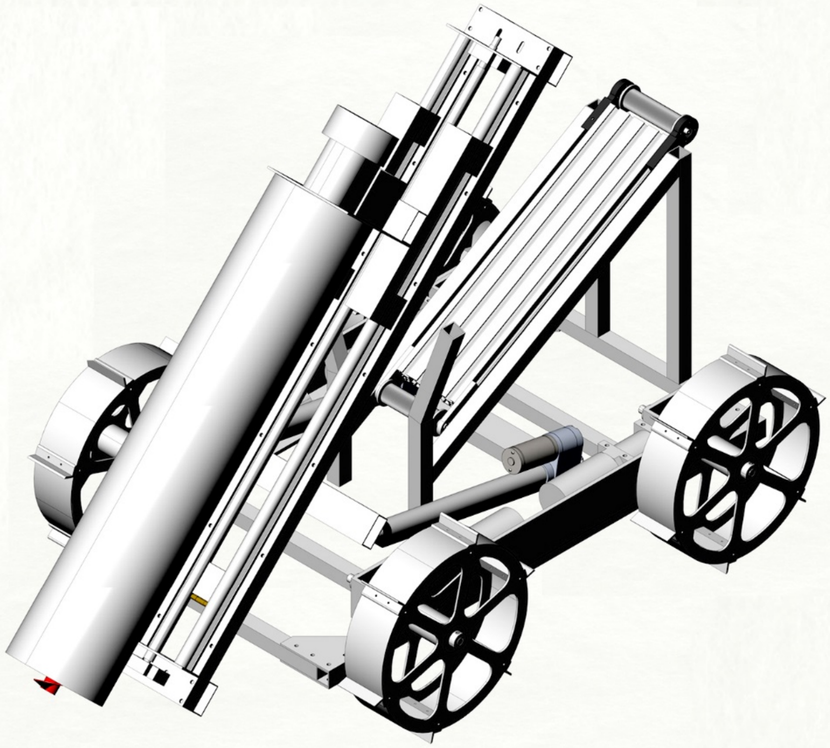


Illinois Institute of Technology

2018 NASA Robotic Mining Competition

SYSTEM ENGINEERING REPORT

SCARLET SPACE HAWKS

Team Members:

Declan Mulroy, Alyssa Deluca, Tristan Noble, Tammy Phan, Michael Duncan

Hieu Nguyen, Yuzhe Lim, Clive Gomes, Tuan Tran, Tung Nguyen, Trung Pham, Qiguang Yang, Jackson Johar, Norina Mae Balderas

Alexander Messina, Pedro L. Morales, Adrian Rivera

Shawn Zachariah, Lu Yang, Anthonny Paredes

Faculty Advisor:

Mahesh Krishnamurthy

The faculty advisor has reviewed this document prior to submission to NASA and verifies that it reflects the design process of the 2018 NASA Robotic Mining Competition entry for Illinois Institute of Technology.

ABSTRACT

Colonizing Mars is said to be the beginning of the future. As humans continue to use up all of Earth's resources, it is detrimental for the survival of humans to start considering alternatives for a new home; this means making humanity a multi-planetary species. Mars is the most inspiring and valuable of all Space Travel goals within reach. To make this dream a reality, space excavation will be play a crucial role in future Martian missions.

There are two critical components that will play major roles in the complicating Martian missions. Design, building, and testing the right excavation for the rugged, and dangerous Martian terrain. Secondly, remotely controlling a robot which is 54.6 million kilometers away will not be feasible; this will result in to developing a fully autonomous mining robot that can self-navigate and mine in the Martian terrain.

The NASA Robotic Mining Competition is an event hosted by NASA at the NASA Kennedy Space Center each year where students across the country, including Alaska, Hawaii, and Puerto Rico will be able to share their innovative designs, concepts, methods which someday could potentially be used in future NASA's space exploration missions. NASA's objective for this competition is "design and build a mining robot that can traverse the challenging simulated chaotic off-world terrain. The mining robot must then excavate the ice simulant (gravel) and return the excavated mass for deposit into the collector bin to simulate an off-world, in situ resource mining mission."

The Scarlet Space Hawks, returning for the second year will be representing the Illinois Institute of Technology, Chicago. Using the Systems Engineering process, the Scarlet Space Hawks will be developing a fully-autonomous mining robot that not only can be used for the NASA Robotic Mining Competition but can be used towards future space exploration missions.

Table of Contents

1 - Introduction

1.1 - Purpose and Objective

Developing robust and efficient mining mechanisms, autonomous methods for mining and traversing the rugged Martian terrain are the fundamental challenges that will play major roles within Martian missions. If these challenges are mastered, humans will, in return, be able harvest the Martial Soil (regolith) which can have processed to obtain vital substances like water. In a macro perspective, humans will be able to utilize these techniques to further explore asteroids and other distant planets to harvest resources and expand the colonization of humanity.

Nine years ago, NASA thought of a way which they can bring in more concepts that could serve as potential design ideas for future missions. This year, NASA will be hosting its ninth annual NASA Robotic Mining Competition at the Kennedy Space Center located in Cape Canaveral, Florida. Fifty universities from all over the country, including Puerto Rico and Alaska will come together to display and operate their Martian excavating prototype. This is turn will allow NASA to consider each design as a potential proof of concept for future missions. Teams are also required to promote Science, Technology, Engineering, and Math (STEM) within their local community as outreach events.

The Scarlet Space Hawks team formed out of the Illinois Institute of Technology at Chicago will be returning for a second year at the annual competition. Their objective is design, test, and build a fully-functioning Martian excavating autonomous robot that will autonomously traverse and dig within the rugged Martian terrain. This Martian rover will serve one central purpose: provide NASA a tool and will serve NASA in future Martian missions. Besides aiding NASA, student will be able to use this project to gain experience within manufacturing, hardware design, software design, etc.; all by applying the Systems Engineering approach.

1.2 - Problem Definition

In the ninth annual NASA Robotic Mining Competition, NASA has increased the complexity this year which has not been seen before. Students will be operating in a 3.78m x 7.38m arena which consists of a starting area, mining area, and obstacle area. To qualify for the mining category this year, students will need to collect at least 1 Kg of icy regolith (gravel). The topsoil of 30cm will consist of regolith (BP-1), and beneath there is approximately 30cm of gravel. The mining robot will be placed in an arbitrary orientation in the starting area. During the run, the robot must traverse the obstacle are avoid boulders and craters which potentially cause harm to the robot. The robot can only dig in the mining area and navigate back to the starting area where it will deposit the icy regolith and BP-1 simulant into the 1.575m x 0.475m bin that is 0.5m above the Martian simulated surface.

1.3 – Deliverables

Scarlet Space Hawks will deliver the following:

- A modular robot which can autonomously navigate, mine, and deposit icy regolith, while adhering the 2018 NASA RMC rules and regulations.
- Documentation including the Systems Engineering Report, Outreach Report, and Team Slide Presentation.
- Video demonstrating proof of life with a 30 second to one-minute video of the mining robot collecting and depositing material.
- Proper communications systems following NASA RMC communication protocols for wireless router, laptop, monitor and controllers for manual operation.

2 - Systems Engineering

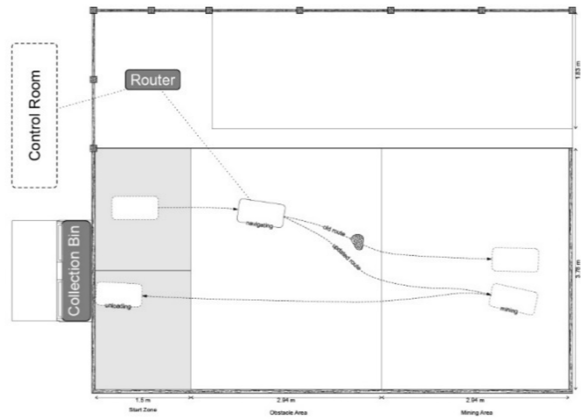
2.1 - Phase A: Concept Development

2.1.1 - Mission Objective & Design Philosophy

The primary mission objective is to design and manufacture an extraction robotic vehicle which can collect at least 1kg of icy regolith autonomously or with user input.

