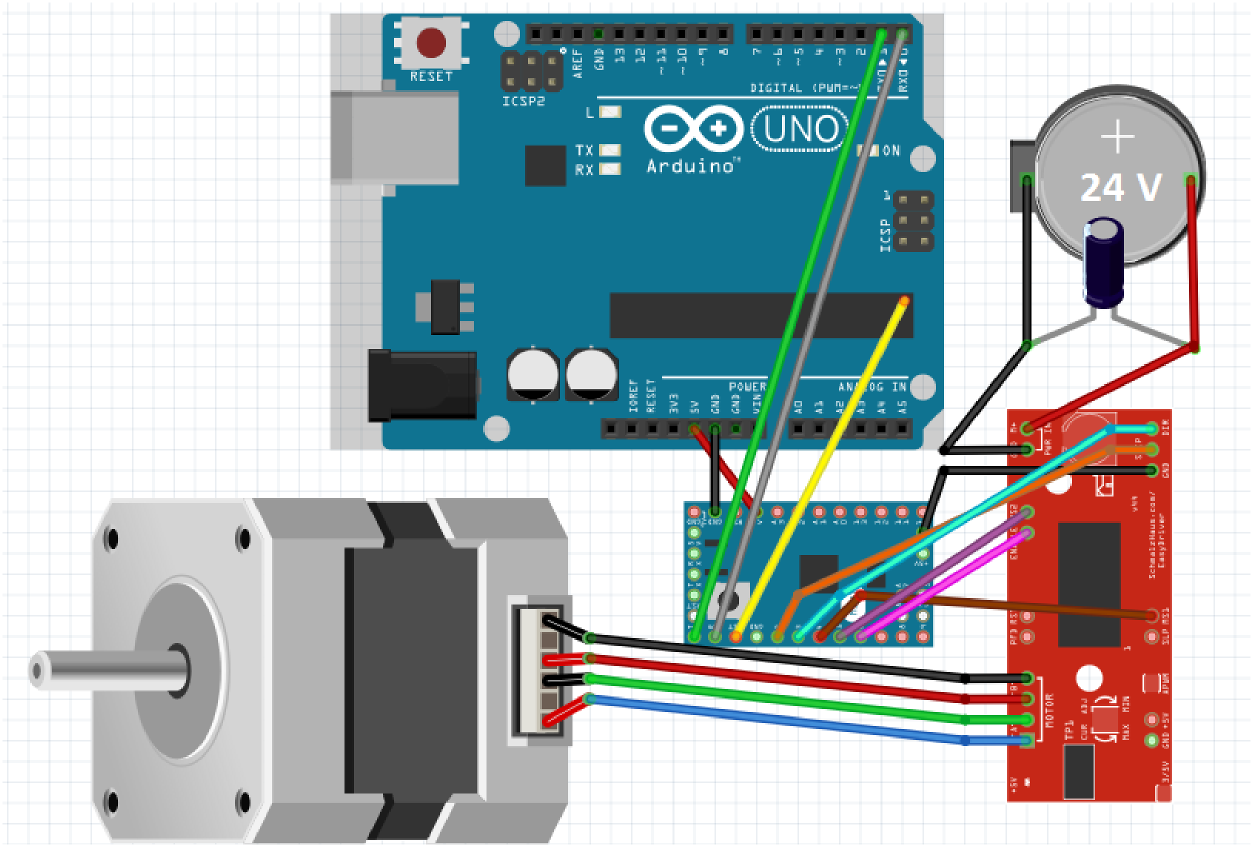


Figure 25: *circuit setup for Arduino UNO serial communication*

Circuit Setup with FTDI adapter for Serial Communication

Parts and Equipment:

- Arduino Mini 05 microcontroller
- FTDI adapter
- Mini USB cable for FTDI adapter
- 1 EasyDriver Motor Driver
- Breadboard
- 24V power source (adapter)
- Multimeter
- jumper wires

