CYBERFROG ROVER

Front Range Community College

Faculty Mentor: Stephanie Beck

Sam Kasten

Nick Chiang

Jorge Arroyo

Federico Chavez

Chloe Busick

Table of Contents

ABSTRACT

The goal of this project was to create a fully autonomous robot that could navigate obstacles in uneven, sandy terrain with a budget of $500. Our team consists of five community college students with little to no prior robotics experience, with a goal to learn from each step of the process. Taking inspiration from the famous Mars rover, Curiosity, we set out to design and build a six-wheeled, autonomous robot featuring a rocker-bogie suspension system. The team split into two separate groups, one to design the body using Fusion 360 and the other to start researching components and controlling the electronics. The electronics team settled on a list of components needed to start work on the project and quickly purchased them. The design team started on a basic prototype using the information gathered at the time. The team wanted to incorporate 3D printing into our project as it gives engineers a way to rapidly create prototypes custom to their needs. This led to printing a prototype at fifty percent scale, and using the information gathered from the physical prototype, the design team made changes to improve the design. The electronic components were successfully tested separately to get a better understanding of their individual functionality. By the time this paper was finalized, the design team was finishing the updated design and working on printing the final full-scale components, while the electronic team was taking all the vital electronic components and combining each of the systems. The team was confident that the rover they had designed would be able to successfully navigate the rough terrain at the robotics challenge.

INTRODUCTION

The goal of our seven-month long experiment was to create a fully autonomous robot under nine pounds that could successfully navigate obstacles in sandy terrain without exceeding a budget of $500. For our purposes, success in navigation is defined as the robot detecting obstacles (such as hills or ditches) and moving over or around them without falling apart or getting stuck. In order to accomplish this goal, our team had to consider multiple aspects of the robot's design and assess how each respective aspect would contribute to or potentially detract from its overall functionality. The design of the robot was broken down into several primary interdependent components: the body, movement, and navigation.

In order to work up to our final goal, the design process was broken down into smaller goals that were then built upon as they were achieved. For the first design review, the goal was to have a general idea of what the maximum functionality for the robot would look like and decide upon some specific components such as sensors, motors, and number of wheels. For the second design review, our goal was to have the robot's subsystems planned out and determine how they would interact. By the competition, our goal was to have a functional robot including all components with maximum functionality. Each person on our team also had a personal goal to develop faculty with CAD, programming, soldering, 3D printing, and the project design process overall. Our team had access to two different Ender 3 printers- one owned by the school and one owned by one of our teammates.

METHODS/ DESIGN PROCESS

Prior to beginning the design process for our final robot, introductory kits provided by COSGC were completed in order to introduce everyone to programming and wiring a four-wheeled robot through Arduino. By the time we had completed the introductory kits, we all had basic soldering skills and were able to make a four-wheeled robot drive forward, backward, and avoid an obstacle.

For the first design review, we had to decide on how the robot would move and navigate its surroundings. Our deliberation brought us to the conclusion that we wanted a six-wheeled robot with motors housed in each wheel, a rocker-bogie suspension system, and an IMU sensor for navigation (Figure 1). The benefit of the six-wheel design with rocker-bogie suspension was its potential for navigating unevenness in sand without being derailed. Having a motor housed in each wheel of the design added to the benefit of the rocker- bogie by increasing the overall power of the robot when going uphill or through a ditch. An IMU sensor was chosen for navigation due to its capability for measuring angular rate and magnetic field (direction). At this point in time, our primary ambition for maximum functionality was to incorporate 360 degree turning wheels to enable the robot to have crab-like motion capabilities.

After the first design review, a very rough six wheeled prototype (Figure 2) was created out of foam board and components from the COSGC Virtual Robotics Workshop kits in order to create driving code for a six wheeled design. During this same time, members of our team began learning Fusion 360. From this point on, our team divided into two sub-teams focusing on physical design and programming, respectively.

