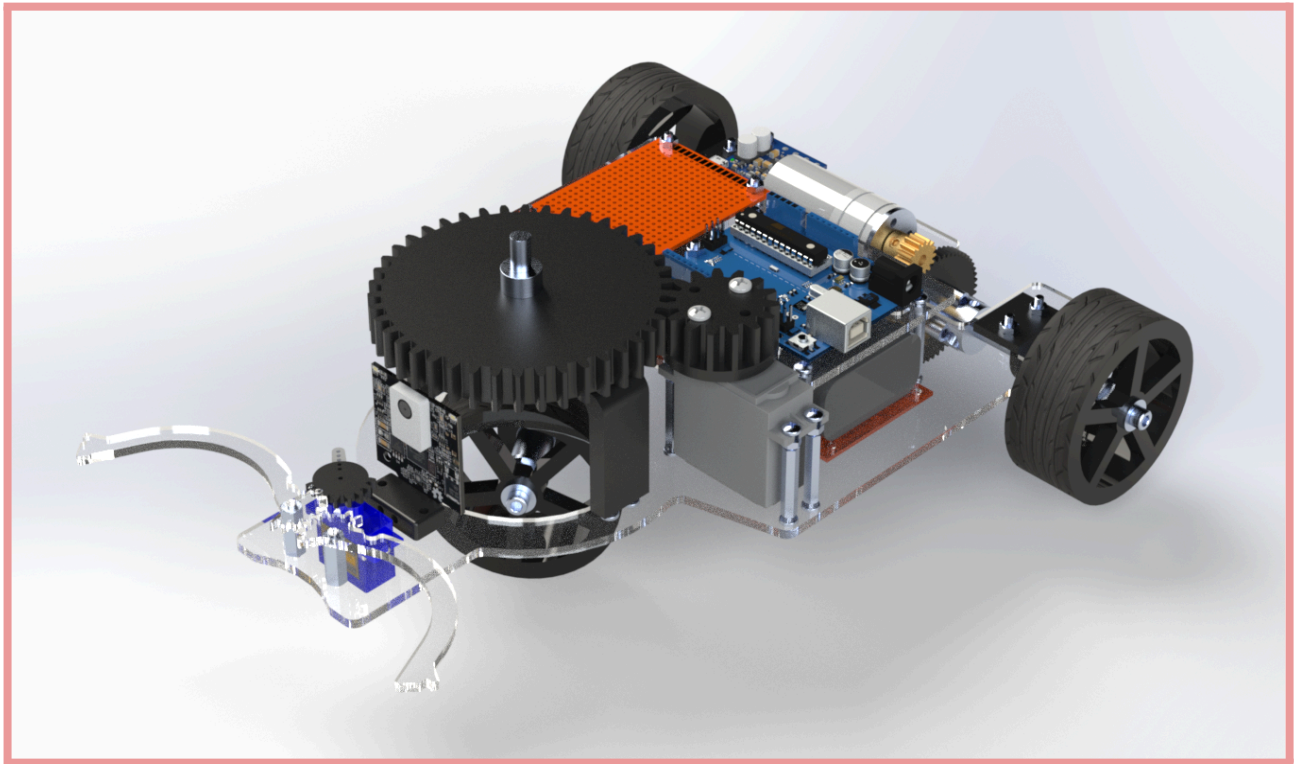


Design Report

N.O.O.O.M.S

Lab 5 Wednesday 1PM



Autonomous Rover Designers

Team Captain = Nicole Schaeffer

Documentation/Presentation = Nicole Schaeffer, Ozzy Sanchez-Aldana

Design/Solidworks/Research = Matthew Scott, Owen Fang, Ozzy Sanchez-Aldana

Budget = Shane Parslow, Nicole Schaeffer

Fabrication/Assembly = Matthew Scott, Octavio Adame

Coding = Shane Parslow

Photographers = Octavio Adame

ENGR 7B: Introduction to Engineering

University of California, Irvine

Winter 2022

TABLE OF CONTENTS

EXECUTIVE SUMMARY...	...1
-----------------------------	------

PROBLEM DEFINITION...	...2-3
I) Introduction	
II) Technical Review / Background	
III) Design Requirements	

DESIGN DESCRIPTION...	...4-9
I) Summary of Design	
II) Design Details	
III) Wiring Diagram	
IV) Algorithm Design	

ACTION ITEM REPORT...	...10
I) Task Assignment	
II) Gantt Chart	

EVALUATION...	...11
I) Calculations	
II) Test Plan	
III) Results & Discussion	

APPENDIX A: SOLIDWORKS DESIGN...	...12-17
---	----------

APPENDIX B: BILL OF MATERIALS...	...18
---	-------

APPENDIX C: ARDUINO AUTONOMOUS MAPPING CODE...	...19-20
---	----------

APPENDIX D: REFERENCES...	...21
----------------------------------	-------

EXECUTIVE SUMMARY

We are Team n.o.o.o.m.s., and we have assembled this quarter in order to create an autonomous rover with a fully functional claw mechanism and steering system. Our original goal was to have the rover navigate a random path using photoresistors and Arduino coding. It also needed to pick up a cup by using a PixyCam to distinguish it by color and activate the claw mechanism.

Our first step of the process was designing the rover; figuring out the layout of all necessary components and how the wires would connect throughout. Then came the specific claw and steering designs. We originally had worked on the body first for simplicity's sake to visualize our limits, but then refocused on our custom claw and steering mechanisms instead, modifying the rover body to accommodate these sub-assemblies.

After the Solidworks design was greenlit by our advising lab staff, the fabrication team worked on laser-cutting and 3D printing the parts with assistance from the design team, and then moved forward with assembly.

During the fabrication process, we began developing the code in order to influence decisions made during design and fabrication. This ensured that the brain of the rover would run smoothly while the body was being designed to accommodate the necessary functions. We presented our milestone progress through A.I.Rs each week, both for our advisors and to gain a better understanding of technical presentations.

During testing, our rover was able to pass through the competition into the top ten, where it competed in the final challenge. Our team was overjoyed with the results. The rover successfully followed the path set by the instructional team and guided itself to the cup which the PixyCam recognized and activated the claw mechanism.

At the end of the project, we decided to apply our rover design to a practical real world purpose and dubbed it the Autonomous Mineral Extractor (A.M.E.). This is due to the rover's precise steering and sensory input which would easily allow it to detect precious minerals in enclosed spaces.

We have overcome many challenges along the way, from capitalizing on the few options during remote learning to delays in assembly and coding malfunctions. However, we have delivered on our goals to create an efficient, unique rover in both design and cost. This design report is intended to showcase our progress and the final results of the rover.

PROBLEM DEFINITION

I) Introduction

Our original objective was simple: create a rover that could pick up a cup on its own after navigating on a path. Achieving that objective was not as simple as we thought, and it sparked many questions: How would we create a precise enough steering that would keep up with a line detected by photoresistors? How would our claw work to pick up the cup? How would we get the steering and claw to work in sync on its own? Over the course of this quarter, we were able to answer all of these questions via the creation of A.M.E., here is a walk through of our problem-solving process in this report.

II) Technical Review / Background

In recent history, autonomous cars used to be a pipe dream of many people. Now? It's a reality. They aren't just used for transportation. They are used by kids in toys, by the military in combat, and even used by NASA in space. The possibilities with self-driving cars are unlimited and the future is looking bright.

Autonomous cars and rovers have been staples of the future for many years. In cartoons and futuristic tv shows, you often find these self-driving cars. The future is now with many tech companies becoming invested in autonomous cars. Companies like Tesla, Waymo, and Pony.ai are paving the way with their groundbreaking technology and innovation. These companies, especially Tesla, are revolutionizing the way we view self-driving cars. We aren't quite at full automation though. In late 2015, Tesla introduced its semi-autonomous autopilot feature. It's not at the level where it can safely navigate every situation and environment, but it is a massive step in the right direction (Noonan, 2021). This wouldn't have been possible without major breakthroughs in the past. The first major autonomous vehicle was actually a plane. In 1933, the Mechanical Mike autopilot was invented and is made possible through the use of gyroscopes, which are vital in autopilot nowadays in cars (WIRED, 2016).

Learning how to design rovers that are able to make the right decision autonomously is the first step in being able to build an autonomous car. Safety will always be the priority when designing self-driving cars, and by building off past inventions and by continuing to innovate the autonomous car world will only get safer. Manned cars will soon be a thing of the past with autonomous cars taking over.