2.1.2 - Concept of Operation

The NASA RMC Rules and Rubrics describe the basic Concepts and Operations of the robot during the competition such as:



- Setup of a wireless router, laptop, beacons/reference points, and initial start position of the robot.
- Robot is powered on and electrical components communicate with the command room via wireless network.
- Using the lidar, the robot will locate the marker on the collection bin and define it as the origin position.
- Move to the predesignated digging locations on the far side of the arena.
- If an obstacle is detected a new path will be designated to avoid it. (i.e. rocks and craters). The position of the obstacle will be saved to the robot's coordinate system for future use.
- When the robot reaches its designated extraction location and will begin mining.
- Onboard sensors will detect the amount of icy regolith that is collected, if a set threshold is not reached the robot will move to another extraction location. Additionally, if there is less than 2 minutes left on the timer, the robot I'll start moving back to the collection bin regardless of how much regolith has been collected.
- The regolith is then taken back to the starting point by again navigating through the obstacle area.
- As the robot approaches the collection bin, it will automatically move to a set distance away to prevent loss of material upon transfer. Sensors will detect if no more material is being transferred at which point the robot will complete the delivery sequence.
- Steps 3-9 will repeat for the ten minutes duration of the competition trial.
- If autonomy is unsuccessful, the robot operator will gain control and complete the trial in manual mode.
- At the end of the competition trial, the energy consumption of the robot will be taken and reported to competition officials.

2.1.3 - System Requirements

After reviewing the NASA RMC Rules and Rubrics, the design requirements were organized into the following categories:

- May operate with or without user input (autonomous).
- Must not exceed the weight of 80 kg.
- The robot must be contained within 1.5 m length x 0.75 m width x 0.75 m height.
- Bandwidth usage must not exceed 50 kb/s
- Report energy and power usage after each competition trail.
- Must contain a red emergency stop button that is easily accessible.
- Must be sealed to protect from dust particles.
- Any navigational marker mounted to the collection bin my not exceed the length of the bin and is restricted to 9kg in mass.

2.1.4 – Systems Hierarchy

The main system for the Robot were split up into three main categories: Mechanical, Electrical, and Autonomy. From these categories, the team broke them down even further into subsystems and components shown in figure 2[1].

[1] See APPENDIX A, Figure 1

2.2 - Phase B: Preliminary Design

2.2.1 – Baseline Infrastructure

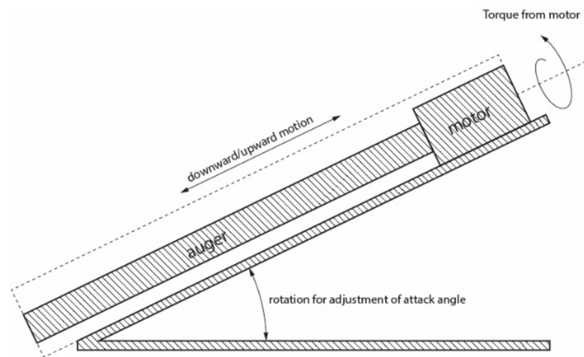
Due to the success of many elements of the 2017 robot, the team retained many features of the robot for the 2018 competition. Rather than completely redesigning from beginning, the team sought to improve the individual components as needed.

2.2.2 – Mechanical Objective and Subsystems

The objective of the mechanical team is to design, fabricate and troubleshoot all mechanical systems of the robot. This includes three main subsystems: excavation system, deposition system and mobility system. Along with constraints by NASA, additional constraints were implemented on the mechanical team itself.

2.2.2.1 – Excavation System

Objective: The objective of the excavation is to collect the rock which is the target material of the competition. The excavation system must go through the regolith soil layer approximately 300 mm deep and collect the rocks approximately 300mm deep itself. The excavation mechanism would then deposit the rock into the collections in depositing system.

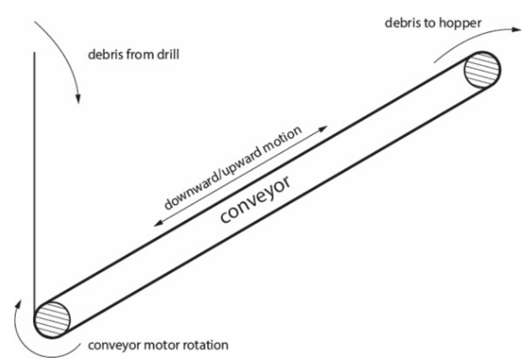


An Auger, a type of drill with a coarse pitch was chosen as the team broke down its required movements. The auger must have a way of adjusting its depth, adjusting its angle of attack, and rotating to collect and penetrate the ground.

- **Depth adjustment:** For controlling the depth a ball screw mechanism was selected by the team. A ball screw mechanism was chosen because a ball screw provides control and minimum torque requirements to control the Augers depth. A ball screw provides control because the drill would not move unless the ball screw rotates allowing for no slippage to occur as well as ease in autonomous movement. The Programming team would be able to determine the position of the drill by counting the number of rotations of the ball screw.
- **Angle of attack:** To adjust the angle of the attack for the auger two linear actuators would be used with built in potentiometers. A motor or a pulley system was not chosen because a motor located at the pivot point of the mechanism would require a high level of torque since in that position it would be a 3rd class lever vs with two linear actuators would be a 2nd class lever.
- **Deposition guide:** To ensure the material collected would be guided to the deposition system a tube needs to be constructed and placed around the auger. This tube is to guide the material displaced by the auger vertically and into the deposition system.

2.2.2.2 – Deposition System

Objective: to store the rock collected and deposit it into the collection bin in the arena.

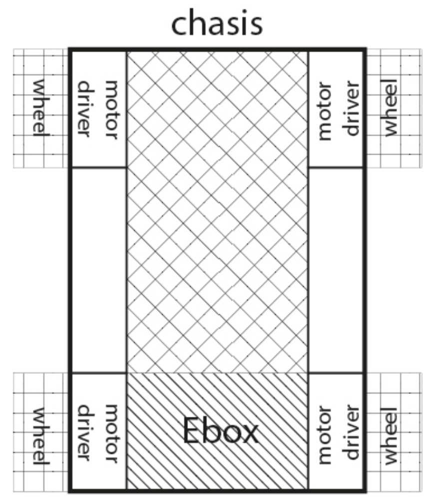


For the deposition system a conveyor was chosen as the method for depositing, for the required single motor to move the conveyor belt allows more compact design. The material would fall onto the conveyor from the excavation mechanism. There will be four Lexan walls to keep the material contained. Once the hopper is filled the robot would proceed to the collection bin and deposit the collected material into the hopper by having the conveyor move the material forward. See figure 4.

2.3.2.3 – Mobility System

Objective: The objective of the mobility system is to move the entire robot from the starting position to the excavation zone to the collection bin.

The team decided to proceed with using 4-wheel drive system. The reason is last year there were significant difficulties with using a tank drive system and the team recognized to improve the issues additional effort and mechanism would need to be implemented. This would increase in the mass as well as complexity of the system. Four-wheel drive was felt to be the simplest least complicated method.