PROGRAMMING

The materials chosen and the reasons for them being chosen were as follows:

- Arduino Mega 2560 board – had more versatility and inputs which we knew would be beneficial for wiring a six-wheeled robot.
- VL53L0X Time-of-Flight – two time of flight (ToF) sensors were used to help calculate the slope of an incline to determine whether to go over or around an obstacle.
- Solid wire - kept everything connected by fitting into breadboard better than more flexible wires.
- Bread board – allowed for more connections for other hardware, including the motor drivers.

- BNO055 IMU – helped guide the robot by keeping track of the general direction the robot was facing. It started at 0 degrees in the general direction it was facing and went up to 359 degrees.
- Greartisian DC 12 Volt 50 RPM motors – used to provide sufficient torque to move the robot via the wheels.
- TB6612 motor drivers – dual motor drivers allowed us to drive two motors at a time.

The team took components and parts of code from the Robotics Virtual Workshop to develop a basic avoidance prototype. Using an Arduino Uno, an IR Sensor, and a combination of motors, the prototype was quickly able to traverse around objects in its path. After the part list was finalized, work started on calibrating and initializing individual components. The Time of Flight Sensors were the first to get tested and were calibrated to detect objects within one meter of the front of the sensors. This was then followed by the 9-Axis IMU Sensor, this sensor can detect the Pitch, Roll, and Yaw of the robot. Using Adafruit libraries and stellar example code, the team was quickly able to calibrate the IMU and modify the code to work with our robot's functionality (Figure 3). There was an issue with pin switching functionality when upgrading from the prototype's Arduino Uno to the final Arduino Mega. Once that was fixed and all the sensors were tested, the team started to put it all together into a functional avoidance system. The obstacle avoidance system has the IMU scan and temporarily store the Yaw degree positioning. Then the robot starts traveling along its path by going forward slightly, followed by scanning multiple Time of Flight (ToF) Sensors. These ToF sensors are able to use light to detect objects within a little over one meter in front of the robot. This is used in combination with a few if-else loops to determine the best route forward. The process repeats on loop until power runs out or an issue arises elsewhere.

Our final code is designed so that once the robot is plugged into the battery, a set-up loop runs to calibrate the IMU and both time of flight sensors. The setup loop is also programmed to let us know if there are errors with calibrating sensors via an LED light. Once the setup loop is finished, the main code begins. The main code starts with a scan by the IMU sensor to help orient the robot and initiate driving. This module also gives a direction using the Yaw Axis or the Z Axis with a goal of staying at zero degrees. This function is then followed by a scan done by our top

ToF sensor in order to detect obstacles such as hills, ditches, or objects. If the time-of-flight sensor does not detect an obstacle within 500 millimeters, it would continue moving forward. Though, if the ToF sensor detects an object within 500 millimeters, it determines how to avoid it. Once exiting the time-of-flight scan function, the robot moves forward by approximately one-half wheel rotation. Finally, the IMU conducts a scan on itself to determine if the robot has gone too far off course. If the IMU sensor determines that the robot has veered more than 5 degrees north or south of its initial direction, it decides whether to turn right or left until it returns to its original path of origin. This process is repeated continuously at a high speed as long as no obstacles are detected.

If an obstacle is detected within 500 millimeters, another module runs that activates and scans both ToF sensors. When the two distances are known, the code takes those values and uses trigonometry to determine whether the robot is capable of traversing over the obstacle. This returns the grade or steepness of the object. In case of a hill less than sixty degrees, the robot can traverse over that hill. In the case of an object with a more vertical grade, the robot is not able to continue forward. When the robot is unable to continue forward, it uses the IMU to turn ninety degrees to the right, move forward by approximately one wheel rotation, turns ninety degrees left, and then scans the ToF sensors again. This process is continued until the bot either goes around the object entirely or finds a path that it can traverse. Once the robot no longer detects the obstacle, the IMU sensor scans to account for any potential diversion from the robot's cardinal path and corrects accordingly.