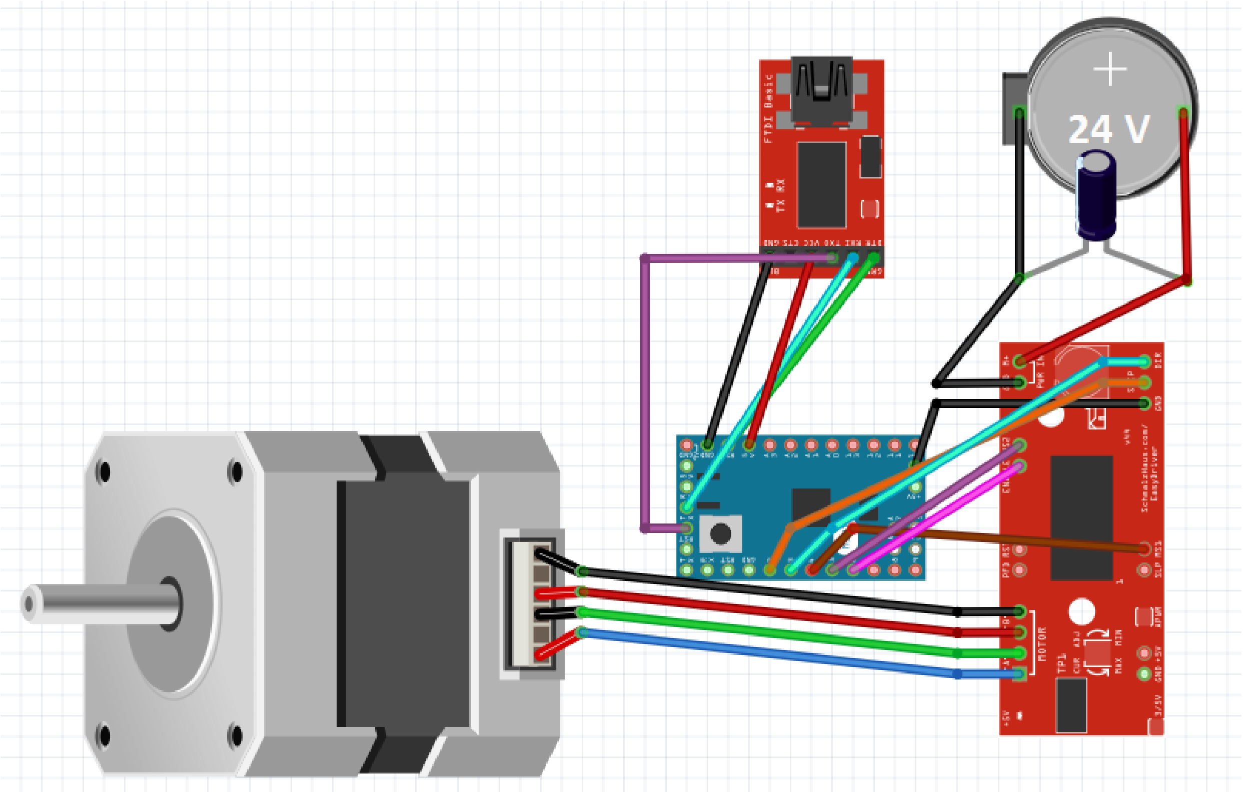


Figure 26: *circuit setup for FTDI adapter for serial communication*

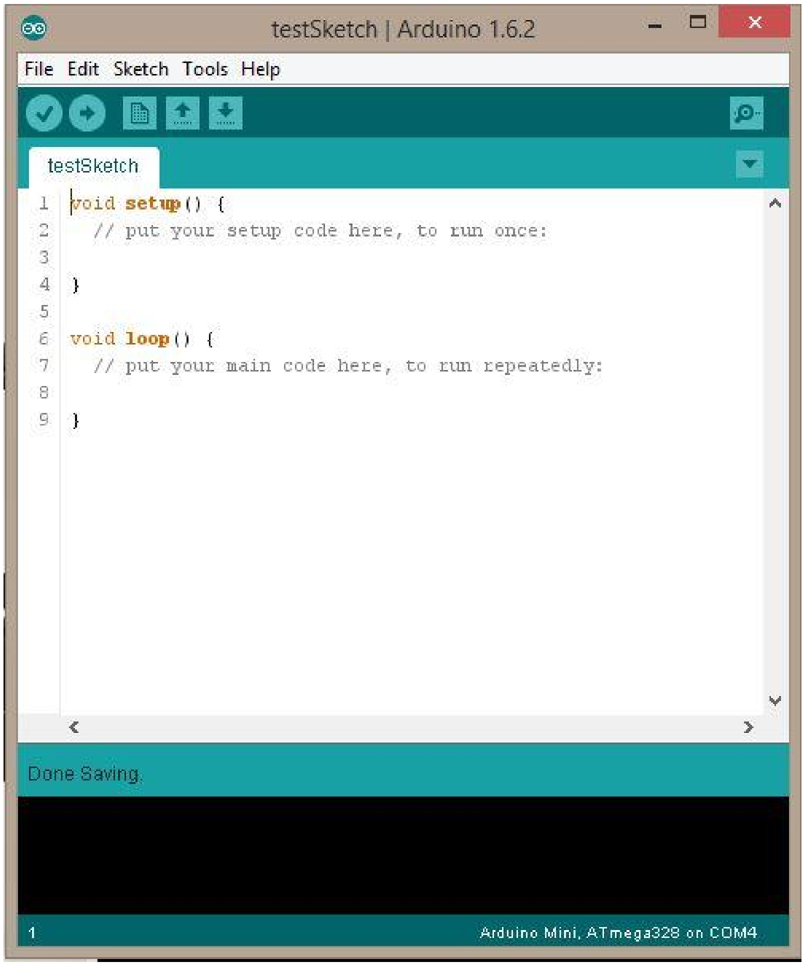
Software Controls (Code and Arduino IDE)

Parts and Equipment:

- Laptop computer
- Arduino IDE

1. Download and install Arduino Integrated Development Environment (IDE) from: http://www.arduino.cc/en/Main/Software

All software development is done in open-source Arduino environment. The program looks like this:



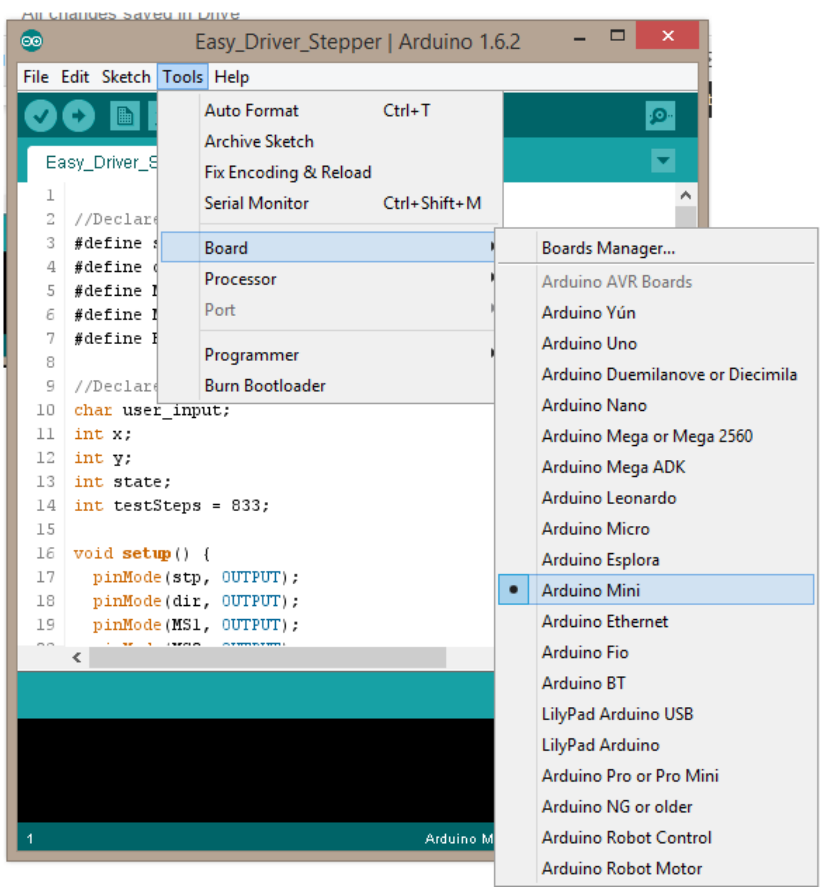
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Done Saving.