- **Motor torque values:** Based on the soil mechanics study conducted by Angelo Gero from Oakton Community College. Our projected cargo is 20 kg and the robots estimated mass is 60 kg. The desired wheel diameter is 300mm. The program indicated a torque requirement is 23.88 N-m.
- **Space constraints:** The mobility system is limited by the digging and excavation systems. The team decided to pace the drive system on top of the chasses to protect the motors from any rocks and or debris. With space issues the team felt it would be necessary to lay the motors parallel to the wheel with a right-angle gearbox.
- **Motor choice:** Based on the time, torque values, and size constraints the team chose to use CIM motors with a 3 stage planetary gear boxes from BaneBots with 64:1 gear ratio. To transmit the torque a right-angle gearbox was needed. A cim is a 12V DC brushed motor with a stall torque of 2.42 N-m. The torque output at peak power output is 1.05 N-m. With a 64:1 gear increase makes the theoretical torque output at 67.2N-m which is 2.82 times more than the projected torque needed.

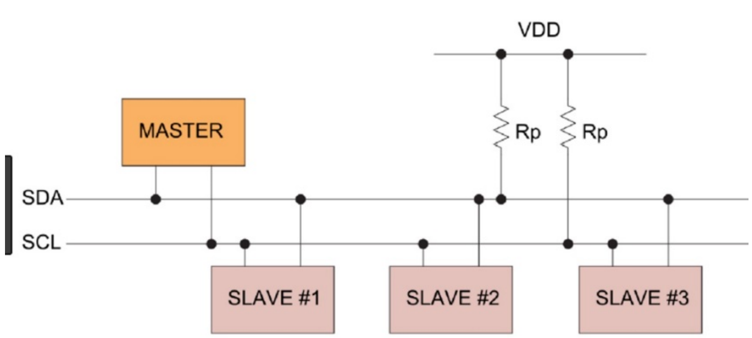
2.2.3 – Electrical Objective and Subsystems

- Ensure that all mission critical subsystems are supplied adequate electrical power to remain fully functional throughout the mission life of the robot.
- Establish a reliable means of data and feedback communications.
- Provide to appropriate subsystems, proper feedback data from various sensors placed about the structure of the robot. feedback sensors will retrieve data associated with motor rotation and actuator orientation.
- Secure all sensitive electrical devices within a singular dust proof and modular enclosure.

2.2.3.1 – Communication System

The Raspberry Pi is the Computer that the Team initially decided to use for this project, but the multitasking power of the pi is very limited due to its architecture. Keeping track of all the data from the sensors, motor controls, and algorithmic functions is too much for the pi to handle. Arduinos will be used to manipulate hardware on the low and the robot will need a way to communicate with everything. Based on research, the team has decided to implement a “Slave and master” concept. The Pi would play the master and the Arduinos would work as slaves.

There are many ways in which these components can communicate with each other. using serial connections through USB ports, wireless signals, and plain digital signals. Serial (USB) connectivity would be ideal for this project, but due to the scarcity of Ports on the Raspberry pi, the team would like to refrain from using them as much as possible. I2C is the most appealing during this stage of the project. The following image shows a rough idea of how the devices will be communicating.



2.2.3.2 – Power Distribution System

Because there are different power demands amongst components that make up the electrical system, there must be a means of safely and efficiently distributing power to all devices. Instead of using one large battery for all components, the design will implement smaller capacity batteries specifically picked to sustain the exact voltage demands of the attached devices. This is done for three reasons.

The first reason is to prevent massive wastes of electrical power, in the form of heat, because of stepping down the voltage of a larger battery down to a lower magnitude for sensitive computing devices, for example. The second reason is to evade excessive electrical magnetic interference, or EMI, because of many devices demanding power from the single battery. By electrically isolating the motors from the computers, for example, there will be a less chance of system failure because of EMI.

2.3.3.3 – Electromechanical Feedback System

Feedback of system status is primarily focused on measuring mechanical orientation. This means that all electromechanical components (i.e. DC motors, linear actuators) must return orientation data to computing devices and then to mission control. The intent of having feedback is so that in manual mode, the operator can have accurate control over electromechanical devices. Likewise, in autonomous mode, the robot will be able to retrieve and interpret orientation data. This is useful when the robot is attempting to maneuver the Martian terrain, and during the excavation of regolith.

2.3.3.4 – Electrical Enclosure Module

One major dilemma with the Mars environment is the massive amounts of dust that is present. Any robot which is to be successful in the long term must be equipped to prevent the dust from interfering with sensitive mission critical electrical systems. With that in mind, the final design will incorporate an electrical systems enclosure which will be first and foremost, completely dustproof, and secondly must be fully modular. Modularity is necessary in the event of major systems changes in future iterations of the rover. Additionally, designing a modular enclosure will make final fabrication, testing, and troubleshooting easier.

For the preliminary design the team has created a prototype electrical box that will serve as an enclosure to see if any additional space is needed for all the robot's electronics. The prototype will serve as a template that will allow the team to mount components freely and figure out mounting configurations that will be adequate for the robot's electronics. The robot has been configured with the electrical enclosure module and it has been noted that there is some extra room on the robot that will allow components to be spaced more openly. Another version will be designed within the week that will make use of the available space. The images that follow are the status of the prototype enclosure, which is currently being used on the robot.

The main objective is to fabricate an electrical enclosure that will allow the team to work on the electrical component separately from the main robot. The modular design will allow the team to troubleshoot, modify, and handle electrical components efficiently after startup or test run. A prototype has been designed to field test the placement of internal electrical components and whether there will be adequate space for each device. Upon mounting the prototype enclosure on to the chassis of the robot it has become known that there is more available space for the electrical enclosure. Therefore, the final enclosure design will be modified to reflect a more appropriately sized enclosure.[2]

[2] See APPENDIX A, Figure 2

2.2.4 – Autonomy Objective and Subsystems

The objective of the autonomy team is to have the robot perform total autonomous operations during the competition runs and to have the team members exposed the process of programming an autonomous mining robot. To achieve this goal, there are three major problems the autonomy team needs to solve. [3]

2.2.4.1 – Navigation System

1. **Localization**
   Designing the robot to be aware of its location on the arena is crucial for autonomous operation. After considering various restrictions on localization technologies (restriction of using GPS, compass, arena's wall data), the autonomy team has decided to use a Hokuyo 360 lidar(Hokuyo UST-10LN Scanning Laser Obstacle Detection Sensor) for localization. The strategy is to have the LIDAR scan for the distances and angles from the robot to a fixed marker attached on the arena bin (possibly a white foam board). The justifications for this strategy are 1) laser reflection quality from the form board is distinct from those from arena's walls; 2) the marker length is different from the arena's walls length, so although quality of reflected beam from the walls is unknown, it can be dynamically detected based the physically length of equal-reflectivity LIDAR readings. One Hokuyo LIDAR will be mounted on the back of the robot, which means that the robot should always have it back facing the arena's collection bin, which is working well with our front digging and back dumping design.
2. **Obstacle avoidance**
   Since it's not guaranteed that the robot has adequate ground clearance to run over the rocks, obstacle avoidance must be in place to ensure successful autonomous operation. The Microsoft Kinect system was chosen to help obstacle avoidance. The justification of using Kinect is it can take snapshots of the depth map in the area around the robot. Unlike LIDAR which can only scan a 2D map, the Kinect can obtain a 3D image of the area, which contains both rocks and craters info. The goal is to capture rocks' and craters' positions with respect to the robot, which will be imputed to a pathfinding code to calculate the best path to navigate through the obstacle field.

2.2.4.2 – Motor Control System

Having robot and obstacle positions only solves half of the autonomy problem, the other half involves precisely control the robot's actuators based on those position data (feedback). Closed loop control is extensively used in this year's robot. Some applications of closed loop control in the robot include speed and position control of the drive system (already done), positional control and synchronization of linear actuators (already done), path following (using LIDAR feedback to follow a calculated path to the mining and dumping area). Moreover, two short range distance sensors will be used to 1) measure gravel level stored in robot's bin and 2) position to robot to a safe distance the arena's bin so gravel will be dumped into area's bin, not area's ground. Finally, to ensure gravel flow into and out of the robot's bin, two break-beam IR sensor will be mounted on robot's bin entrance and the exit of dumping conveyor.