DESIGN

Materials/tools chosen and why:

- PLA- cheap and strong
- Hex bolts- cheap and allowed for varying degrees of tightness.
- Screws of various sizes
- Jump rings- heated and melted into PLA to secure wires in the arms of the robot
- Heat gun- used to pressure fit components retroactively
- Fusion 360- was free for students
- Ultimaker Cura- free slicer that was compatible with Ender 3 printer

In the beginning, nobody was familiar with Fusion 360. YouTube tutorials were relied on heavily and a significant amount of time was spent overcoming the learning curve. We knew we wanted a six-wheeled robot that would be able to overcome obstacles in the sand such as hills and ditches. In addition to six wheels for stability, a rocker-bogie system was added to increase the robot's adaptability to uneven terrain. We also knew we wanted to house the electronic components in a way that would protect them from potential harm from sand and impact from falling/turbulence resulting from terrain. The matchbox design of the body was developed early on in order to keep the Arduino board and motor drivers secure as well as easily accessible. We decided to house the motors in the wheels which necessitated a motor hanger. Motor hangers were designed to fit tightly around the motors to protect them from sand while not being too heavy. Arms of the robot were designed to be hollow in order to protect the wires and the wires were secured using metal jump rings melted into the PLA.

A prototype was printed at 50% scale (Figure 4). This prototype enabled us to identify points of failure such as lack of clearance between the body and the wheel hangers, exposed wires, wheels being smaller than we wanted in relation to the body, and the height of the robot being potentially precarious and not in line with housecat size requirements. With more redesigning, printing, and help from our industry mentor to modify warped components, we were able to better stabilize the robot while staying within requirements. We gave the robot large wheels with a V-shaped tread in order to better handle the sand, hollowed out the wheel hangers to better manage wires, and became much more familiar with the machines we were using and how long it took to print components.

We began fully assembling the robot about two weeks out from the competition and encountered significant issues when transferring the electrical systems from the code-testing prototype onto the 3D printed robot. The primary aspect of the physical design contributing to malfunction was the matchbox floor design. While we initially thought this functionality would be advantageous for accessing electronics quickly and protecting electrical components from sand, it proved to add an extra layer of complexity when troubleshooting. Namely, the slack in the wires that was necessary to enable the floor to slide out took up too much room inside the body and the colliding of bunching wires caused components to become unplugged intermittently. We planned to amend this by securing the electrical components to the roof of the chassis instead of the floor so the wires could remain stationary while still being accessible. During the same time period of assembling the bot and reprinting components, we realized we had ordered the incorrect size of gears for our differential system. Because we were quickly approaching our deadline, we opted to resize and print a differential system found on Thingiverse (Figure 6) which proved to be functional when it was implemented but did contribute to bowing. The wooden dowel axel turned out to be too weak once the electrical components were added onto the robot, so we opted to design and print an axel with a higher infill for strength and add a fourth gear to the top of the differential for more rigidity and horizontal force.

We were not able to test the robot on sandy terrain, but the design was successful when tested on an even surface. If the surface was too smooth, the robot would sometimes fall into the splits, but we reasoned that would not be an issue on sand. We decided to bring electrical tape and string with us to the sand dunes to help mitigate potential traction issues.

RESULTS

Certain aspirations for maximum functionality were abandoned around the time of the critical design review due to time spent overcoming learning curves. One hurdle to overcome was when the team accidently fried multiple motor drivers, this cost the team valuable time trouble shooting. Ultimately, we did have a functional six wheeled robot with rocker-bogie system and externally mounted electronics that could successfully avoid obstacles without being derailed when traversing an even surface (Figure 5). Whether or not the robot could successfully navigate uneven sandy terrain was not tested prior to the competition so final results were inconclusive. However, we were confident that once both subsystems were successfully integrated, the robot would perform to the degree we wanted it to. It is also worth noting that as a team, we did feel as though we had improved our technical skills significantly over the course of the project and, having made the assumption that the robot would not fall apart in the sand, considered the project and our design successful overall.

DISCUSSION

Towards the end of the project, our team realized that it could have been beneficial to have chosen to focus primarily on developing design or code, rather than both. By splitting into sub-teams of design and code, we opted to focus equally on both. While this strategy did enable members of our team to pursue their more niche interests, it did complicate final integration due to both subsystems being moderately complex. We reached a crisis moment a week and a half before the competition where we realized we were spending a significant amount of time troubleshooting aspects of the design that we had not previously been able to test when we could have been spending time working on advancing the code. Setting firmer deadlines or deciding early on whether we wanted the robot’s sophistication and competitive advantage to be in the physical design or code could have ameliorated this issue.