1 Arduino Mini, ATmega328 on COM4

Figure 27: *Arduino IDE*

2. Select Arduino Mini as microcontroller from Tools>Board>Arduino Mini

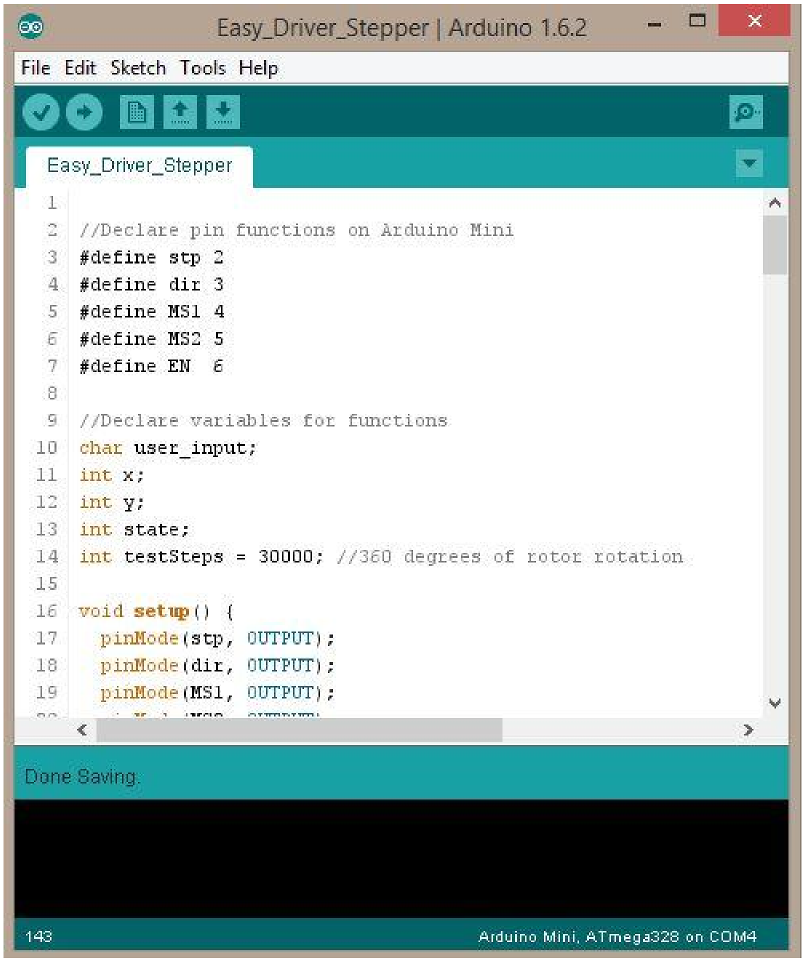


```

//Declare
#define s
#define d
#define M
#define M
#define E

//Declare
char user_input;
int x;
int y;
int state;
int testSteps = 833;

void setup() {
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(MS1, OUTPUT);
```

Figure 28: *Accesing Arduino Mini 05 via Tool tab*

3. Set up circuit as shown above, according to chosen use of Arduino UNO or FTDI adapter. See circuit diagrams above.

4. Obtain Arduino code zip file for modules from project members (contact Numair Ahmed: mnahmed@ucdavis.edu or numairahmed1@gmail.com) via email or copy & paste code from Appendix. Install Arduino code found in Appendix (or obtained from project members) into IDE library by following steps listed here: http://www.arduino.cc/en/Hacking/Libraries
Your Arduino IDE should have code that looks something like this:


Easy_Driver_Stepper | Arduino 1.6.2
File Edit Sketch Tools Help
Easy_Driver_Stepper

```

//Declare pin functions on Arduino Mini
#define stp 2
#define dir 3
#define MS1 4
#define MS2 5
#define EN  6

//Declare variables for functions
char user_input;
int x;
int y;
int state;
int testSteps = 30000; //360 degrees of rotor rotation

void setup() {
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(MS1, OUTPUT);
```

```
Done Saving.
```

```
143                                   Arduino Mini, ATmega328 on COM4
```

Figure 29: *Arduino IDE with module motor control code. Notice variable "testSteps"*

5. Connect USB cable to Serial communication interface of choice (Arduino UNO or FTDI adapter). Your Arduino Mini 05 should now be powered, indicated by LEDs lighting up.