2.3.4.3 – Communication System

To provide and intuitive control interface of the robot, the autonomy team set out to create a graphical user interface that can perform manual and autonomous commands to the robot. In addition, the control software will also display robot health info, so the controller can monitor its progress as well as its health.

2.2.5 – Preliminary Design Review (PDR)

On December 1, 2017 the team met with the Project Manager for the PDR. He confirmed that the designs met the system requirements, the technical interfaces were consistent with the overall technical maturity, and the risk mitigation plans were adequate. After the PDR the team was cleared to move on to Phase C.

[3] See APPENDIX A, figure 3.

2.3 – Phase C: Final Design and Fabrication

2.3.1 – Mechanical: Excavation System[4]

2.3.1.1 – Auger

Auger mechanism consists of a drill head, a motor, a motor mount, a tube covering, and supports.

The auger drill head and the motor are salvaged from a purchased ice auger and salvaged the components due to efficiency and economical consideration. The purchased ice auger provides a motor and gearbox, along with the OD of the auger and ID of the output shaft of the auger to match.

The primary components taken were the DC brushless motor and gearbox, the power and reverse switches, the battery, the housing, and the auger itself. The auger is 34 inches long with a 6-inch diameter.

Blade: The auger had two blades attach at the tip of the blade. The drill taken from the ice auger was tested on soft sand ground and was proven to dig 15 inches deep without getting jammed. However, modification of the head was made by attaching a 75-degree angled tip to the auger head, so that when it intersects with pebbles the tip loose the material and prevent the auger being jammed by peddles.

Tube: The tube is 6.5 inches in diameter and made of Lexan or polycarbonate. As discussed in the previous section, it is meant to funnel crushed basalt up the auger into the hopper. The approach was to use HVAC duct with a 6in OD, approximately weighted 1kg. It came pre-fabricated and the diameter can be easily adjusted.

2.3.1.2 – Slider

The slider mechanism consists of the frame, the lead screw, the guide rails, and the sled.

3. **Frame:** The frame is made of 1”x2”x0.0625” aluminum tubing. It is 1300 mm long by 150mm wide. The frame is produced in right angle support on either end to support the bearings for the ball screw.
4. **Sled:** The sled is where the motor and tube mounted. it consists of several 1”x2”x0.0625” tubing. The motor is seated on a 3”x3” aluminum, which is mounted on sled beams. The axel must sit at least 3 inches up to prevent interference between the slider frame and the 6in-diameter drill.
5. **Lead screw:** The ball screw, guide rails, and bearings where purchased as a set. The lead screw is the primary driving mechanism of the assembly. The lead nut is attached to the sled and fixed to it, so it cannot turn. The slide motor will drive the sled to go forward or backward based on direction. The slide rails have 4 linear bearings on it which support the load of the drill rather than the ball screw. It also prevents the sled from rotating around the lead screw when the ball screw is in forward or backwards motion.
6. **Angle of attack:** To adjust the angle of attack of the drill two actuators were mounted to the chassis with a cross bar across located on the slider frame. The actuators are Firgelli 12 V DC. The Static load was 150 pounds with a stroke length of 300mm.

2.3.2 – Mechanical: Deposition System[5]

2.3.2.1 – Frame

The frame is what holds the conveyor stationary and in place. It consists of 1in Square tubing .0625-inch-thick walls. The frame as Lexan sheets attached to the frame to hold all the Crushed basalt and rocks.

2.3.2.2 – Conveyor

The conveyor was an ordered part. The motor was swapped and replaced with a Pololu 2827 Metal Gearmotor. The Stall torque is 1.2 N-m. The conveyor body come with T slotted on both sides, where the frame can be connected and tightened with bolts.

[4] See APPENDIX A, Figure 4.
[5] See APPENDIX A, Figure 5.

2.3.3 – Mechanical: Mobility System[6]

2.3.3.1 – Chassis

Chassis is the frame of the entire robot. This provides the structural support for all components as well as the base for all mechanical and electrical components. It is 900 mm long 500 mm wide made of 1”x1”x0.0625” tubing. Three major portions of the chassis the outer frame, the cross supports, and the Lexan plates. The Lexan plates provide dust protection as well as the ability to run cables and wires for the electrical enclosure module (introduced in electrical team).

The outer frame provides general support for the entire structure. The cross supports provide additional structural stability as well as support for the conveyor and the linear actuators.

2.3.3.2 – Mounts

- **Motor mounts:** The main components of the motor mount are the motor gearbox, right angle gearbox, and the L channel.
- **L bar:** The bar is 3.5”x3.5”x0.375” square extruded aluminum angle. The motor and both gearboxes are attached to the bar forming one assembly. The Assembly is then bolted to the chassis.
- **Sensor mount:** The sensor mount was designed and constructed for lidar to yaw and pitch on its x, y, z axis. A U bracket is attached to the platform to allow sensor’s rotation.
- **Gear boxes:** Right angle gearboxes are designed and manufactured to transmit the torque from the motor to the wheel. The walls of the gearbox are 3”x3”x0.25” tubes. Two bearings are mounted on either side of output shaft to ensure the it is supports and not interfering with the walls of the gearbox. The gear ratio of the gearbox is 1:1.

2.3.3.3 – Wheels

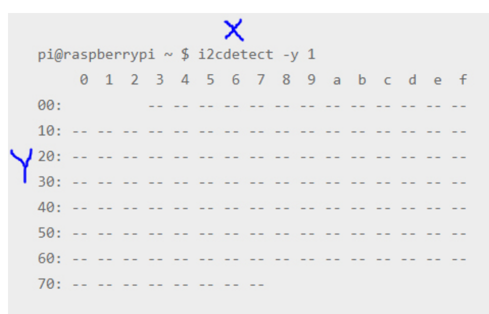
The wheels consist of 4 components: grousers, wheel face, wheel rim and support blocks, hub. The overall diameter of the wheels is 300mm. The Wheel is completely made from aluminum for repeatability increase strength and cost efficiency.

The wheel hubs were turned and broached and the wheel faces were press fitted together with the blocks in between them with the grousers.

- **Grouser:** The grousers are 1.2”x1.2” aluminum l bracket. The blocks are 0.5”x0.75” solid aluminum blocks.
- **Rim:** The wheel rim is made of .015” thick aluminum sheet.
- **Hub:** The wheel hub is 1.75-inch diameter solid aluminum. The wheel faces are 1.8in thick aluminum sheet. The drive shaft is connected through the hub.
- **Wheel Face:** Wheel face acts as secondary support as well as holding the Hub in place. The wheel hub is .125 in thick aluminum sheet that was laser cut to its current shape.
- **Blocks:** The blocks are made of 12.5mm by 18.5 mm solid aluminum blocks. They have 3mm tapped holes on the sides to hold the wheel faces to it and 6mm threaded holds on the top to hold the rim and grousers to the whole wheel frame.

2.3.4 – Electrical: Communication System

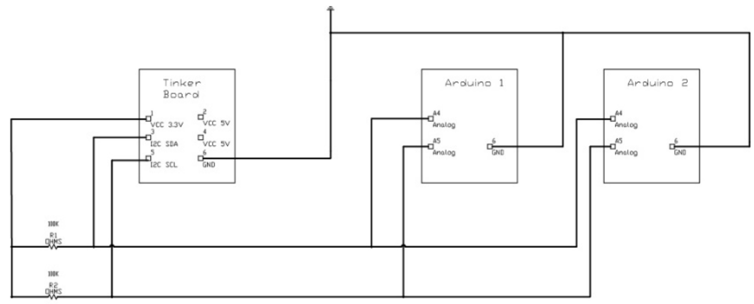
The Master-slave concept has been finalized. The Raspberry Pi, which has been replaced with the Asus Tinker Board will act as the master and will be using Python as its preferred language. The Tinker must be configured with I2C tools to access devices. To check this, the team runs a test which scans the data bus for devices which is shown on the following image. The X axis represents the address lines while the Y axis represents the number of data bits. The Arduino must be configured as a slave device by running a program initializing that Arduino to a specific address (ex. 0X04) and has the Arduino consistently listening for a command from the master.