For future projects, we would highly recommend using more flexible wire, as the wire we chose to use for this project proved to be unyielding and we had trouble working it into our ideal design. If a future project also comprises of a complex physical design (as ours was in some respects), then we would also recommend assembling a physical version as soon as possible, so potential complications can more quickly be observed and addressed. On a similar note, we would suggest finalizing and ordering a materials list as soon as possible, as any mistakes or mechanical incompatibilities could also be identified, as we experienced with our differential. It is also recommended that if future projects are printing components between multiple printers, that they spend the time to calibrate the printers to have exactly the same settings. In our experience, even seemingly innocuous differences in code contributed to fit issues between components and wasted resources such as time and filament. Allocating some portion of the budget for replacement components would have been beneficial as well. Thankfully, our team had accidentally ordered six motor drivers which turned out to be helpful when we fried some. Overall, it is recommended that future teams agree on an overall design as soon as possible, set hard off-ramps, and spend as much time as possible testing and troubleshooting.

CONCLUSION

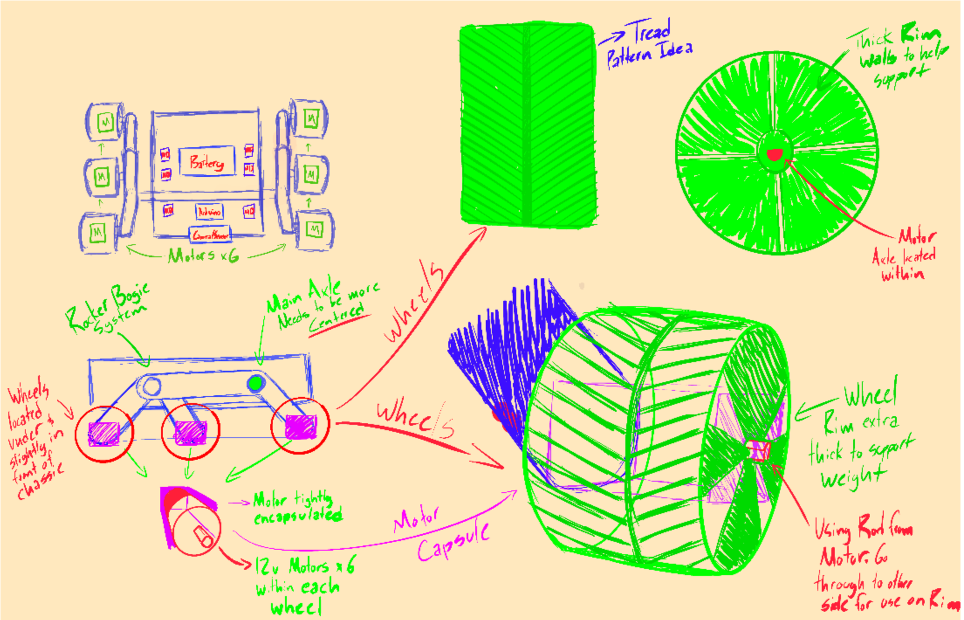
We'd like to thank our industry mentor David Stearns, faculty mentor Stephanie Beck, and previous team members Adam Hosburgh and Nick Woolsey. With their help, our entire team had their first experience with a project of this magnitude and grew in the skillsets we hoped to develop. Overall, we were proud of our achievements, and felt that we made the most out of being selected for the team and having the opportunity to work with one another. We would advise future groups to plan ahead, over communicate, and not be discouraged by releasing the magic smoke. While inhaling the magic smoke might not make you smarter by any virtue of its own, it may help you avoid making the same mistake twice.