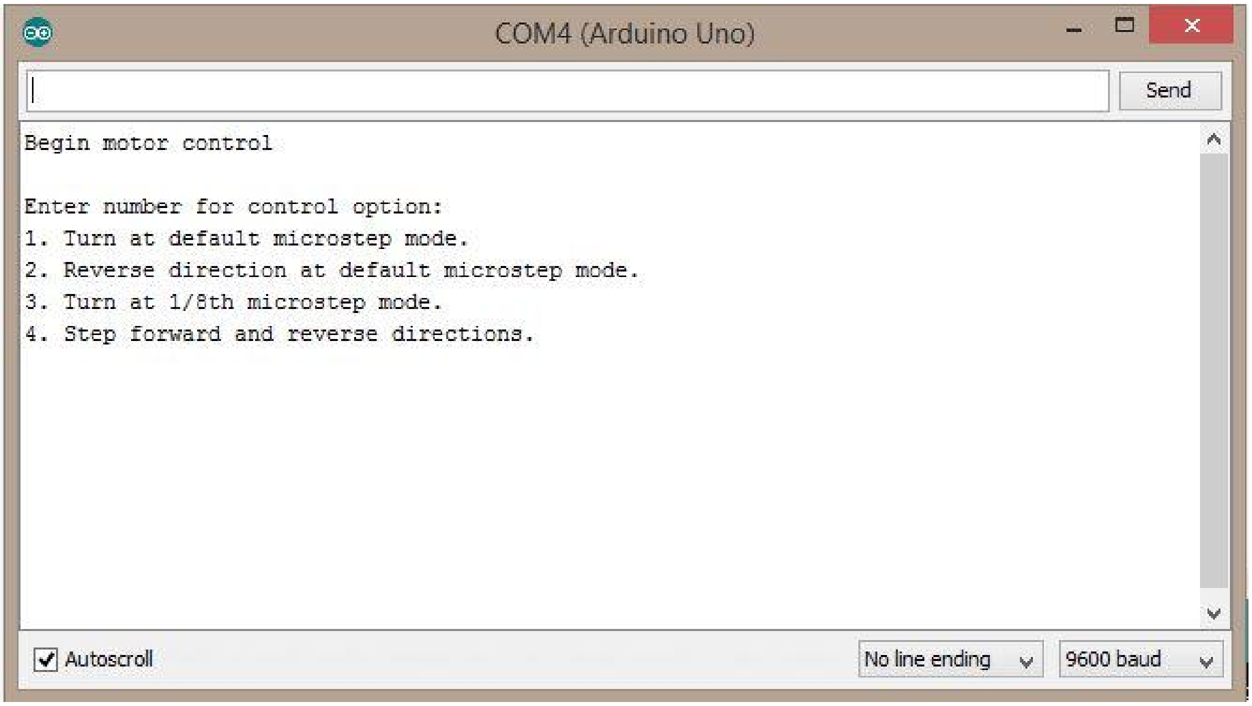
6. The software controls require two user inputs, 1) the desired angle of rotation for outer aluminum rotor and 2) the mode of rotation (forward, backward, microstep forward, forward & backward). At the current level of development, the desired angle of rotation can only be hard-coded into the software controls - future development will allow user to submit this input via a graphical user interface (GUI). The GUI currently allows the user to submit an input only to specify mode of rotation mentioned above.

Hard-code in desired angle of rotation in the form of motor steps into the variable "testSteps" according to table below:

Table 3: *Angle to testSteps input guide*

Desired Angle (°)	testSteps =
5	416
10	833
25	2083
45	3750
90	7500
180	15000
270	22500
360	30000

7. Upload Arduino code to microcontroller by clicking the UPLOAD button in the IDE toolbar and open the Arduino IDE Serial Monitor by clicking . This will allow for user to communicate with the module and submit desired mode of rotation.



```
Begin motor control

Enter number for control option:
1. Turn at default microstep mode.
2. Reverse direction at default microstep mode.
3. Turn at 1/8th microstep mode.
4. Step forward and reverse directions.
```

☑ Autoscroll

No line ending

9600 baud

Figure 30: *Serial Monitor for user input*

Appendix III: Manufacturing and Assembly

PARTS MANUFACTURING

Equipment

Manufacturing of the motor module parts will require the following machinery:

- Bandsaw
- Lathe
- Mill
- Grinder
- FDM 3D printer (available at UC Davis TEAM fabrication facility)

PLA Housing

Manufacturing Procedures:

1. Save Solidworks CAD model of housing as "**.stl**" file
2. Contact TEAM printing facility for 3D printing

Rotors

Materials:

- solid stock aluminum rod, 2.5" diameter, 3" length plus 1" additional length per rotor
- #28 drill bit
- 10-24 tap

Manufacturing Procedures:

3. Using a lathe and 3 jaw chuck, face the aluminum rod.
4. Turn 0.31" diameter feature at depth of 0.583" from original faced surface
5. Turn 0.156" diameter feature at depth of 0.14" from original faced surface
6. Turn 2.4" diameter feature to depth of .9" to 1" from original faced surface
7. Bandsaw roughly 0.9" to 1" depth of total material from original faced surface
8. Using a lathe and 5/16" collet, face rotor down until 0.25" thick.
9. Using mill and #28 drill, drill four holes 0.85" from center.
10. Tap holes with 10-24 tap
11. Mill 3/16" key geometry with ¼" end mill.

Gear Rod

Materials:

- Carbon Steel rod, 3/16" diameter, 3' length
- Hacksaw
- Dremel

Manufacturing Procedures:

1. Using the hacksaw, cut 1.9" pieces off the carbon steel rod.

2. Using the grinder, get rid of the rough edges on the rod.
3. Using the dremel, create a flat edge on the rod where the worm gear set screw will sit.

50T Gear Modification

Materials:

- Pre-made 50-tooth gear
- Gear guide-plate

Manufacturing Procedures:

1. Using gear guide-plate (available at EFL) insert gear into largest gear insert.
2. With edge finder, set proper coordinates on mill
3. Using 3/16" end mill, position the end mill to the center of the gear
4. Mill through gear and move end mill 0.063" along one axis ("X" or "Y")
5. Recenter end mill, and move 0.063" in opposite direction, along same axis.

10T Gear Modification

Materials:

- Pre-made 10-tooth gear
- Gear Guide-Plate
- 3/16 drill bit

Manufacturing Procedures:

1. Using gear guide-plate (available at EFL) insert gear into smallest gear insert.
2. With edge finder, set proper coordinates on mill
3. Using a 3/16" drill bit drill out the center of the gear so that the bore is 3/16.