[6] See APPENDIX A, Figure 6.

On the Tinker Board end, it must be initialized as the master Device. It needs to be able to write and read from the different addresses on the Bus. The number of addresses will be determined by the number of slaves which will be used. A manageable way to accomplish this is to write functions in Python which will make it easier to understand what is going on with the devices (ex. Read data, write data). After the devices have been initialized and the bugs have been fixed, it was time to figure out a how these devices will be connected. The following schematic is a great representation of how the devices are connected assuming we will use two Arduinos. This method of connectivity is expandable to up to 8 Arduino slaves if necessary.

The overall goal has been met and a brief description of the functionality of the system is as follows. The Arduinos will be labeled as slaves and will enter a loop which has them consistently listening to the data bus. The Tinker Board will give tasks to the Arduinos and access them through their addressable labels they have been assigned. The command will be sent to the Data bus, but only the Arduino with the Address that matches will respond to the command.



2.3.5 – Electrical: Power Distribution System

Four onboard batteries will provide complete power to all robot systems. The batteries are comprised of lithium polymer and lithium ion battery packs. The figure below is an illustration of how power will be supplied to various devices.[7]

2.3.5.1 – Lithium Polymer Batteries

Power will be provided to all DC motors using two 3-cell 11.1V Lithium polymer, or LiPo, battery. This battery has a discharge rated value of 2.2 amp-hours and can sustain continuous discharge of 30C with burst discharges at 40C. Both 3-cell batteries will be arranged electrically in the parallel configuration to provide the constant 11.1V required to run DC motors for the duration of the mission life. This exact model of LiPo battery has proven to be a satisfactory choice of a power supply because prior competition designs had incorporated the same model without any issues.

For the lower voltage, and more sensitive electrical devices such as sensors and microprocessors, a smaller 2-cell 7.4V LiPo battery will be used. This 2-cell has a rated discharge capacity of 2.5 amp-hours and is capable of burst discharge rates of 5C. 7.4V will be used to supply power to the printed circuit board which houses the Arduino Due, the Asus Tinker board, connection ports for I2C enabled sensors, and the enable relays for the auger module.

2.3.5.2 – Lithium Ion Batteries

Because the icy regolith extraction module is comprised of an off-the-shelf handheld auger system, certain aspects of its original design needed to be incorporated into the robot design. More specifically, the brushless motor which will control the rotation of the auger is powered by a 40V lithium ion battery pack which came included with the auger system. The battery pack will be used in the design because its output specifications are exactly what the auger motor needs to operate normally.

2.3.5.3 – DC to DC Converters

Mounted onto the printed circuit board inside the electrical enclosure module are two DC/DC converters, or buck converters. This circuitry is used to step the voltage from the 7.4V LiPo battery down to 5V and 3.3V. The 5V will be supplied to the Asus Tinker board, and the 3.3V will be supplied to I2C enabled sensors.

[7] See APPENDIX A, Figure 7.

2.3.6 – Electrical: Electromechanical Feedback System

For autonomy to run successfully there must be some sort of feedback system that allows the robot to be aware of its location always. The sensors used for this feedback system include Encoders, LIDAR, Distance Sensors, Gyroscope, and Kangaroo, which without autonomous operations would not be obtainable. To understand the actual functionality of each sensors and a better understanding on how the sensors were implemented the autonomous section of the systems report. On the electrical side of operations, it is critical that all connections are properly put together and sufficient power is applied to all sensors. It is also necessary that the electrical team make sure that there is no magnetic interference, which will disrupt communications and data transferred.[8]

2.3.7 – Electrical: Electrical Enclosure Module

2.3.7.1 – Wire Connector

Devices that require large amount of power will use cord grip connectors. For small signal connections the team will use Amphenol connectors coming in and out of the electrical enclosure module remain sealed from the outside environment. The choice also allows the robot to remain modular as it was originally intended for ease of use and troubleshooting if there seems to be an issue with the electronics. The connectors allow an easy disconnect from the main electrical enclosure module which makes a more fluid and easy procedure to troubleshoot

.

2.3.7.2 – Printed Circuit Board

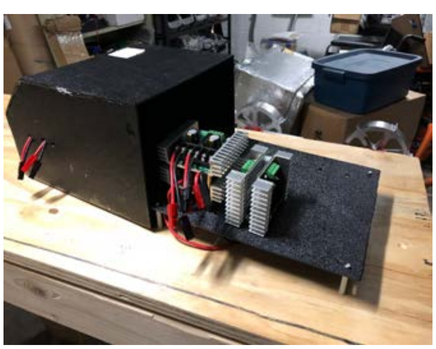
The printed circuit board (PCB) internal to the electrical enclosure module has multiple purposes. First, it allows a convenient way to centralize all I2C and serial communication ports. Second, 7.4VDC supplied by a LiPo battery is applied to the board to be distributed to the Arduino Due and Tinker Board. Third, at 10cm x 17cm the PCB provides a way to reduce the footprint of the electrical enclosure module. Last, because the PCB is to be a 4-layer board, with the 2 internal layers being designated as Vcc and ground. This provides a means of reducing electromagnetic interference to sensitive electronics. Having a 4-layer board also vastly simplifies the routing of copper between pins. [9]

The PCB layout includes:

- Header pins for the Arduino due to be mounted directly on the board
- 2x20 pin header port for the Asus Tinker Board to be connected via ribbon cable
- 2x9 pin GPIO port for interfacing with Tinker board for testing and troubleshooting
- 2 DC/DC converters for stepping down voltage to 5 and 3.3 VDC
- 8-port serial and 16-port I2C hubs for communications
- 2 screw terminals for supplying 5 and 3.3 VDC
- 2 relays for enabling auger control

2.3.7.3 – Electrical Enclosure Module

The electrical enclosure has been made with a quarter inch thick clear acrylic and has been modified to use.[10]



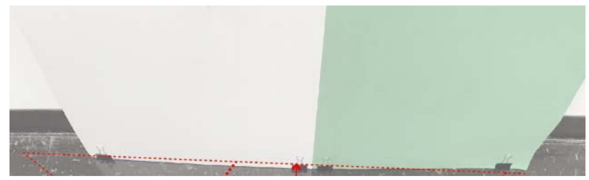
[8] See APPENDIX A, Figure 8.
[9] See APPENDIX A, Figure 9.
[10] See APPENDIX A, Figure 10.

2.3.8 – Autonomy: Navigation Module

In order to have the robot autonomously maneuver to the mining area and return home safely. The IIT robot has been equipped with following features: self-localization, the ability to detect and avoid obstacles and the ability to self-align with the collection bin.

2.3.8.1 – Localization

Purpose: Localization is the most important features and the foundation for the robot's navigation system because in order to navigate through the obstacle field and mine at a designated area, the robot needs to know its position at all time. For example, our obstacle avoidance and path finding methods require the relative position of the obstacles to the robot, and the relative position of the robot to the arena's collection bin to come up with the best path to the mining. With localization, the robot can be commanded to go to any arbitrary position, to face any orientation, and maneuver along any set path.