III) Design Requirements

The chassis consists of a 12" by 6" rectangle of $\frac{1}{8}$ " polycarbonate with four separate $\frac{1}{2}$ " by $\frac{3}{4}$ " tabs on the side. We decided on the 12" by 6" base to best emulate the CyberRover's own dimensions (thus fulfilling a cosmetic purpose) while also providing enough space for electrical components. Four rectangular tabs are on the left and right sides of the base to provide a surface for the shell to attach to. We chose the $\frac{1}{8}$ " polycarbonate for its sturdy and lightweight properties, thus allowing us to carry all the necessary components without having to worry about overworking the motors.

Our 3D printed claw mechanism uses a continuous servo to actuate the motion of a rack and pinion, which closes and opens the claw. The maximum distance the claw is able to open is ~3.7" and is able to close with a ~1" gap. The claw structure also incorporates the Pixy onboard.

III) Design Requirements (continued)

The rover must have an easily accessible way of turning on and off the rover, which we added a key switch to turn on and off the rover.

No protruding sharp objects can appear, and if they are sharp, they must be filled or sanded down, as well as wires must be completely covered and insulated to avoid injury or rover malfunction.

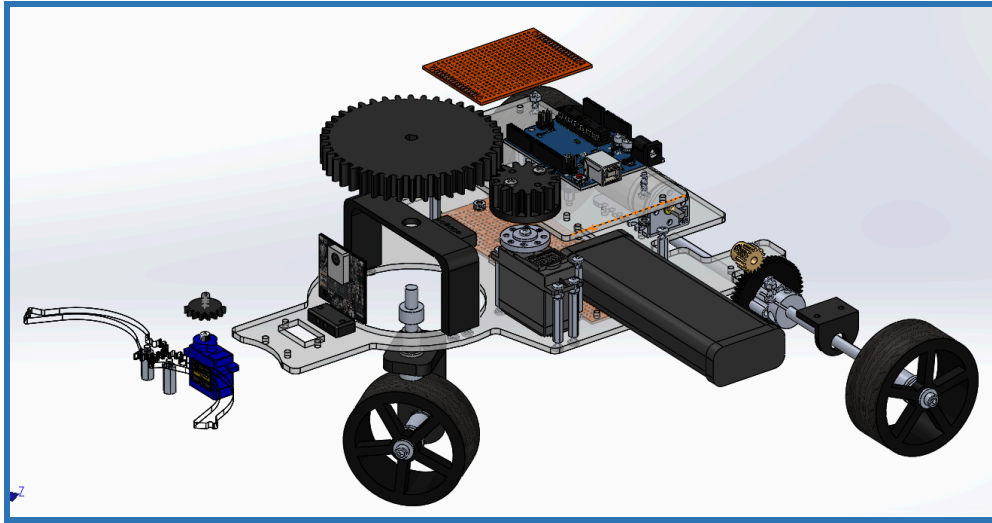
The total for the price of the rover must not exceed \$260, but since we did 3D printing, that set maxx is raised by \$50 setting the max price to \$330.

Lastly, one of the most important requirements is the rover must be able to follow the path of a black taped line on the ground, as well as design a mechanism, or claw to pick up a certain payload, which in our case is a cup.

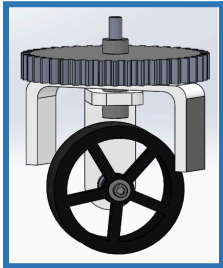
DESIGN DESCRIPTION

I) Summary of Design



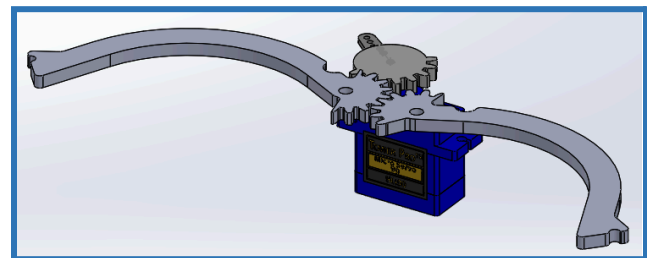


With the availability of 3D printing and laser cutting, we optimized our design to minimize hand fabrication for efficiency. The structure is composed of three sub-assemblies: the body (Figure 2), steering mechanism (Figure 3), and claw mechanism (Figure 4). For the body, all of our electronics are mounted to the main acrylic chassis, with the exception of the Arduino board. The Arduino board and a perf board are mounted to a separate acrylic piece that sits on standoffs above the battery. Two additional perf boards are also mounted underneath the chassis and below the battery. The rover remains open to minimize material costs and grant us ease of access to electrical components.



The steering mechanism consists of an arch-like chassis (Figure 3b), wheel holder (Figure 3a), and a large gear (Figure 3c). All of these pieces are 3D printed and are assembled independently then mounted to the rover via the arch. An additional, smaller gear (Figure 1b) is fitted to the large servo motor using the 25T servo hub. This creates a gear train where the large servo motor controls the steering mechanism.

The claw mechanism has two acrylic claw arms (Figure 4a) and a 3D printed gear (Figure 4b) that is attached via a servo horn to the small servo motor. Each claw arm has a gear end. The right claw arm forms a gear train with the small servo motor and the left claw arm. The motor exerts control on both by directly moving the right claw arm.



The rover structure has a maximum width of 9.6 inches and a maximum length of 13.5 inches with the claw fully extended frontwards. The mass of the rover equates to 3.5 pounds and the total cost is \$309.37 for all materials.

II) Design Details

Our rover structure has its center of mass at (X: -0.14, Y: 0.54, Z: 1.52) in inches.

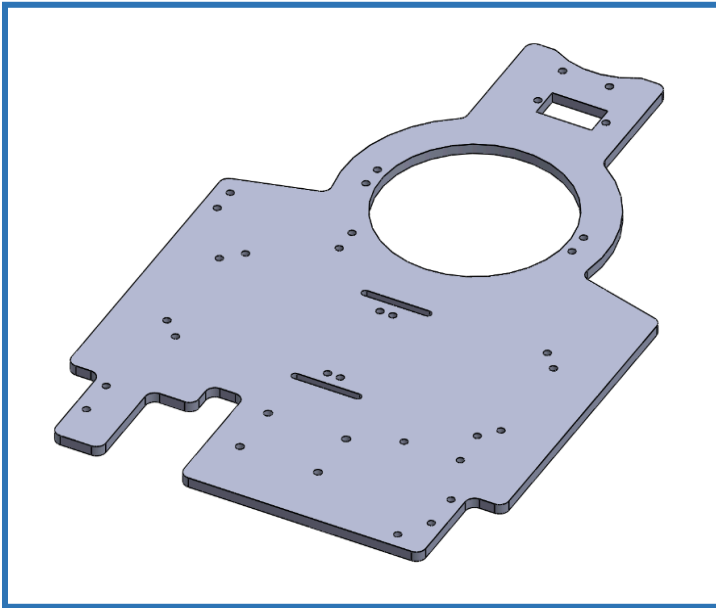
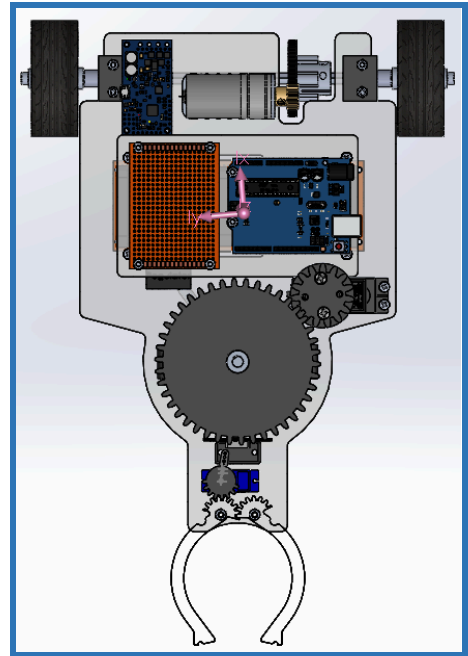
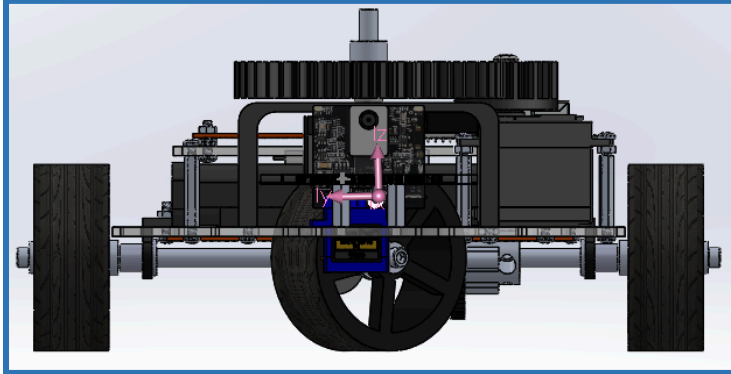
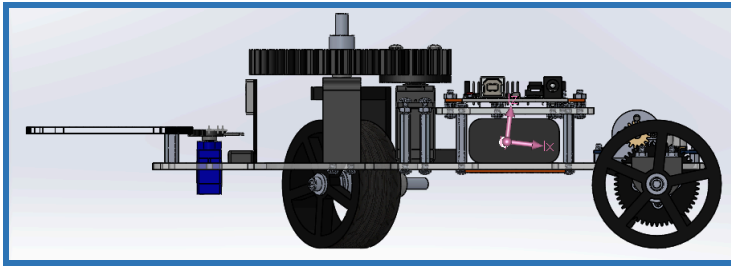
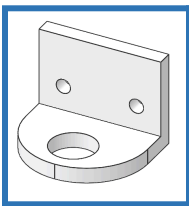
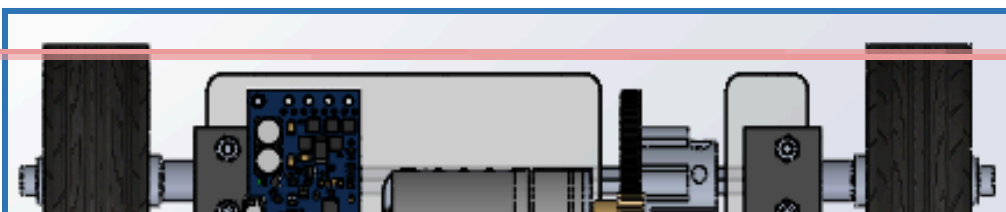