Worm Modification

Materials:

- Pre-made worm from Boston Gear
- 5 mm Reamer
- 11/32 Collet
- 2-56 Tap

Manufacturing Procedures:

1. Using the lathe and 11/32 collet, expand the center hole to 5 mm (0.1966") using the reamer.
2. Using the 2-56 tap, tap the pre-made hole perpendicular to the center hole.

Worm-Gear Modification

Materials:

- Pre-made worm-gear from Boston Gear
- 51 Drill Bit

- 2-56 Tap

Manufacturing Procedures:

1. Using a lathe and 3 jaw chuck turn the smooth diameter of the gear to 0.35”.
2. Using a mill, drill out a 0.061“ diameter hole using a 51 drill bit.
3. Tap the 0.061“ diameter hole using a 2-56 tap.

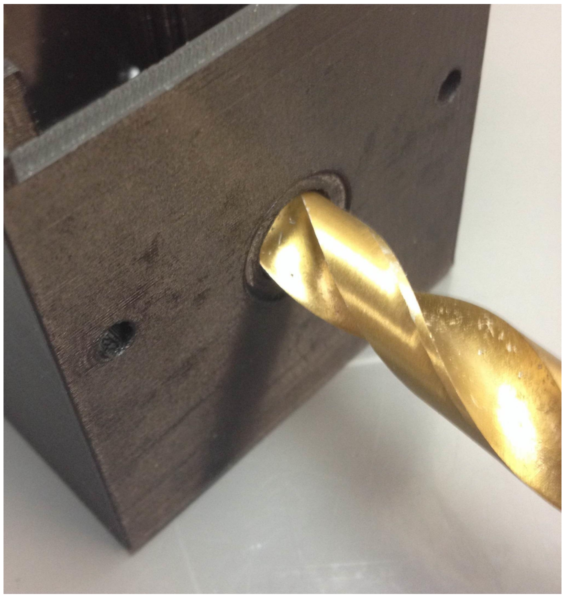
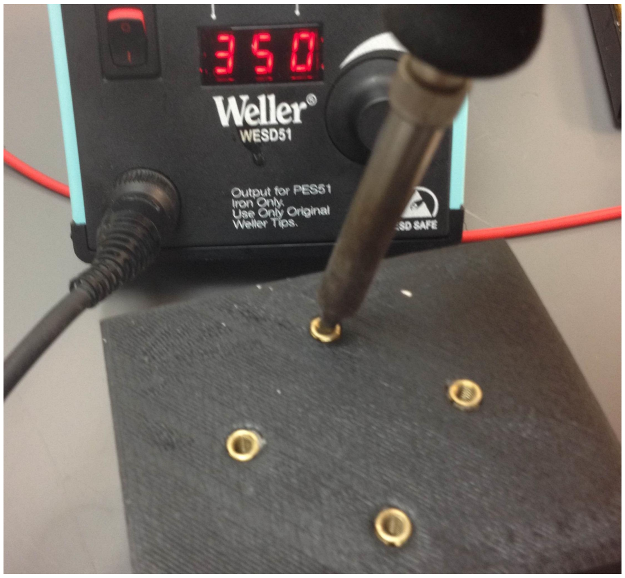
ASSEMBLY

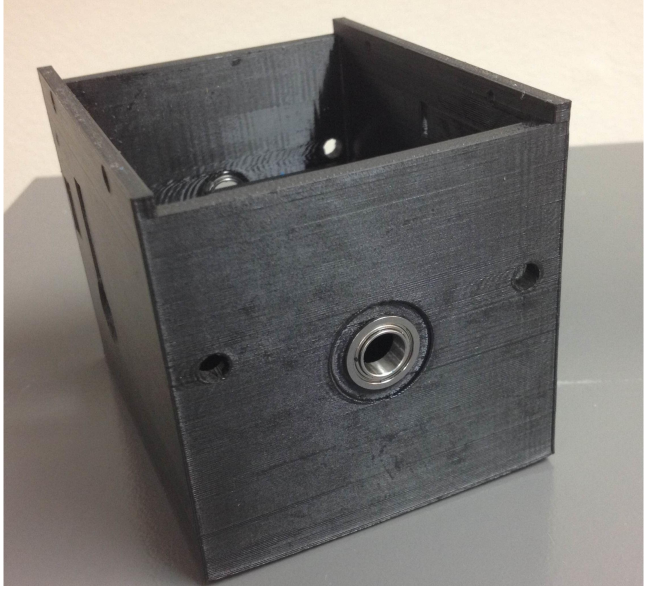
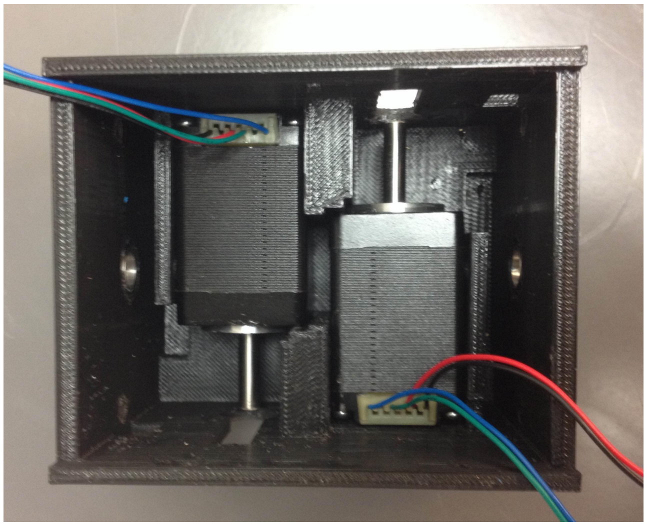
Equipment and Materials:

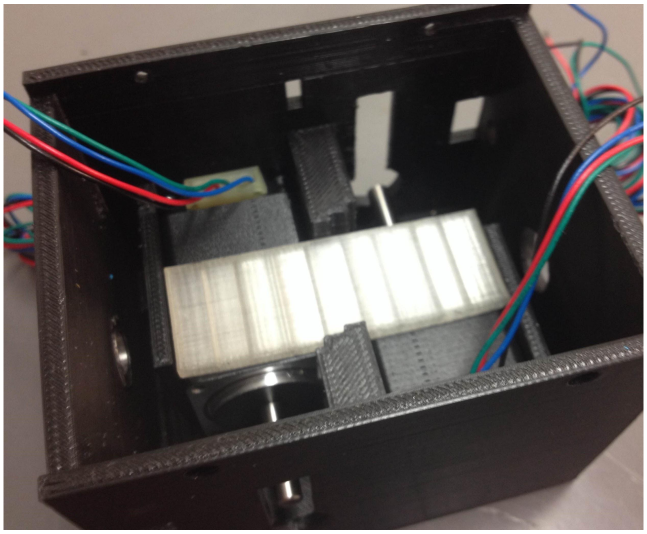
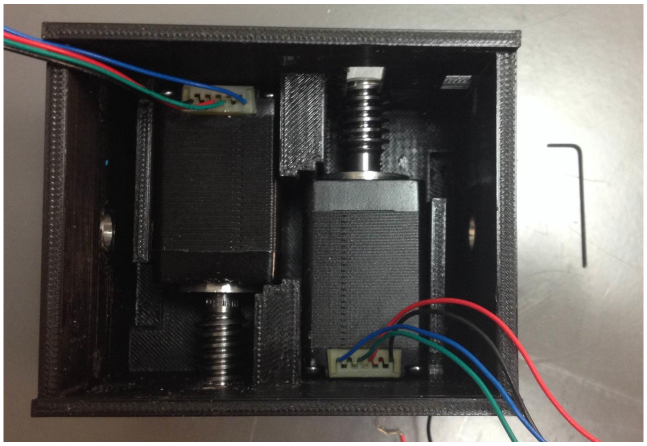
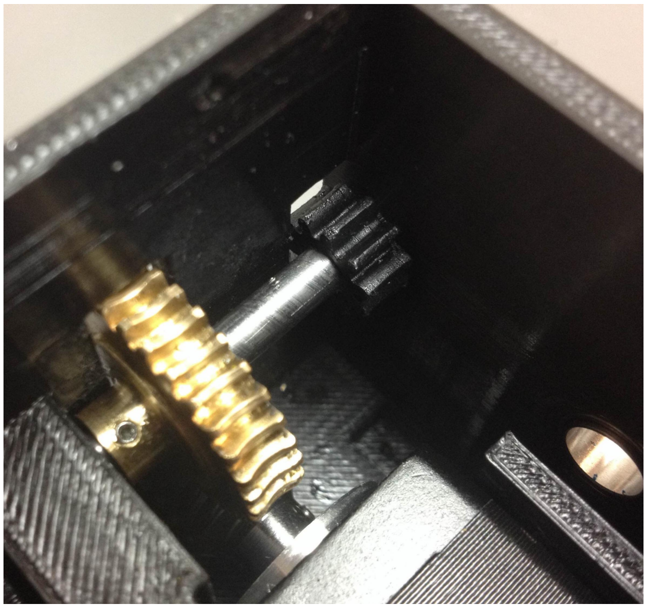
- Phillips screwdriver
- Slotted screwdriver
- Needle-nose pliers
- Epoxy (plastics-glue)

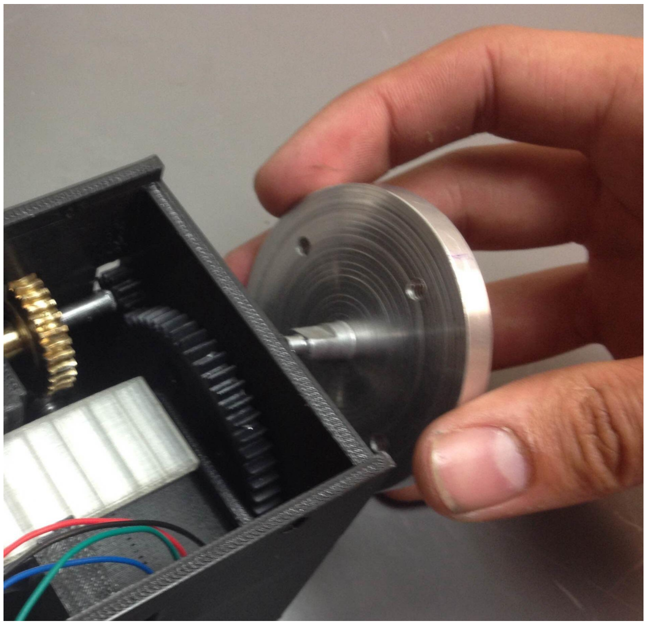
Instructions:

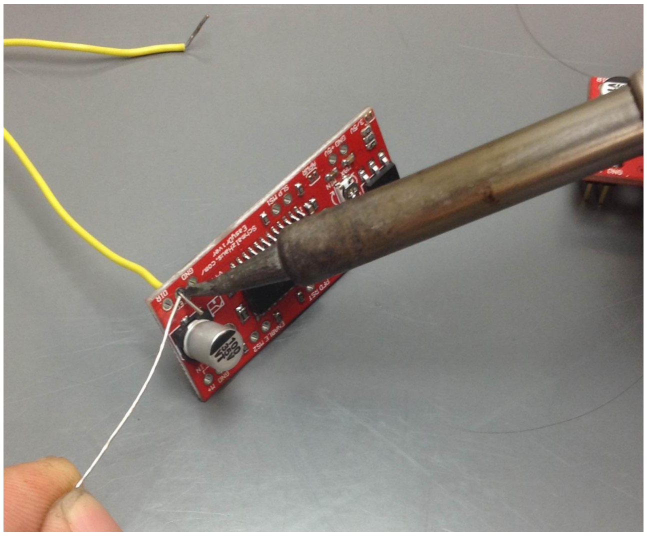
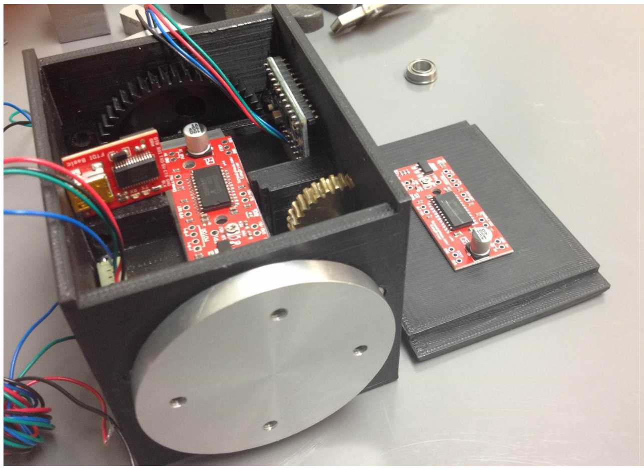
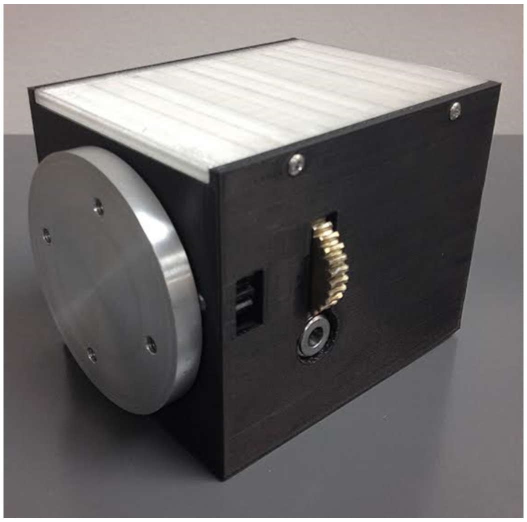
Table 4: *Assembly steps with pictures*

#	Step	Image
1	Clean up holes on the prints by using a 3/16ths reamer (or drill bit) on the holes for the gear shaft and a ½ reamer (or drill bit) on the holes for the bearing. Note: To ensure the holes of the gear shaft are aligned, ream the holes with motor/worm drive assembly.	
2	Using a soldering iron at 350°C, install the threaded brass inserts on the housing and housing top.	 350 Weller® WESD51 Output for PES51 Iron Only. Use Only Original Weller Tips. ESD SAFE

3	Install the bearings on both sides of the housing	
4	Install the two motors into the housing using the inner walls as a guide.	

5	Install the T-Motor Housing between the two stepper motors.	
6	Install the worm onto the motor shaft using the housing hole as a guide. Make sure to tighten the set screw using a 0.035 hex key.	
7	Install the worm gear 10 tooth gear and gear shaft by feeding the 10 tooth gear first followed by the worm gear. A light tap with a hammer may be needed to push the knurled part of the shaft into the housing.	

8	Test to see if the current assembly works on each side. Any misalignment may seize up the gear box.	STOP
9	Install the rotor and 50 tooth gear for both sides	
10	Test to see if the fully-assembled gearbox works for each side.	STOP

11	Solder the electrical components using solid wire. See Figure 25 for general testing setup with one motor driver.	
12	Install the electronics into the motor housing as position in the picture. For a more detailed picture, see Figure 2.. Apply hot glue on needed surfaces. Take care with fragile electronics. Make sure none of the wires can be tangled up with the gearbox.	
13	Run a final check to see if module correctly operates. Congratulations, you are done!	

Appendix IV: Arduino Code for Motor Control

```
//Declare pin functions on Arduino Mini
#define stp 2
#define dir 3
#define MS1 4
#define MS2 5
#define EN  6

//Declare variables for functions
char user_input;
int x;
int y;
int state;
int testSteps = 30000; //360 degrees of rotor rotation

void setup() {
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(MS1, OUTPUT);
  pinMode(MS2, OUTPUT);
  pinMode(EN, OUTPUT);
  resetEDPins(); //Set step, direction, microstep and enable pins to default states
  Serial.begin(9600); //Open Serial connection for debugging
  Serial.println("Begin motor control");
  Serial.println();
  //Print function list for user selection
  Serial.println("Enter number for control option:");
  Serial.println("1. Turn at default microstep mode.");
  Serial.println("2. Reverse direction at default microstep mode.");
  Serial.println("3. Turn at 1/8th microstep mode.");
  Serial.println("4. Step forward and reverse directions.");
  Serial.println();
}

//Main loop
void loop() {
  while(Serial.available()){
    user_input = Serial.read(); //Read user input and trigger appropriate function
    digitalWrite(EN, LOW); //Pull enable pin low to allow motor control
    if (user_input =='1')
    {
      StepForwardDefault();
    }
    else if(user_input =='2')
    {
     ReverseStepDefault();
    }
    else if(user_input =='3')
    {
     SmallStepMode();
    }
    else if(user_input =='4')
    {
     ForwardBackwardStep();
    }
    else
    {
     Serial.println("Invalid option entered.");
    }
    resetEDPins();
  }
}

//Reset Easy Driver pins to default states
void resetEDPins()
{
  digitalWrite(stp, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(MS1, LOW);
  digitalWrite(MS2, LOW);
  digitalWrite(EN, HIGH);
}

//Default microstep mode function
void StepForwardDefault()
{
  Serial.println("Moving forward at default step mode.");
  digitalWrite(dir, LOW); //Pull direction pin low to move "forward"
  for(x= 1; x<testSteps; x++)  //Loop the forward stepping enough times for motion to be visible
  {
    digitalWrite(stp,HIGH); //Trigger one step forward
    delay(1);
    digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
    delay(1);
  }
  Serial.println("Enter new option");
  Serial.println();
}

//Reverse default microstep mode function
void ReverseStepDefault()
{
  Serial.println("Moving in reverse at default step mode.");
  digitalWrite(dir, HIGH); //Pull direction pin high to move in "reverse"
  for(x= 1; x<testSteps; x++)  //Loop the stepping enough times for motion to be visible
  {
    digitalWrite(stp,HIGH); //Trigger one step
    delay(1);
    digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
    delay(1);
  }
  Serial.println("Enter new option");
  Serial.println();
}

// 1/8th microstep foward mode function
void SmallStepMode()
{
  Serial.println("Stepping at 1/8th microstep mode.");
  digitalWrite(dir, LOW); //Pull direction pin low to move "forward"
  digitalWrite(MS1, HIGH); //Pull MS1, and MS2 high to set logic to 1/8th microstep resolution
  digitalWrite(MS2, HIGH);
  for(x= 1; x<5000; x++)  //Loop the forward stepping enough times for motion to be visible
```

```
 {
  digitalWrite(stp,HIGH); //Trigger one step forward
  delay(1);
  digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
  delay(1);
 }
 Serial.println("Enter new option");
 Serial.println();
}

//Forward/reverse stepping function
void ForwardBackwardStep()
{
 Serial.println("Alternate between stepping forward and reverse.");
 for(x= 1; x<5; x++)  //Loop the forward stepping enough times for motion to be visible
 {
  //Read direction pin state and change it
  state=digitalRead(dir);
  if(state == HIGH)
  {
   digitalWrite(dir, LOW);
  }
  else if(state ==LOW)
  {
   digitalWrite(dir,HIGH);
  }

  for(y=1; y<1000; y++)
  {
   digitalWrite(stp,HIGH); //Trigger one step
   delay(1);
   digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
   delay(1);
  }
 }
 Serial.println("Enter new option:");
 Serial.println();
}
```