APPENDIX

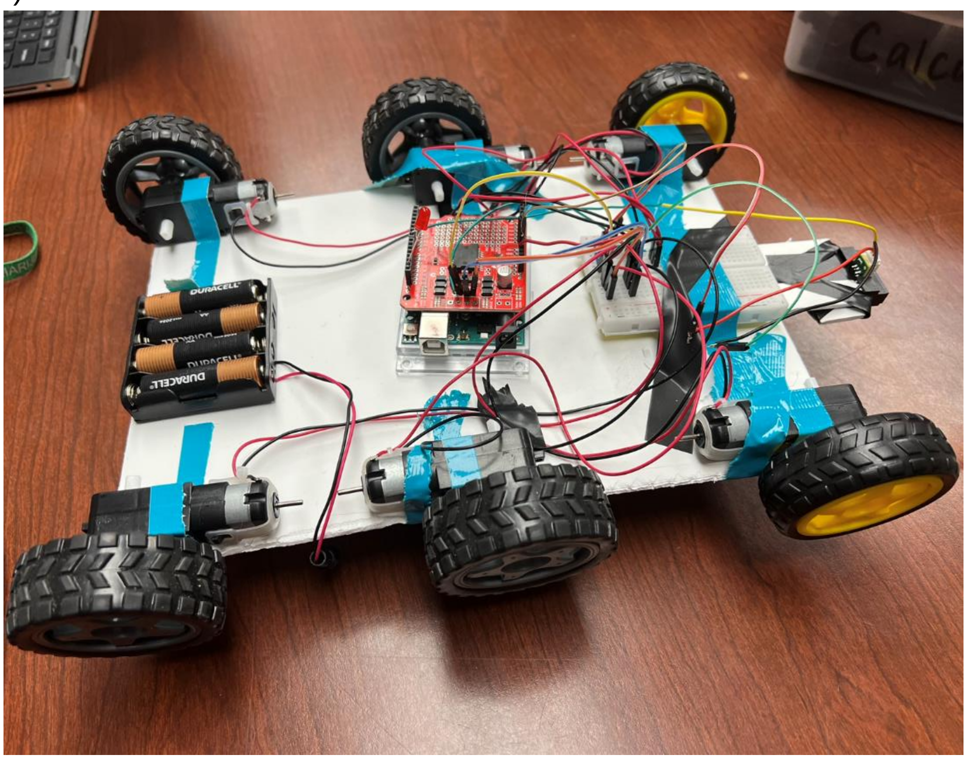
Budget Breakdown

Arudino Stackable Header Kit- R3 (PRT-11417)	3	https://www.sparkfun.cc	$5.25
Greartisan DC 12V 50 RPM Gear Motor	6	Amazon.com: Greartisa	$89.94
Telescoping Antenna Pole (2-pack)	1	Amazon.com: Bingfu 7	$6.92
Adafruit 9-DOF Absolute Orientation IMU	1	Adafruit 9-DOF Absolut	$32.84
Sparkfun Motor Driver (ROB-14450)	6	SparkFun Motor Driver	$58.80
Adafruit VL53LOX Time of Flight Distance Sensor	1	https://www.adafruit.cor	$19.50
Arduino Mega 2560 Rev 3	1	https://store-usa.arduin	$55.20
Tenergy 2 Pack 12V 2000mAh Battery Packs	1	https://www.amazon.co	$21.49
Tenergy Universal RV Battery Charger for NiMH/NiCd 6V-12V Batter Packs	1	https://www.amazon.co	$22.99
JFTech Tamiya Big Style Connector Male & Female Plug with 100 mm 14AWG Soft Silicone wire	1	https://www.amazon.co	$9.99
Red solid 22gauge electrical wire	1	https://www.amazon.co	$13.06
BNTECHGO 22 Gauge PVC 1007 Solid Electric Wire Red and Black Each 100 ft 22 AWG 1007 Hook Up Wire	1	https://www.amazon.co	$14.48
BNTECHGO 22 Gauge PVC 1007 Solid Electric Wire Green 100 ft 22 AWG 1007 Hook Up Tinned Copper Wire	1	https://www.amazon.co	$11.98
Inland PLA 3D Printer Filament 1.75m-orange	1	https://www.amazon.co	$22.49
Inland PLA 3D Printer Filament 1.75mm-Peak Green	1	https://www.amazon.co	$22.49
Inland PLA 3D Printer Filament 1.75mm-Neon Green	2	https://www.amazon.co	$22.49
Adafruit VL53LOX Time of Flight Distance Sensor	1	https://www.adafruit.cor	$19.50
HOPLEX RC Helical Gear Set 24T/12T Alloy Steel Helical Gear Set for 1/18 RC Crawler Car TRX4M Front Rear Axles(2PCS)	1	https://www.amazon.co	$20.99
			$470.40