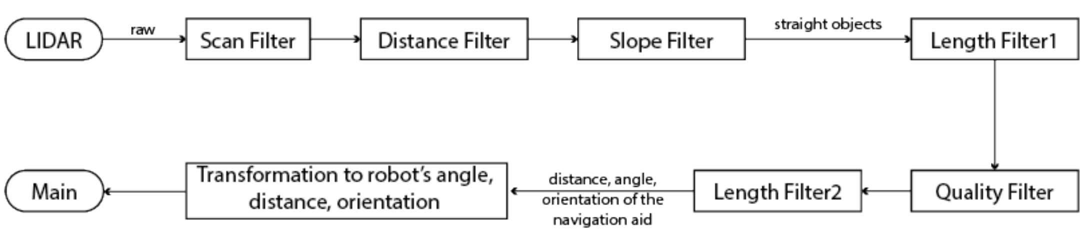
Implementation: As stated in the previous phase, the Hokuyo UST-10LX LIDAR range finder is used as a data acquisition device. This LIDAR device has high range (10m), and high angular resolution (0.25 degrees). This is especially helpful when the robot is too far away from the navigation aid, which is located at the arena's collection bin. After each scan, the Hokuyo LIDAR returns raw data in forms of angle, distance and reflection quality of the surrounding objects. The goal of the implementation is to search for the angle, distance, and orientation of the navigation aid from the raw data. Before discussing our searching strategy, we shall talk about how our navigation aid looks like.

The navigation aid is a form board whose dimensions are is about 500 mm x 900 mm. It consists of two equal length, distinct regions: one of high reflection quality (white) and one of low reflection quality (green). The board is designed to be completely flat for simple computation. It is light and requires no electrical power to operate.

The strategy for localization is to place the navigation aid on the ground at the collection bin and use it as a reference point, e.g. the origin (0,0). Since the Hokuyo LIDAR is placed on the back of the robot, which limits its view from -90 degrees to 90 degrees, the robot needs to face the mining area at all time so that the navigation aid is always in the LIDAR's view. This is perfectly compatible with our mechanical design, which has a mining mechanism in the front and a delivery system in the back.

The LIDAR operates continuously during the mission to obtain the robot position and orientation. Each scan, the localization program processes raw data through a series of steps to search for the navigation aid. In general, our localization method consists of two major steps: 1) to find all straight objects whose length is approximately 900mm, 2) among those straight objects it looks for an object that has two equal length, but distinct reflection quality regions. As a result, distance, angle, and orientation of the navigation aid relative to the robot is obtain and will then be transformed to another Cartesian coordination system where the midpoint of the navigation aid is at the coordinate's origin. Below is the detail process of robot localization:

The approach follows the below steps to filter for the navigation aid:



1. **Scan angle filter**: Input to this filter is raw data collected from the LIDAR, one of which is the scan angles. To be considered a single object, the scan angles of adjacent data points of that object must be spaced out equally. For example, figure A shows that two objects that are sorted out the scan angle filter based on an abrupt change in scan angle from β, note that all other angle gaps are approximately the same (α).
2. **Distance filter**: Getting input from the scan angle filter, distance filter further classifies the clusters to many sub clusters based on the average rate of change of the distance between the adjacent points. Going counterclockwise, if there is an abrupt change the rate of change of distance between the adjacent points, the distance filter will separate the cluster into two sub clusters and continue going counterclockwise. As an example, figure B shows a

situation where the scan angle filter fails to identify two clusters because all angle gaps are the same (α). However, the distance filter can look at the abrupt change in distance and recognize the change. Within a certain tolerance, the distances ι are considered equal, but the distance lambda exceeds the tolerance, which splits the cluster into two sub clusters.

3. **Slope filter**: Similar to the distance filter, the slope filter looks at the rate of change of the slope between adjacent points. An abrupt change in the slope will cause the slope filter to separate the cluster into two sub clusters. To illustrate, figure C show a possible scan data where the distance filter fails to detect the difference because the distance between adjacent points are all the same ι. This is when the slope filter comes in and clusters the data points.

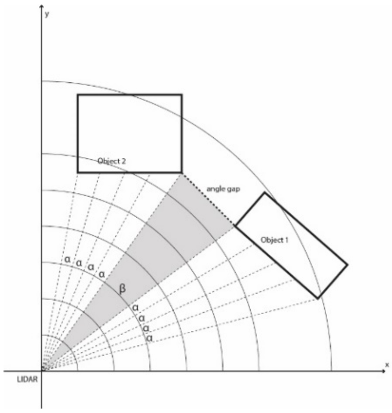


Figure A

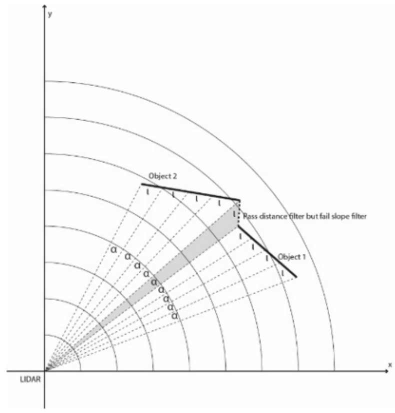


Figure B

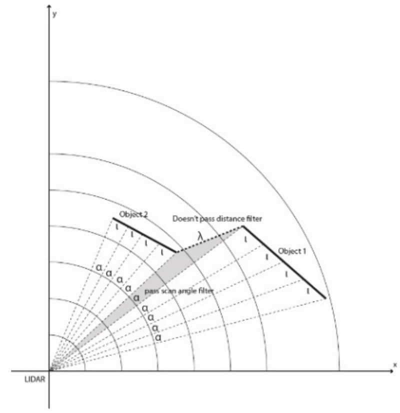


Figure C

4. **Length filter 1**: Up to this point, all clusters created by previous steps should contains only a single object whose scanned surface is straight. Given that the navigation aid is about 900mm long, all straight objects that are longer than 900 mm are eliminated; all objects whose length is less than 600 mm are also eliminated by the length filter. The min length was set to a very low value because only small portion of the object can scanned when it is too far away from the LIDAR. Figure D and figure E show the clusters before and after the length filter 1 is applied to the data points.

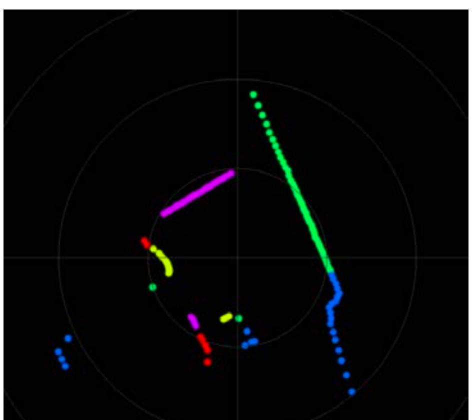


Figure D: Red data points are eliminated from the previous filters, and other colors indicates different straight objects in the test area.

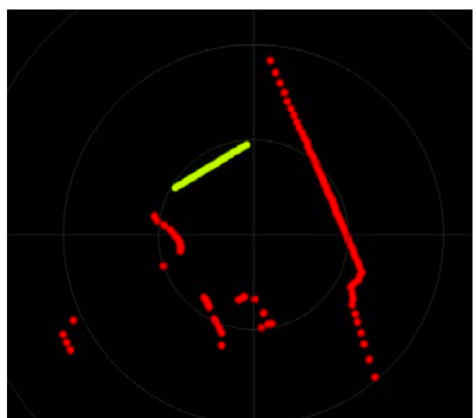


Figure E: Yellow cluster is the navigation aid detected, and the red data points is everything else.

5. **Quality filter:** There might be a chance that there are more than one straight objects detected in the arena, so the quality filter takes advantage of the fact that the navigation aid consists of two equal length, but different in reflection quality regions. It makes all clusters contains an abrupt change in reflection quality to the be candidate for the next filter.
6. **Length filter 2:** Given that the iso-quality regions on the navigation aid has approximately the same length. Length filter 2 will conclude a cluster to be the navigation aid by comparing the length of two iso-quality regions in the cluster.