Appendix V: Alternative Motor Housing Design

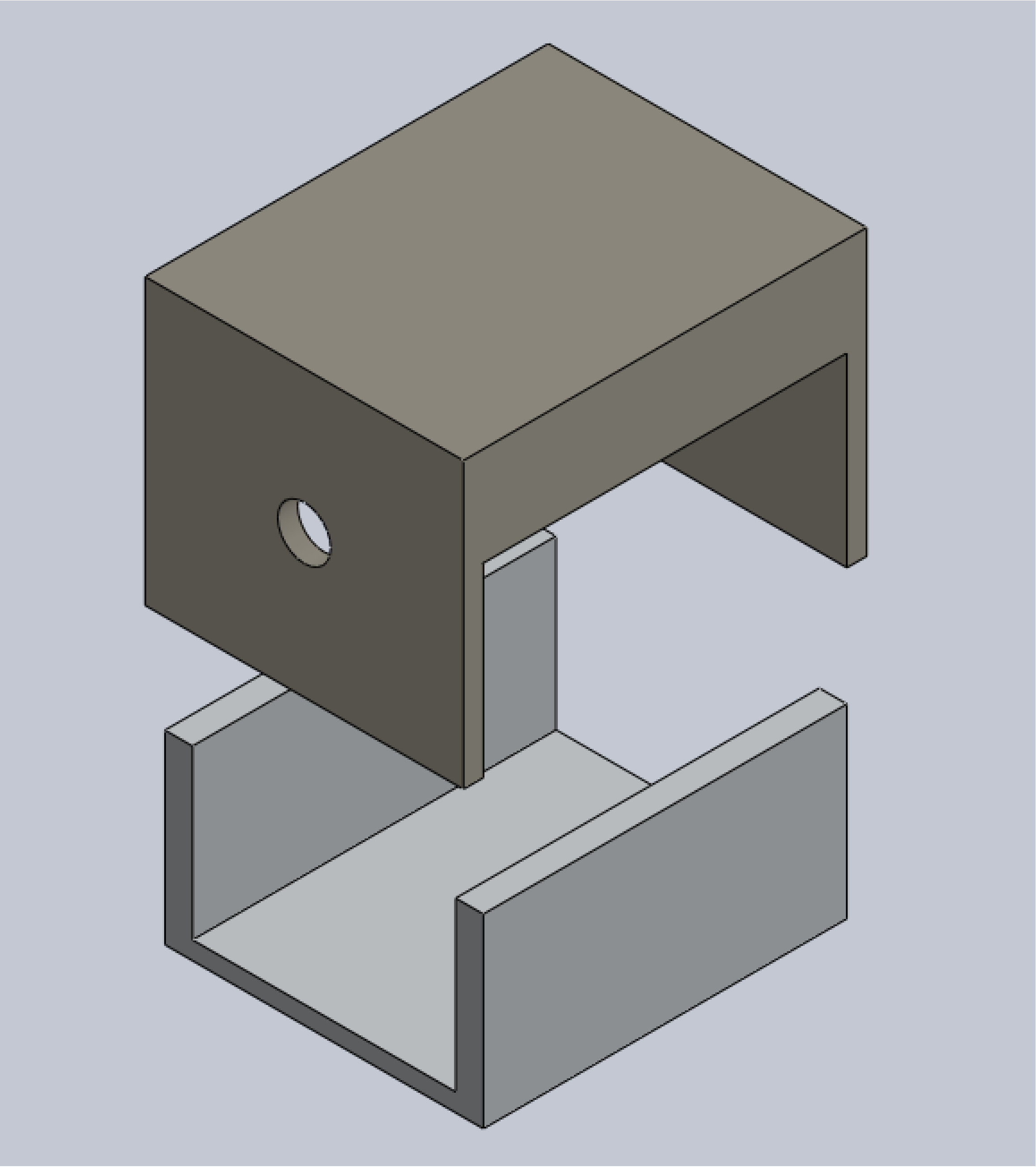


Figure 31: *Alternative Motor Housing CAD model with the PLA plastic in beige and aluminum U-channel in grey*