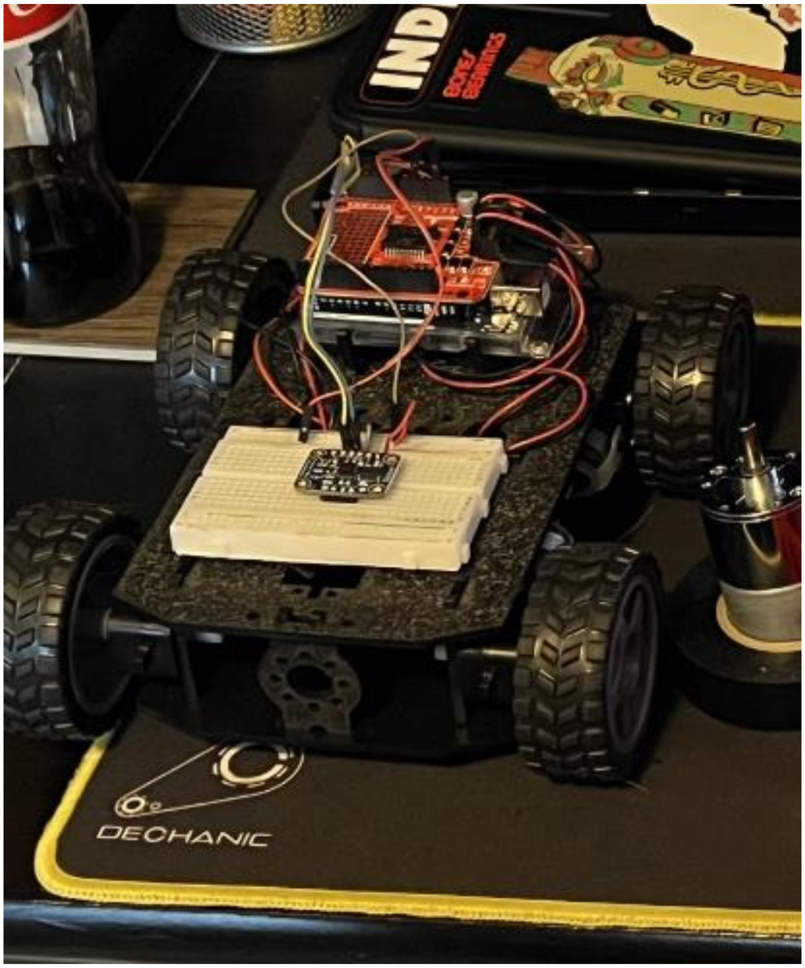
(Figure 1)