Figure 2b from Appendix A is the one piece chassis where all electronics and sub-assemblies are mounted. It is laser cut from $\frac{1}{8}$ " acrylic with rounded edges to prevent fracturing at the corners. There are several holes that are sized for M3 screws used to mount sub-assemblies and electronics to the chassis. The large hole at the front is space for the wheel on the steering mechanism to turn. The rear has spacing for the rear wheels, the 52T driven gear, and gear hub. The full body sub-assembly, Figure 2, is in Appendix A.



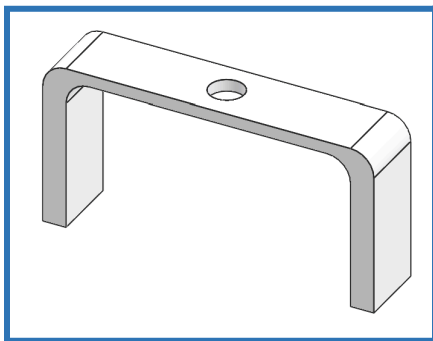
This piece, from Figure 2c, is one of the rear-wheel holders. There are two for the rear-wheel propulsion system. It is 3D printed from ABS with a 0.38" hole to house the bearing that the $\frac{1}{4}$ " by 8" d-shaft axle runs through. There are also two holes for M3 screws to mount the piece to the main chassis board. These can be seen on the illustration below as black pieces on each end of the chassis board next to the wheels:

II) Design Details (continued)

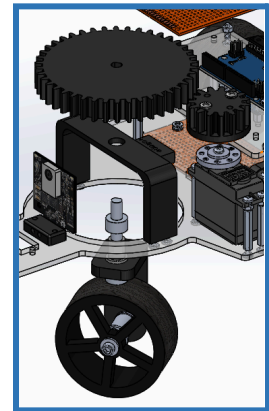


The rear wheel system uses 65 mm wheels, which is the same size used in our steering mechanism. The 8" long ¼" d-shaft axle runs underneath the chassis board through the bearings housed in the rear wheel holder pieces. The DC motor and motor controller are mounted on top of the chassis board using M3 screws. The DC motor is first attached to a motor mount that is actually in contact with the chassis. The 16T pinion gear on the DC motor meshes with the 52T driven gear fixed to the axle with the d-bore barrel hub. To achieve an optimal gear mesh, we utilized this equation to find the optimal distance between the centers of the 16T and 52T gears:

$$\frac{1}{2} \left(\frac{\text{Number of Teeth for Gear 1} + \text{Number of Teeth for Gear 2}}{\text{Diametral Pitch}} \right) = \frac{1}{2} \left(\frac{16 + 52}{32} \right) = 1.0625 \text{ in.}$$

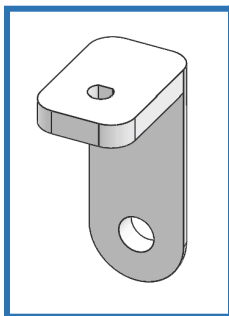


The steering chassis, from Figure 3b, is used as the base for the steering mechanism. It also attaches to the rover chassis using M3 screws on the bottom pillars, fixing the steering mechanism to the rover. It is 3D printed from ABS and has a hole at the top of the arch for a bearing. A 2" d-shaft runs through the bearing and serves as a pivot for the wheel. The wheel moves with the large gear

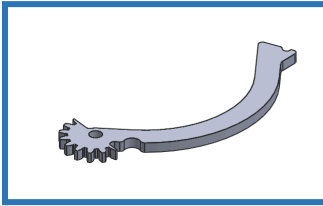


mounted on top controlled by the large servo.

The wheel for the steering system is connected using the Steering Wheel Holder: Figure 3a. The top hole is where the 2" d-shaft that runs through the top of the arch-like steering chassis, and connects to the large gear on top. The 2" d-shaft is held in place using ¼" shaft collars on both ends. The larger hole on the side stretching downwards is for another bearing. The second 2" d-shaft runs through that bearing and is fixed inside the wheel mount using set screws. The other end is secured with the same type of shaft collar. The 65 mm wheel is then fixed to the wheel mount and the steering system is complete. The large hole cutout in the rover chassis permits the wheel to freely pivot a full 360 degrees.

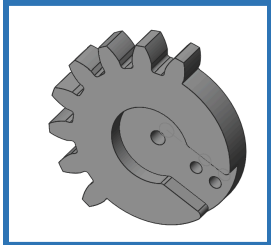


II) Design Details (continued)



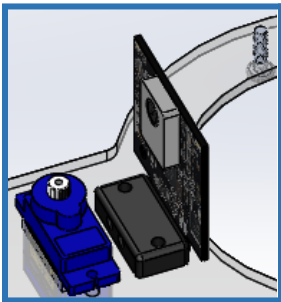
The last sub-assembly in the claw mechanism.

The arm from Figure 4a is laser cut from 1/8" acrylic with a hole slightly larger than an M3 screw. This allows it to be mounted on top of a standoff while also having space to pivot freely. The gear end is meshed with that of the other claw arm with a 1:1 ratio.

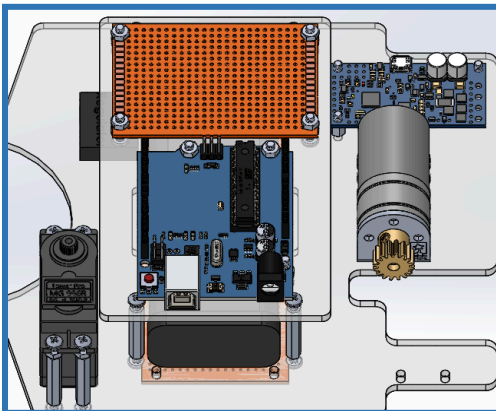


This gear from Figure 4b has a space for the servo horn to be fixed to the underside. The servo horn is attached to the smaller servo motor, which exerts control on the gear. The standoffs supporting the claw arms are positioned close to the small servo motor for this piece to create a gear train. The gear connects with the end of the right claw arm and operates the entire mechanism to open and close.