After going through all above filters, the output consists of a list of data points taken from the navigation aid whose angle, distance and orientation will be calculated and transformed to a Cartesian coordinate system where the marker midpoint is the origin. The robot location is now ready for pathfinding and obstacle detection.

2.3.8.2 – Obstacle Detection

Purpose: For a system to be successfully autonomous, it must not be hindered by any sort of obstacle on the mission to its destination, and thus must be able to effortlessly avoid obstacles. The task of navigating through the obstacle field is a multi-step process which includes obstacle detection/localization as its first step. The main goal of the obstacle detection step is to locate all the possible obstacles on the obstacle field and retrieve their corresponding (x, y) coordinates with respect to the robot. These (x, y) coordinates will be fed to the next step to prompt the robot to traverse a specific path, avoiding all possible obstacles on its way.

Implementation: There are three obstacle detection methods have been researched: LIDAR based, distance sensor based, and Microsoft Kinect based obstacle detection. LIDAR based obstacle detection will be used on the robot, and distance sensor-based obstacle detection will work in parallel with the LIDAR as a secondary layer of detection. Microsoft Kinect based obstacle detection has been abandoned because it was found to be ineffective.

- **LIDAR:** Using Python and its appropriate modules to work with LIDAR, arrays of angle and distance from LIDAR to each point detected are returned as raw data. A radius of 2-3 meters (adjustable) is used as a first attempt to remove unnecessary noise. That is, only points within 2-3 meters of LIDAR will be considered for all the processing steps. For all points that satisfy the radius condition, their corresponding angle and distance arrays will then be used to calculate their (x, y) coordinates with respect to LIDAR. Then, an angle clustering is applied to the data; this filter will cluster data points within a specified angle range to the same group, with each group being a possible obstacle. After that, for each group found, a distance clustering, based on the distance between one point and another, is applied to remove noise from each cluster.

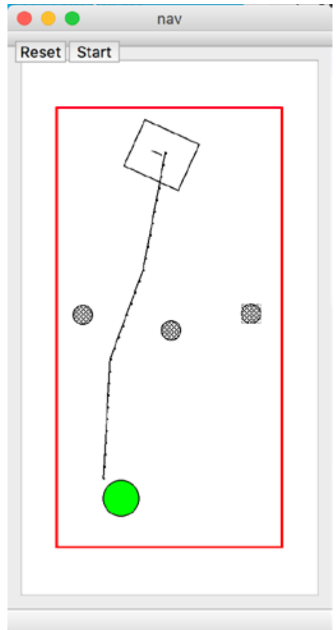
  After the two previous processes, we now have a group of clusters where the points in a cluster are "close" together, in terms of angle and distance. However, not all clusters are obstacles - for example, a cluster of many points (10-20) that are vertically straight can be the side-wall that the LIDAR detected. Thus, another step, to determine which clusters are truly the obstacles, is needed. To do this, we take advantage of our knowledge of the environment and how the LIDAR is setup on the robot. LIDAR will be situated such that it's some few inches higher than the ground. Thus, in this position, for any surfaces that curve up from the ground (rocks for example), LIDAR will only be able to detect about 3-6 points belonging to that surface. Using this knowledge, a lower bound and an upper bound, on the number of points a cluster can have, are set. If for a given cluster, its number of points fall between these 2 thresholds, it can be concluded that this cluster is truly an obstacle and the (x, y) coordinates of such obstacle are returned.

- **Time of flight distance sensors:** The time of flight distance sensors will be used as a fallback if the LIDAR doesn't work during the competition. The Adafruit VL53L0X Time of Flight Distance Sensor can detect distance to obstacle that is directly in front of it from 50mm to roughly 1 meters. To use these sensors for obstacle avoidance, we will place four sensors in the front and back of the robot (8 in totals). One of the sensors will be pointed straight forward (at the right angle) and the other will be placed at a 60 degrees angle from the robot's forward direction. With this placement, we will be able to detect if there's any obstacle that is directly in front of the entire robot and try to calculate its location with respect to the robot.
- **Kinect:** The system takes as input continuous frames of depth map. Based on the depth map and Kinect default parameters, a point cloud, which states the exact location (x, y, z coordinates) of each point in the depth map with respect to the Kinect, will be created. The point cloud will then be down sampled through Voxel Grid Filter. Then, the ground will be segmented from the point cloud by using Random Sample Consensus algorithm (RANSAC). The point cloud now is left with only the obstacles. Finally, Euclidean Cluster Extraction is used to identify each obstacle and its location.

Result:

- **LIDAR:** LIDAR has been able to detect a roughly 15 cm high x 30 cm wide rock with a relatively high level of confidence. Further tuning is needed to make the system more accurate.
- **Time of flight distance sensors:** The distance sensors hasn't been implemented but has been used for range testing. The algorithm for location detection has also been written out and waiting for further implementation.
- **Kinect:** The Kinect has success in this task of obstacle detection. However, due to inconsistency of the results, which stems from the poor depth image quality, as well as inefficiency in terms of speed. Therefore, the team has decided to move on with LIDAR for obstacle detection.

2.3.8.3 – Pathfinding

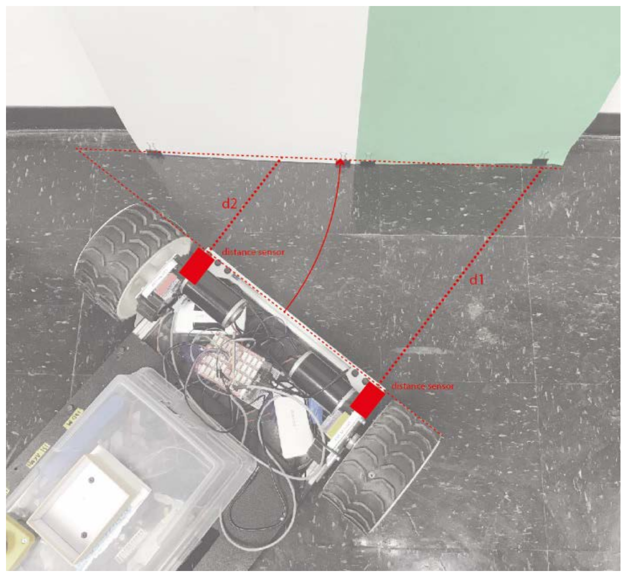


Purpose: Navigating the obstacle field is one of the most challenging parts of the competition. Since the robot does not have high ground clearance, it must avoid the obstacles in order to safely pass the obstacle field.

Implementation: As soon as the competition run starts, the robot performs a series of maneuver to look for the navigation aid. When the navigation aid is in the LIDAR's range, the robot will then starts navigating to a designated mining spot in a straight path until It sees the first obstacle. As it is approaching the obstacle field, our obstacle detection will probably notify the main program about the detected obstacles. The main program will then emit a signal (as in Signals and Slots of Qt library) telling the path finding sub program to calculate a new path to the designated mining spot with the obstacles are registered in its maps. This path calculation will be repeated every time there is a new event related to the obstacles, such as a new obstacle is found.

The path finding algorithm itself is an implementation of A* algorithms in Python. Input to this program are the coordinates of the robot, the destination, and all obstacles. It will output a list of coordinates to which the robot can follow to reach to the destination. The figure to the right shows a screenshot of the simulation program for pathfinding. In figure, the green circle represents the destination, three small circles are obstacles and the square box is the robot.

2.3.8.4 – Self-Alignment



Purpose: This sub-project was initiated because it's been observed that one of the 2017 RMC competitors delivered all its load outside of the collection bin because their robot did not approach the collection bin close enough. As a prevention of that problem, self-alignment was developed for IIT robot to ensure the robot is at a perfect distance and angle respect the collection bin before releasing its load.