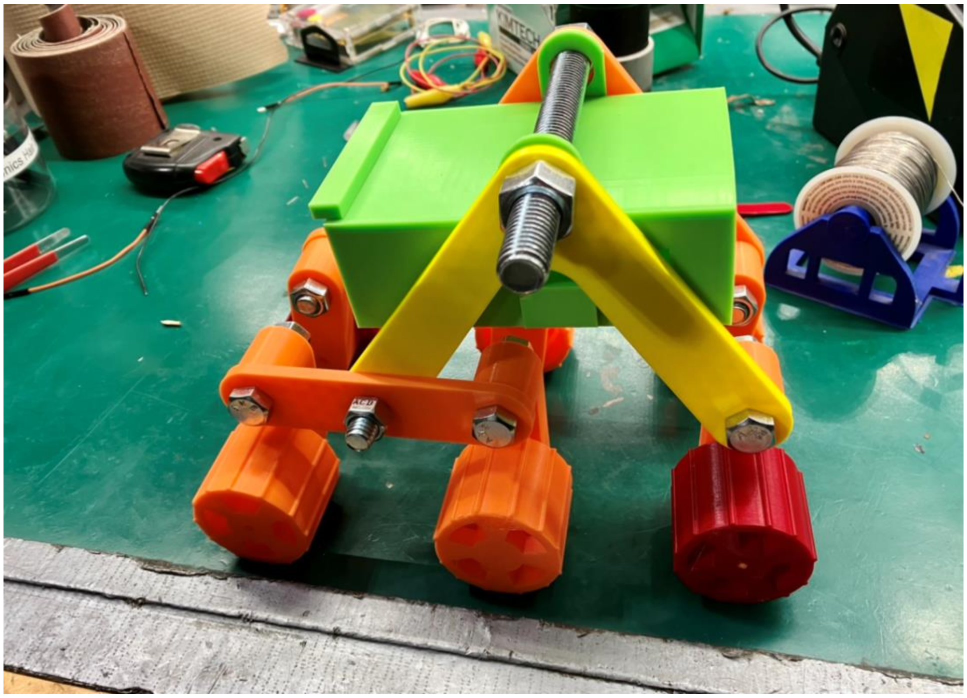
(Figure 2)



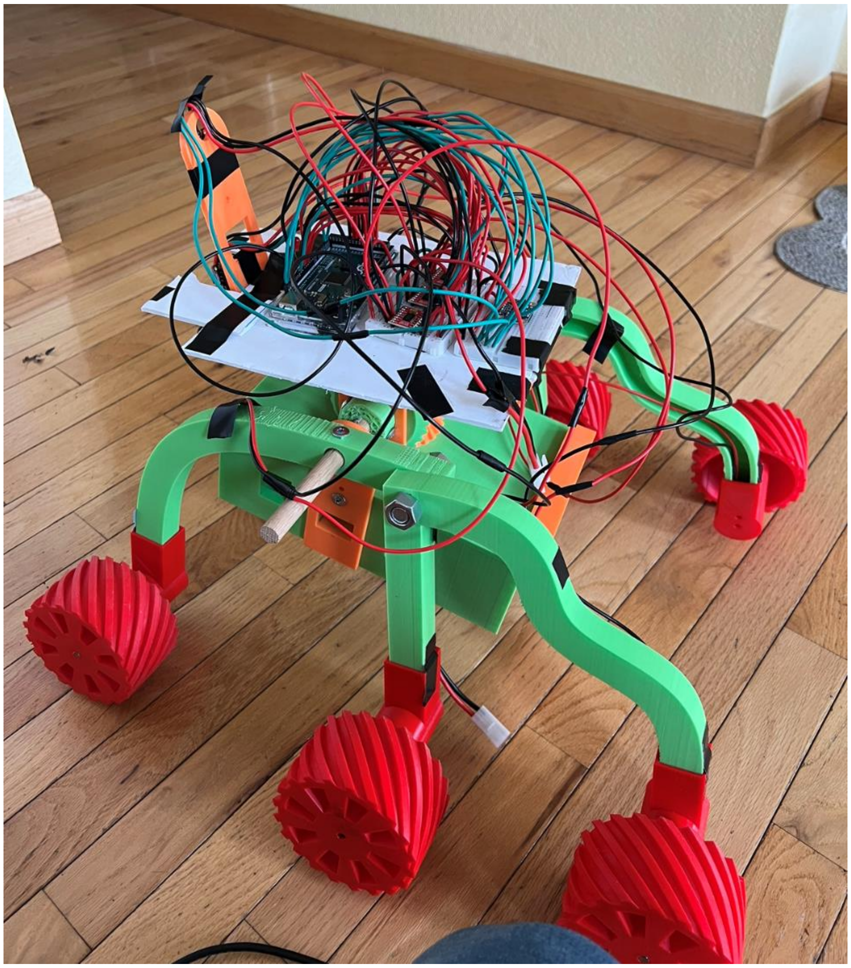
(Figure 3)