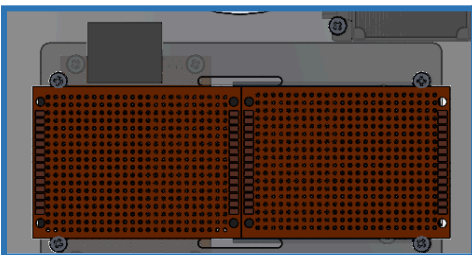
All electronics are directly mounted to the rover chassis using M3 screws for easy removal.



The PixyCam is positioned directly behind the claw mechanism and underneath the large gear of the steering mechanism. Using the black piece in the illustration below 3D printed from ABS. The top holes are sized for M3 screws and the PixyCam is fixed to the side using M3 screws. The back of the PixyCam is where the port for connecting to the Arduino is located. The connection wires are looped around the hole cutout for the steering mechanism to avoid obstruction.



The Arduino is mounted above the battery on an acrylic platform on standoffs, fixed with M3 screws. The perf board is where all connections are soldered, including the voltage regulator, servo motors, motor controller, LEDs, photoresistors, and the battery. The DC motor is connected to the motor controller. The battery is secured using velcro straps that loop through slits in the chassis to fasten it, see Figure 2b. The standoffs for the acrylic platform above are also positioned to prevent it from moving side to side.



This is the underside of the chassis where the battery is. These two additional perf boards are where the LEDs and photos resistors are soldered and connected to Arduino pins. A photo resistor is placed on each side to determine where the rover has gone off course (left or right).

III) Wiring Diagram

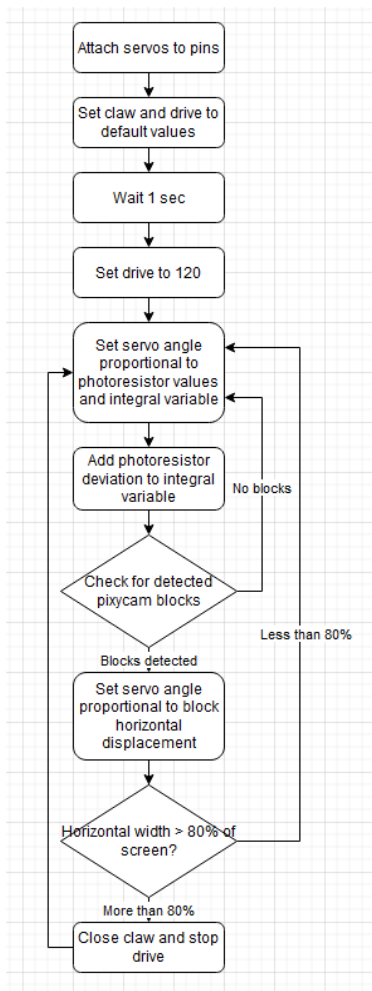


IV) Algorithm Design

The final Arduino program implements separate line-following and cup-seeking algorithms. Cup-seeking occurs as soon as the cup is identified by the Pixycam. Line-following occurs in all other cases.

Line-following consists of a PI (proportional integral) control loop. The integral term was added to attempt to solve the issue of proportional response decreasing to zero after failing to sufficiently correct the rover's course. The theory was that the integral term would provide 'memory' that the rover was off-course. A derivative term was also implemented, but was not used. The photoresistors ramped up quickly enough that the proportional response was sufficient. A derivative term would be unnecessary, and provide little additional information to the control loop.

When the Pixycam detects the cup, the line-following control loop output is overwritten by the cup-seeking control loop. The cup-seeking control loop is simply a proportional term acting in response to the horizontal location of the detected visual block. If the width of the block is above 80% of the Pixycam camera width, the claw closes and the drive motor stops.



```
1  setup:
2      attach servo claw
3      attach servo steering
4      attach servo drive
5      set drive 90
6
7      wait 1 sec
8      set drive 120
9
10     integral = 0
11
12  loop:
13     left = read left photoresistor
14     right = read right photoresistor
15
16     angle = 90
17     angle += Kp * (threshold - left)
18     angle -= Kp * (threshold - right)
19
20     integral += Ki * (threshold - left)
21     integral -= Ki * (threshold - right)
22
23     angle += integral
24
25     set steering angle
26
27     if no blocks:
28         goto loop
29
30     angle = 90
31     angle += Kp_pixy * (PIXYCAM_WIDTH/2 - block_horizontal)
32
33     if block_width > 0.8 * PIXYCAM_WIDTH
34         close claw
35         stop drive
36
37     goto loop
```

ACTION ITEM REPORT

I) Task Assignment

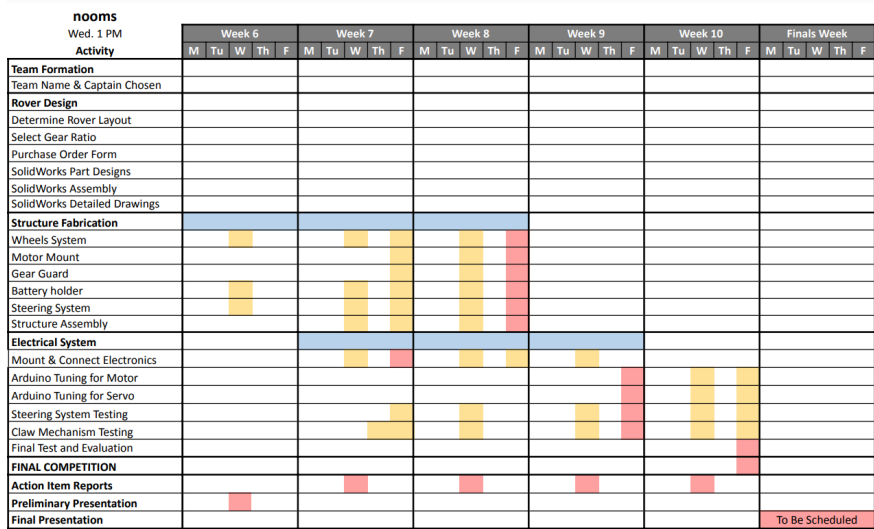
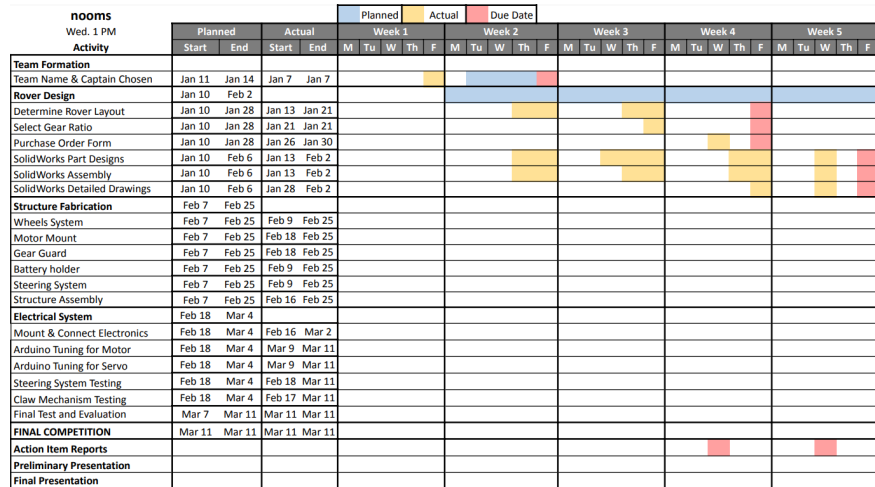
The design team consisted of Owen Fang and Ozzy Sanchez-Aldana, whose creativity conceived the rover and whose diligence virtually brought the rover to life in Solidworks so that the rest of our team could see. Owen came up with the unique claw and steering setups while Ozzy Sanchez-Aldana assembled those and everything else together in Solidworks. The latter rendered the final design as seen on the title page.

The fabrication team consisted of Octavio Adame and Matt Scott. After the Solidworks design had been greenlit by the team, Octavio and Matt worked on laser-cutting and 3D printing the parts with assistance from the design team, and thus assembled everything together.

In the meantime, Shane had been working on the code, making sure the brain of the rover would run smoothly while the body was being designed and brought to life. Nicole had been working as the overseeing team captain, managing the documentation and presentations while providing input when necessary on design or assembly.

In the final week, during rover competitions, Octavio, Matt, Owen, and Shane all worked vigorously to try and get into the rover final competition. Their efforts paid off, because they ended up solidifying 6th place in the rover competition.

II) Gantt Chart



EVALUATION



I) Calculations

Weight of the vehicle. Predicted drive time. Discussion of mechanical advantage, stall torque and stall weight for your chosen gear ratio (include calculations and reasoning for your choice).

Battery life

The rover weighed in at about 3.5 pounds and 8 oz. including the battery. Assuming 40mA for the Arduino, 140mA for the Pixy cam, 3A and 220mA for the two servos, and 3A for the motor and motor controller, the predicted run time (at free speed) would be about 33 minutes with a 3600mAh battery. $(3600\text{mAh}/6400\text{mA}) * 60$

Mechanical advantage

We chose the 52 tooth gear for a middle ground between torque and speed. Later we would find this to still be too fast for the photoresistors. This gave us a 3.25:1 gear ratio, which translates to a 3.25 times increase in force. The listed motor stall torque at 6V was 0.85 kg cm. Our battery was 7.2V, so this value will be an underestimate. The stall weight for this gear ratio was 1.719 kg, or 3.7 pounds. The rover was able to move at a brisk pace, but was close to the stall weight, as any additional load or incline would quickly slow it down.

II) Test Plan

The rover was tested after all components were mounted. Most physical interfaces (gear teeth, steering system) worked on paper in the CAD, and their motion reflected in the final model. Running the motors after assembly showed the expected results (drive motor moved the rover, steering servo and claw servo both correctly actuated their respective subsystems.) Easily the most laborious part of testing was the programming and sensors, as we battled changing lighting conditions and limited a limited photoresistor array.

III) Results & Discussion

The rover was structurally very sound, there was little body flex and had a roughly centralized center of mass. Despite the 3 wheel design, there were no tipping issues thanks to the short wheelbase. The claw was unfortunately very brittle and front end collisions would often require retiming the gears or result in broken gear teeth. The steering angle would be too extreme at high speeds, so a slower gear ratio would have helped prevent skidding. Additionally, the more photoresistors would help with sensor accuracy, as line tracking was essentially impossible. The pixy cam was met with success, as the rover could successfully track targets and collect the cups from a good distance. If done again, the consensus was that we needed more photoresistors.