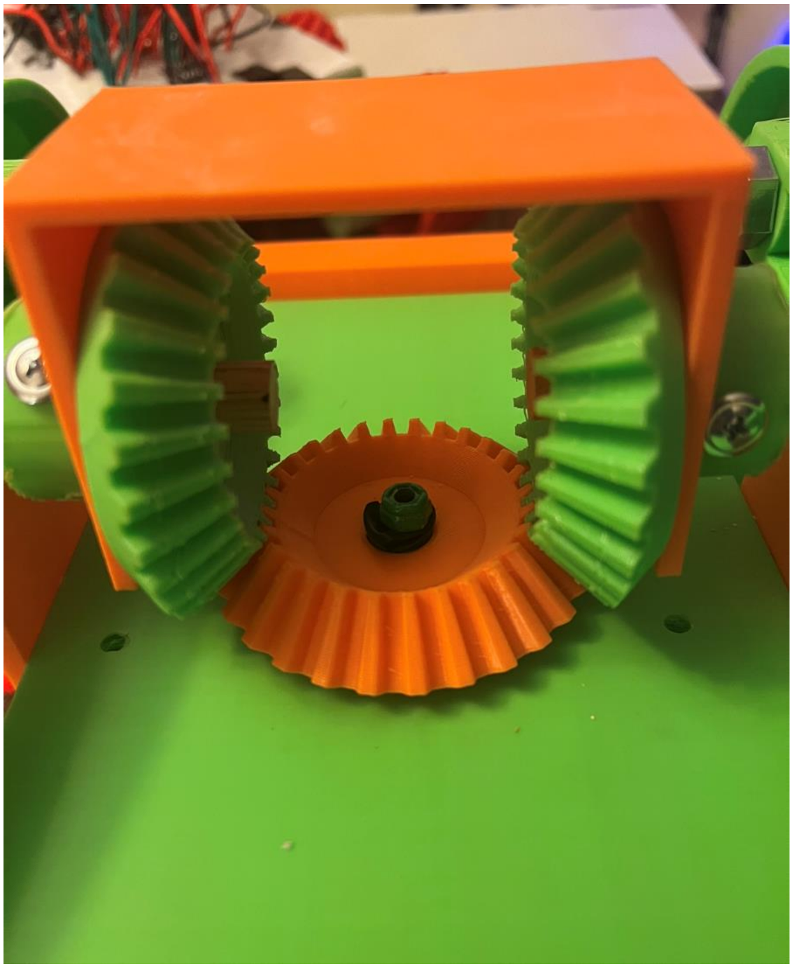
(Figure 4)



(Figure 5)



(Figure 6)



REFERENCES

Code

<Colorado Space Grant Consortium> (<2020>) <Basic Driving (No Sensors)> (<Version 1.0>) [<Source Code>]. https://github.com/adafruit/Adafruit_VL53L0X.

<Townsend, K.> (<2015>) <Adafruit Unified Sensor System> [<Source Code>]. Arduino Code | Adafruit BNO055 Absolute Orientation Sensor | Adafruit Learning System.

<Townsend, K.> (<2015>) <Raw Sensor Data> Arduino Code | Adafruit BNO055 Absolute Orientation Sensor | Adafruit Learning System.

Libraries

<Adafruit> (<2022>) <Adafruit BNO055> (<Version 1.6.1>) [<Library>]. https://github.com/adafruit/Adafruit_BNO055.

<Adafruit> (<2022>) <Adafruit_VL53L0X> (<Version 1.2.2>) [<Library>]. https://github.com/adafruit/Adafruit_VL53L0X.

<Seeed Studio> (<2020>) <Motor Driver TB6612FNG> (<Version 0.0.1>) [<Library>]. https://github.com/Seeed-Studio/Grove_Motor_Driver_TB6612FNG.

Files

<Dhanasekar> (<2017>) <Differential gear -3D printable> [<Source File>]. https://www.thingiverse.com/thing:2615515.

Thingiverse Differential Gear - https://www.thingiverse.com/thing:2615515