Figure 2: Body Sub-assembly

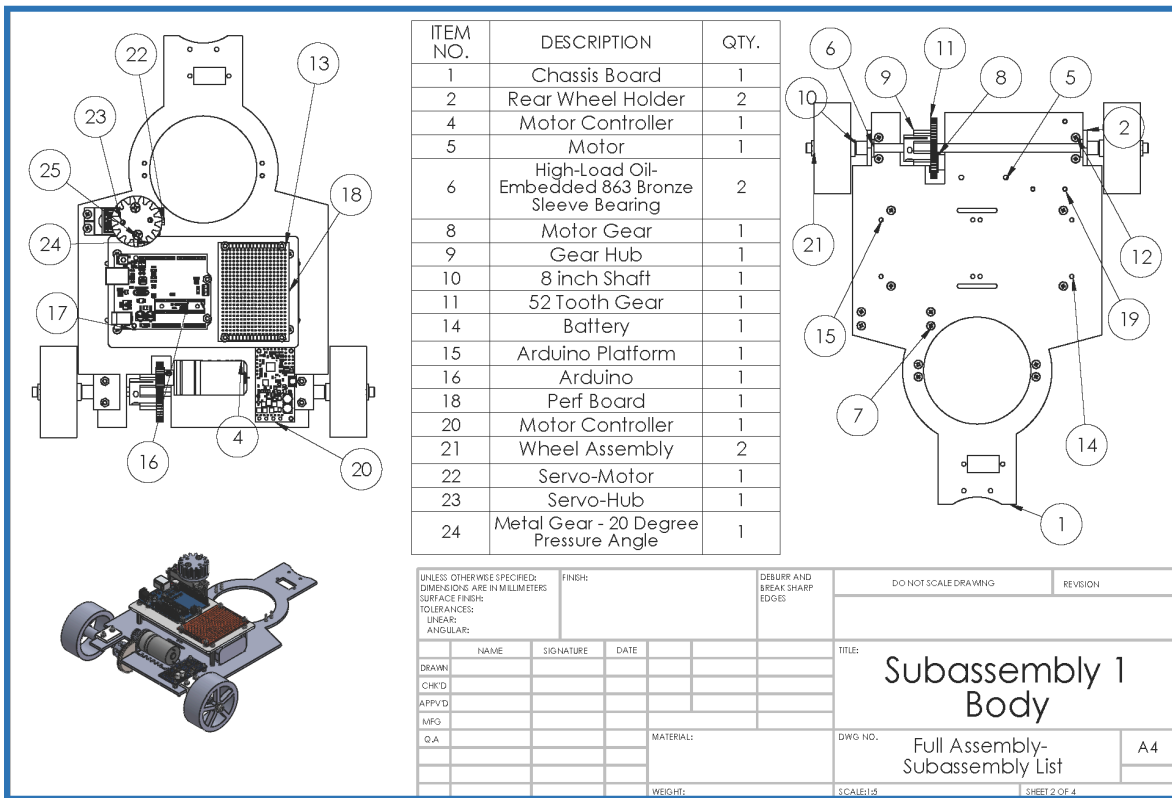


Figure 2a: Chassis

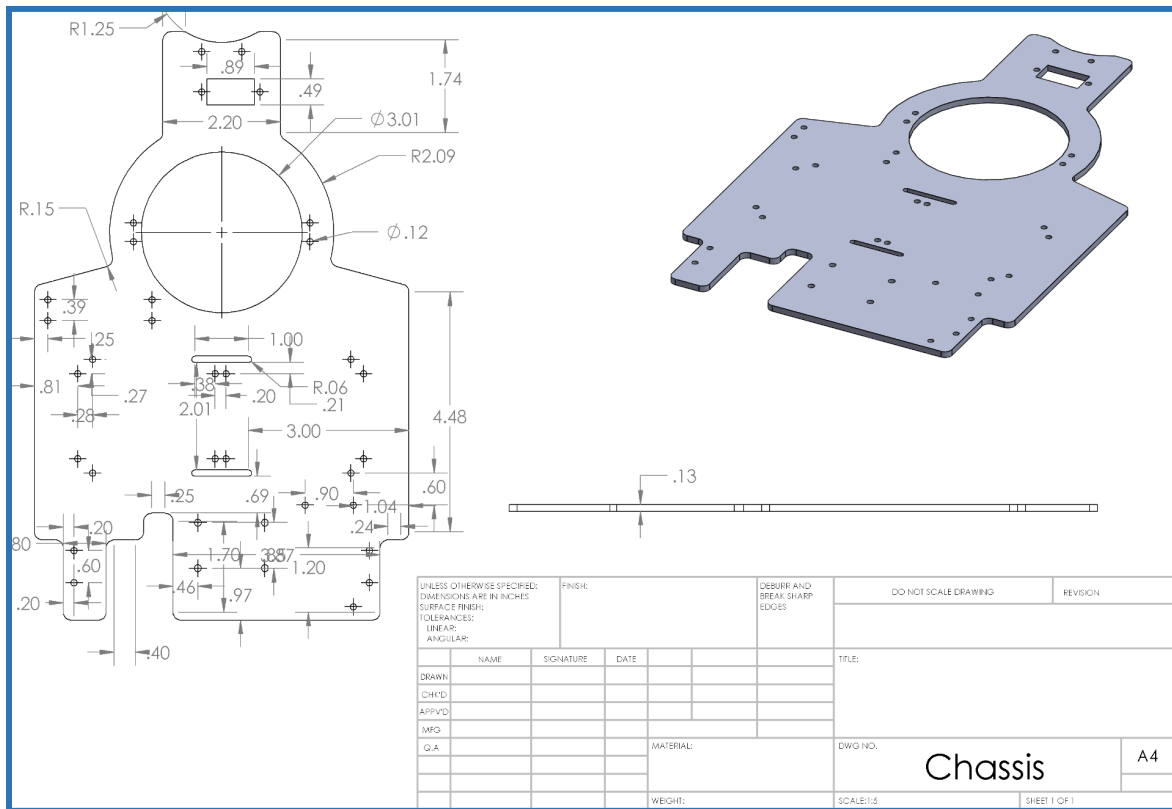


Figure 2b: Arduino Platform

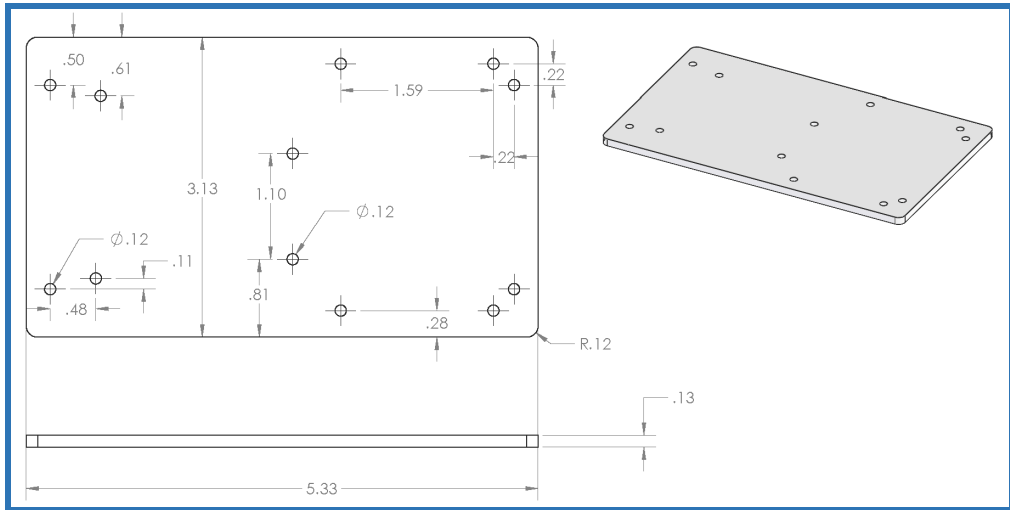


Figure 2c: Rear-wheel Holder

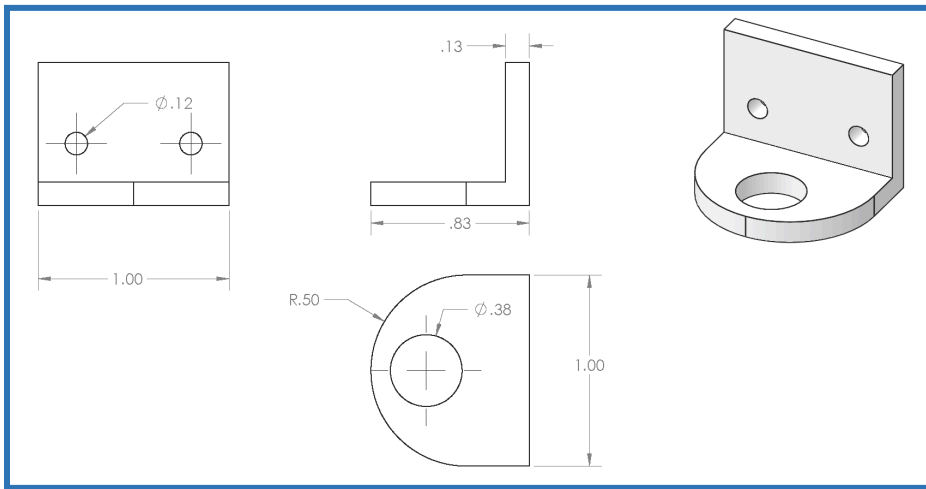


Figure 3: Steering Sub-assembly

ITEM NO.	DESCRIPTION	QTY.
1	Steeromg Wheel Holder	1
2	High-Load Oil-Embedded 863 Bronze Sleeve Bearing	1
3	2 inch shaft	2
4	65 mm wheel	1
5	Wheel Mount	1
6	Zinc-Plated Alloy Steel Socket Head Screw	1
7	Zinc-Plated Steel SAE Washer	1
8	Set Screw Shaft Collar	2
9	Set Screw Shaft Collar	1
10	High-Load Oil-Embedded 863 Bronze Sleeve Bearing	1
11	Steering Chassis	1
12	Metal Gear - 14-1/2 Degree Pressure Angle	1

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS		FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
SURFACE FINISH:					
TOLERANCES:					
LINEAR:					
ANGULAR:					
DRAWN	NAME	SIGNATURE	DATE	TITLE:	
CHECKD				Subassembly 2	
APP'VD				Steering	
MFG				DWG NO.	A4
QA				Full Assembly- Subassembly List	
			MATERIAL:	SCALE:1:15	SHEET 3 OF 4
			WEIGHT:		

Figure 3a: Steering Wheel Holder

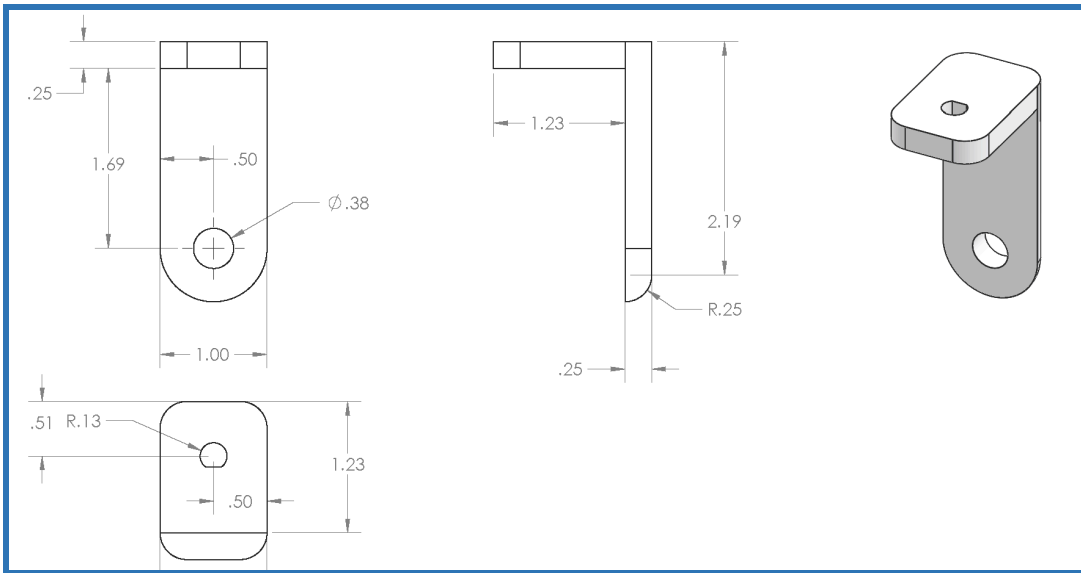


Figure 3b: Steering Chassis

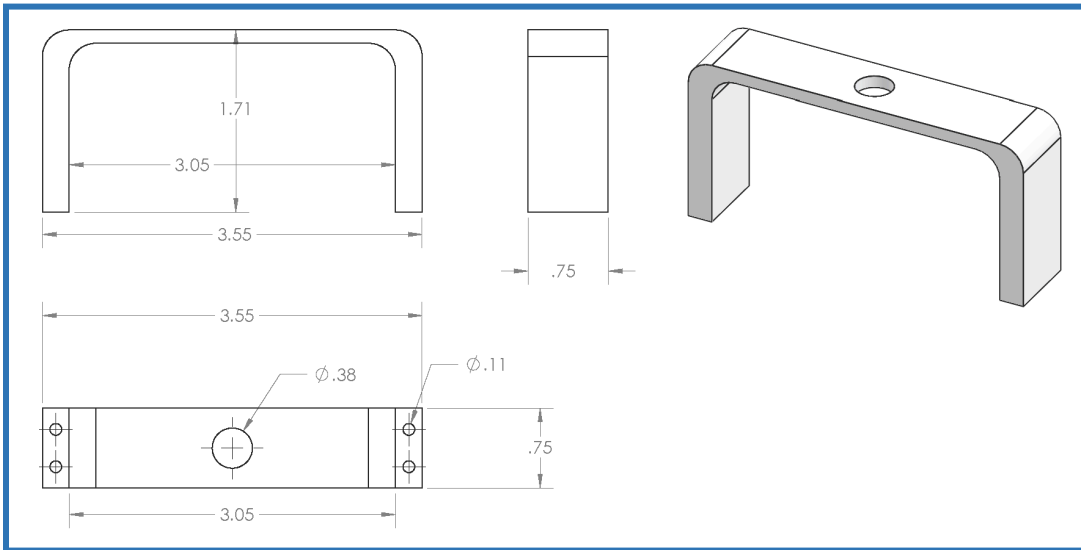


Figure 3c: Steering Gear

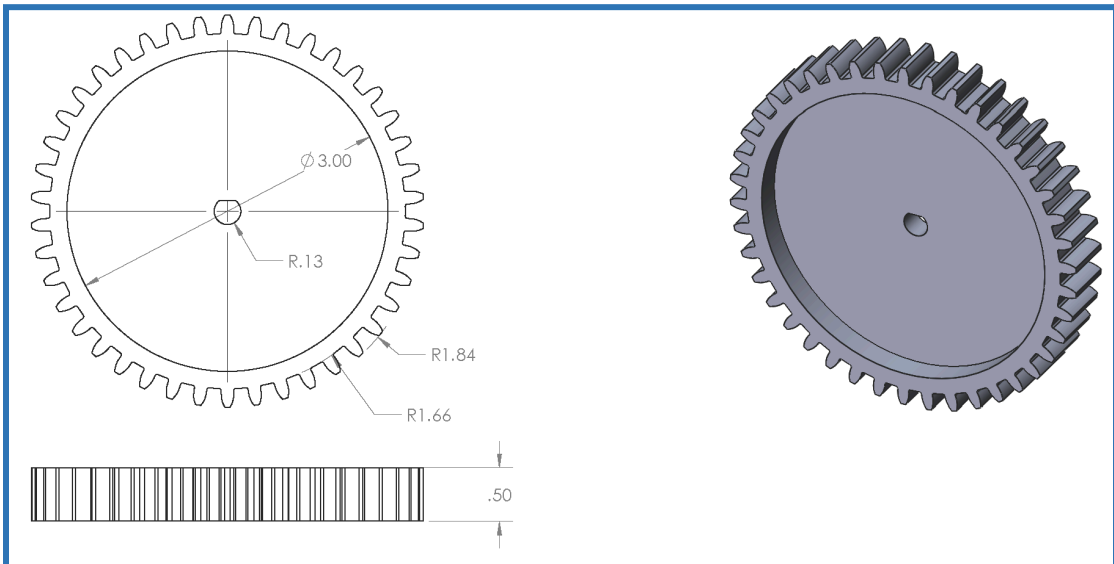


Figure 4: Claw Sub-assembly

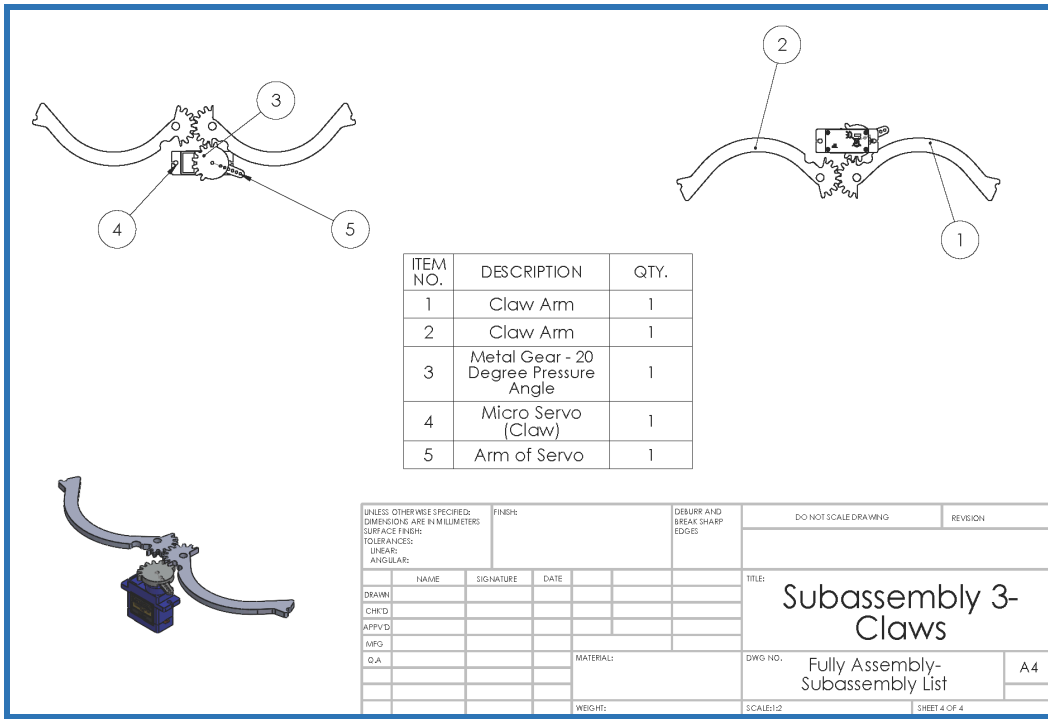


Figure 4a: Claw Arm

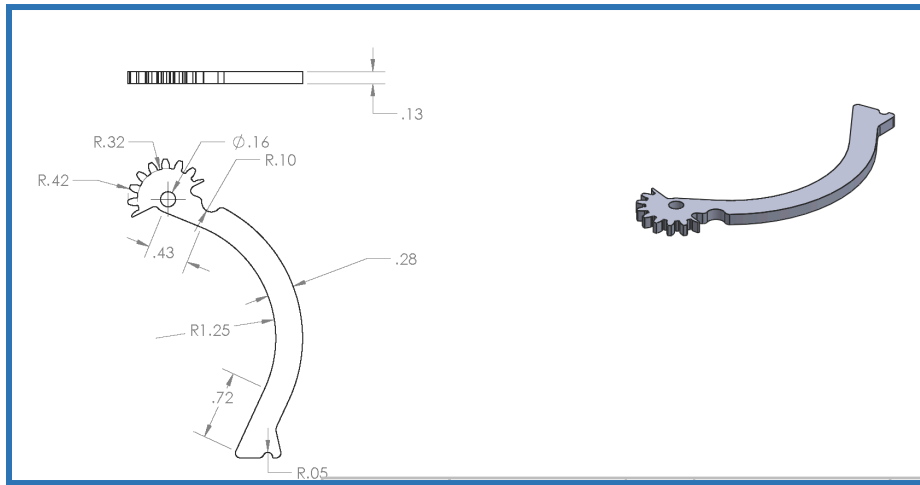


Figure 4b: Claw Servo Gear

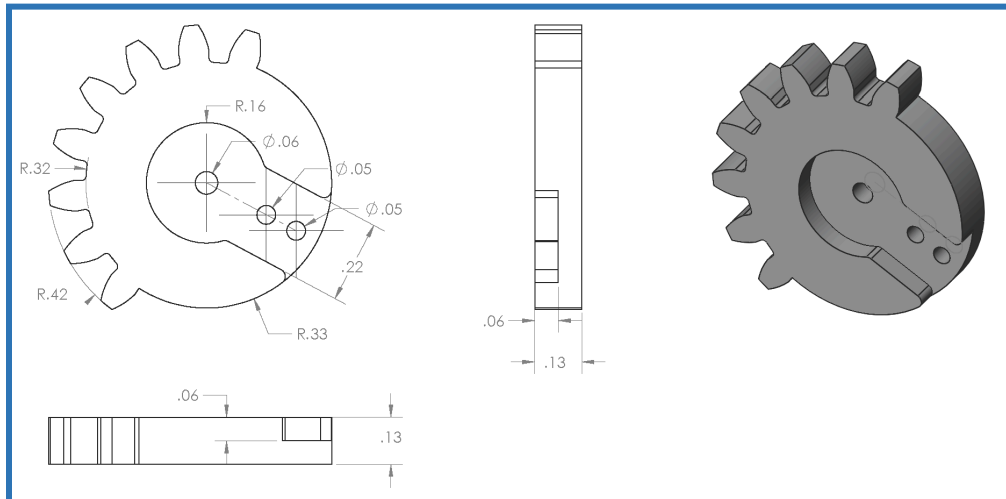
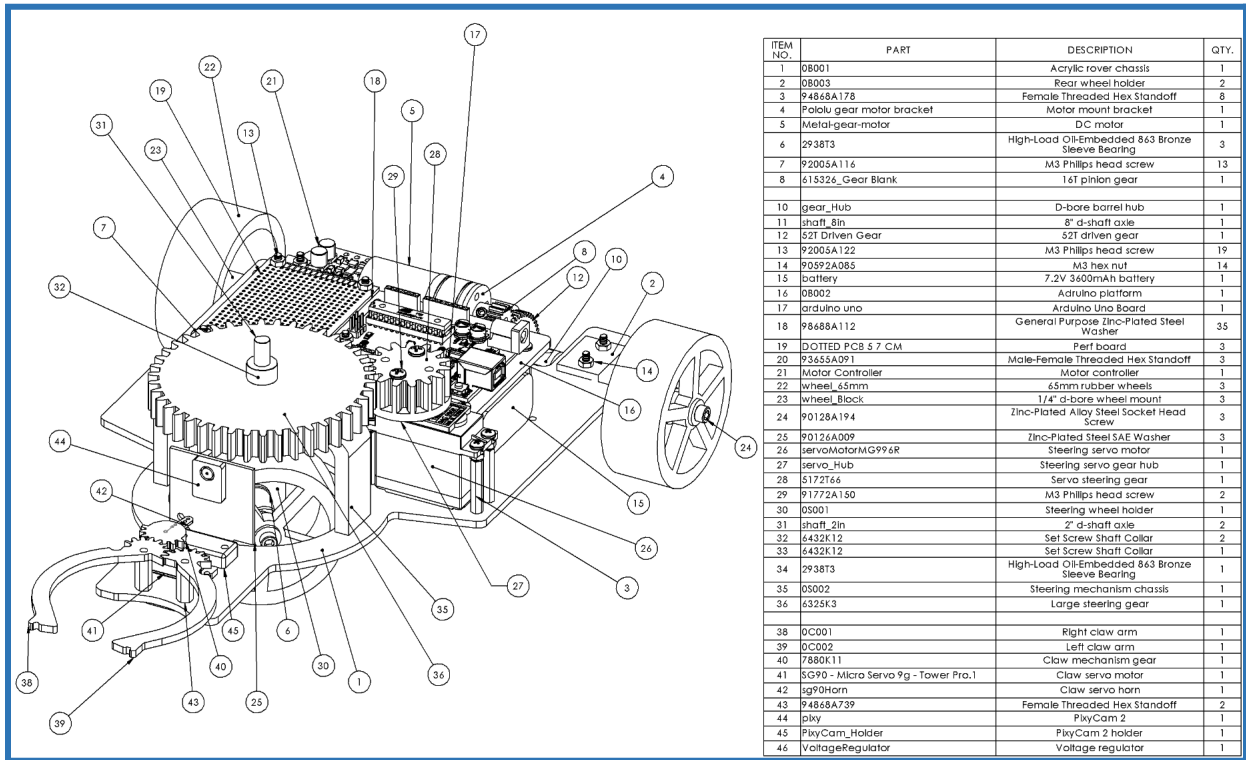


Figure 5: SolidWorks Bill of Materials



APPENDIX B: BILL OF MATERIALS

Quantity*	# of Units in package*	Company	Item Description*	Catalog #*	Price*	Estimated Extended Price*
1	1	Pololu	DC motor	3201	\$21.95	\$21.95
1	1	Pololu	Motor mount	2676	\$3.73	\$3.73
1	1	Pololu	Motor controller	1363	\$50.00	\$50.00
3	20	Amazon	Perf board	B07XYL451Y	\$7.19	\$1.08
4	30	Amazon	Photoresistors	B01N7V536K	\$5.35	\$0.71
1	12	Amazon	microservo	B08FJ27Q1H	\$20.99	\$1.75
1	4	Amazon	large servo (steering)	B07MFK266B	\$27.99	\$7.00
1	2	Amazon	Battery	B07VLKP6RJ	\$31.99	\$16.00
1	1	Servo-City or Actobotics	Servo hub	525125	\$5.99	\$5.99
1	1	Arduino	Arduino	A000066	\$23.00	\$23.00
4	100	Amazon	LED	B077X95F7C	\$6.99	\$0.28
8	100	Amazon	360-Omega Resister	B07HDGCD76	\$4.99	\$0.40
1	2	Amazon	Voltage regulator	B07DYXTX9H	\$11.89	\$5.95
1	1	Sparkfun	Pixycam	SEN-14678	\$59.95	\$59.95
			Section total		\$197.79	
3	4	Amazon	Wheels (65 mm)	B00K73YYBQ	\$12.53	\$9.40
2	1	Servo City	Shaft mount (2 inch)	634068	\$1.69	\$3.38
1	1	Servo-City	8in shaft	634088	\$3.69	\$3.69
4	1	McMaster	Bearing	2938T3	\$0.86	\$3.44
4	1	McMaster	Shaft collars	6432K12	\$1.28	\$5.12
1	1	Servo-City	52 teeth gear	RHA32-36-52	\$5.44	\$5.44
1	1	Servo City	Motor gear	615326	\$7.99	\$7.99
1	1	Servo-City	Gear hub	545692	\$6.99	\$6.99
3	2	Servo-City	wheel mounts	545656	\$7.99	\$11.99
			Section total		\$57.43	
12"x12"	1	McMaster	Acrylic (12"x12")	8589K41	\$6.35	\$6.35
Hours			3D printing: some steering parts		(\$4.00 per hour)	
\$1 per sheet cut			laser cut: chasis, some steering parts, claw			
						\$261.56
					\$330 max	
				*Tax rate:	7.75%	\$20.27
TOTAL ORDER PRICE						\$281.83

APPENDIX C: ARDUINO AUTONOMOUS MAPPING CODE

Include full code from Arduino, use comments within code for added clarity.

```
#include <Pixy2.h>
#include <Servo.h>

#define PIN_PR_L      A0
#define PIN_PR_R      A1
#define PIN_CLAW      5
#define PIN_STEER     2
#define PIN_DRIVE     3

/* Max diversion from 90 degrees */
#define STEERING_MAX  100

/* Proportional response and integral summation begins after threshold */
/* Proportional response reaches maximum at minimum photoresistor value */
#define PR_L_THRESHOLD  430
#define PR_L_MIN        350
#define PR_R_THRESHOLD  270
#define PR_R_MIN        220

Pixy2 pixy;
Servo claw;
Servo drive;
Servo steer;

void setup()
{
  claw.attach(PIN_CLAW);
  drive.attach(PIN_DRIVE);
  // Drive must be set to 90 before setting actual speed
  drive.write(90);
  claw.write(160);
  steer.attach(PIN_STEER);
  pixy.init();

  Serial.begin(9600);

  delay(1000);
  drive.write(120);
}

void loop()
{
  follow_line();
  seek_cup();
}

/* Adjust servo based on cup horizontal location */
void seek_cup()
{
  int angle;
  int num = pixy.ccc.getBlocks();

  if (!num) {
    return;
  }

  Block b = pixy.ccc.blocks[0];
```

```

angle = map(b.m_x, 0, 316, 90 - STEERING_MAX, 90 + STEERING_MAX);
steer.write(angle);
Serial.println(angle);

/* Grab cup */
if (b.m_width > 220) {
    claw.write(100);
    drive.write(90);
}
}

/* Adjust servo based on left and right photoresistor values */
/* This implements proportional, integral, and derivative terms */
/* Derivative is unused */
void follow_line()
{
    int angle = 90;

    static unsigned long prev_ms_int;
    static int integral;

    int left = analogRead(PIN_PR_L);
    int right = analogRead(PIN_PR_R);

    /* Outer left PR proportional response */
    left_val = constrain(left_val, 0, PR_L_THRESHOLD);
    if (left_val < PR_L_THRESHOLD) {
        angle += map(left_val, PR_L_MIN, PR_L_THRESHOLD, -STEERING_MAX, 0) / 2;
    }

    /* Outer right PR proportional response */
    right = constrain(right, 0, PR_R_THRESHOLD);
    if (right < PR_R_THRESHOLD) {
        angle += map(right, PR_R_MIN, PR_R_THRESHOLD, STEERING_MAX, 0) / 2;
    }

    /* Integral response */
    /* Wait a bit to take a snapshot so that we don't blow up this variable */
    if (millis() - prev_ms_int > 100) {
        integral -= PR_L_THRESHOLD - left;
        integral += PR_R_THRESHOLD - right;
        prev_ms_int = millis();
    }

    angle += integral / 3;
    angle = constrain(angle, 0, 180);

    steer.write(angle);
}

```

APPENDIX D: REFERENCES

Numbered list of references used in the report.



Motley Fool, 30 Sept. 2019,
<https://www.fool.com/investing/what-does-the-future-hold-for-self-driving-cars.aspx>.

Person. "A Brief History of Autonomous Vehicle Technology." *Wired*, Conde Nast, 10 Aug. 2016,
<https://www.wired.com/brandlab/2016/03/a-brief-history-of-autonomous-vehicle-technology/>.

END OF REPORT