Implementation: Two distance sensors are placed in two back corners of the robot to measure its distance to the bin once the robot has come back to the dumping area. The implementation of self-alignment consists of two PID loops, each governs distance of one side of the robot. Distance readings are fed to the PID loops to acquire motor speed needed to get the corresponding side of the robot to the set distance. In this operation, the left and right motors operate independently based on how far they are from the set distance. Figure to the right shows a situation where the robot is in front of the collection bin facing away from the collection's center. In activating self-alignment algorithm, the robot will adjust left and right motor speed so that both distances d1 and d2 are equal to a set distance pre-defined to the program.

2.3.8.5 – Measurement of Collected Materials

Purpose: The purpose of this sub-project is for the robot to know how much gravel/regolith it has collected.

Implementation: Sensors are placed at the opening of the collection bin on the robot with the sensor facing the bottom. The setup allows the sensor to measure the approximate high of the material level in the bin. Rolling average algorithm is applied in the calculation of distance data to filter out the frequent noises gained while getting the materials into the collection bin mounted on the robot, providing an accurate reading of material level.

2.3.9 – Autonomy: Motor Control Module

2.3.9.1 – PID

PID (Proportional – Integral – Derivative) control system is being implemented for speed and position control of the actuators. The PID system can command the robot to move an exact distance, at a specific speed or to a certain set position, and repeating it constantly, regardless of the load or disturbances. PID calculations is handled by a motion controller, named Kangaroo x2, which is attached to the motor drivers, Sabertooth 2X32. This motion controller helps reducing a significant processing burden on one of the robot's microcontrollers, an Arduino Due. The Arduino Due will be sending desired speed and position to the motion controller via UART serial, the motion controller will then maintain the set speed/position on selected motors until the next speed or position command is received.

2.3.9.2 – Speed Control

Purpose: As an important part of autonomous operation, the robot must be able to control its own speed precisely. If an external force is applied on the wheels which causes the wheels to stall or slip, current supplied to the motor will be increased or decreased to make sure the wheel rotation is maintained at the set speed. This allows the motor to achieve very high torque at low speed.

Additionally, the ability of the drive system to maintain any given speed allows the wheels to have a slight degree of turning angle and to move a short amount of distance. The advantages of these features are shown when the robot must maneuver along a complicated path in the obstacle field.

Implementation: Speed control is mainly used in the robot drive system. Incremental encoders, attached to each motor shaft, are used to measure the speed of the wheels. The wheels' speed is then fed to the Kangaroo X2 motion controllers, which calculates the motor power needed to achieve or maintain the set speed. Notice that the Arduino Due does not involve the P.I.D. calculation. Instead, it plays the role of a middle man between the main computing unit, Asus Tinker Board, and the motion controllers. The justifications for this setup are 1) motor control library is not available on Asus Tinker, and 2) It allows the Arduino to share some processing burden with the main computing unit.

Differential drive is used for the motor drive system. A major difference between this driving method and its predecessor is only one function will be needed instead of separating the drive and turn functions into two. Both motors on the left and right side will receive individual set speed whenever the Asus Tinker Board sends a speed command to the Arduino Due. This driving system allows the robot to move in a linear motion or turn easily. Besides, the motor speed is scaled to a range of -100 to 100 with 0 being the neutral position, 100 will drives the robot forward at its maximum mechanical speed and vice versa for -100.

For the robot to follow a linear motion, motors on both side will receive the same set speed commands, so all wheels will rotate towards the same direction and hence, moving straight. There are three types of turning situation depending on the need of the robot's turning speed. When the robot only requires a small angle of turning, such situation appears when the robot is positioning itself according to the collection bin, only the motors on the side that the robot is turning toward will receive a set speed command, other motors will remain stationary. When the robot is required to turn while moving, the motors on the side that the robot is turning away from will received a slower speed depend on the scale of turning speed. As such, the robot can remain its motion and turn concurrently. The last condition happens when the robot is required to rotate at a wide range of angle, both motors will receive the same speed setting but turning towards opposite side depend on the direction the robot is turning to.

Ultimately, the motion controller will take set speed from the Arduino Due and quickly change the motor speed to the said speed and maintain until next command takes over.

2.3.9.2 – Position Control

Purpose: To control the position of slider, which is where the auger will be mounted on, a pair of linear actuator is used. Position control is implemented using the PID Control for precise control over the extension of linear actuator, which will place the slider to the desired position for the auger to start mining. The X2 Kangaroo Motion Controller oversees the closed loop control system. The Linear actuator will also move to the set position while maintaining the given speed even if external forces are applied on opposite direction and remain still when target position is reached.

Position control is also applied on the driving system to allow the CIM motor to turn at certain angle or to move a certain amount of distance before stopping. This plays an important role in aiding the pathfinding algorithm for Autonomous Navigation System. Frankly, precise position control over the motor system will smoothen the process of obstacles avoidance.

I/O: The system take feedback from the potentiometer attached to the linear actuator. The motion controller will take input/set position from the Arduino Due and speed to move the linear actuator to the set position with the set speed and maintain at the position until next command gets it. The extension of the linear actuator is scaled down to a range of 0 to 100 for convenience purposes.

Implementation: The linear actuator pair will extend to tilt the slider that carries the auger to an optimize angle, which in turn allow the auger to drill into the land and drill at maximum efficiency. If the auger is having difficulty in drilling due to the hard surface, the linear actuators has an option of extending slowly over a period, which allow the auger to face the ground at a certain angle and drill from the side.

2.3.10 – Autonomy: Communication Module

2.3.10.1 – Robot Control Software

Purpose: Total control of the robot and monitoring its data is crucial for a successful off-world mining operation. The robot control software was built to control the robot in manual mode as well as monitoring the robot's health and its operations in autonomy mode. If there is something abnormal happens to the robot, the Earth screw will be notified right away so they can take immediate actions. The robot control software features the followings:

- Intuitive
- Fast, responsive and reliable.
- Minimal usage of processing resources.
- Minimal utilization of bandwidth.
- Capable of controlling all robot's actuators individually or together.
- Capable of monitoring the robot's current operation and sensors' data.

Implementation: The robot control system consists of two programs: one being a server program running heedlessly on the robot's computer, an Asus Tinker with TinkerOS, the other one being a client program running on a remote laptop. As soon as the server program starts, it establishes a TCP server and starts listening to incoming connections. The earth screw initiates the connection by clicking the "Connect" button, which automatically detects the robot's IP address in its local LAN network and establishes the connection. From here on, the earth screw has total control of the robot. Notice that no authentication was implemented because it is not a concern of this competition.

The robot control software was built based on the Qt library, a powerful library for making graphical user interface. The reason Qt was chosen to be the framework for the control software is because of its superior performance, fast development time, and the powerful slots and signals mechanism, which has been used extensively in both robot server and client programs. As an example, Qt's TCP socket increase processing efficiency by almost 100% comparing to build in Python TCP (Python was the programming language of choice and PyQt library was used). With a conventional Python TCP, both server and client programs requires to run an additional parallel thread checking for incoming messages to avoid blocking the main thread as it's waiting for a new TCP message. However, the QT's TCP solve this problem impressively, Qt programs run with a very efficient event loop, which monitor the all changes of the TCP socket. For example, a Qt's TCP client object will emit a "signal" whenever there is a new message, which trigger a function call (called "slot" in the Qt's world) to process the message. So instead of having an additional thread sitting there and wait for incoming messages, the Qt event loop can obtain the same result with superior performance and no additional threads.

Low bandwidth usage is one of the features of the robot control software. Per competition rule, the robot is suggested to use an average of less than 50kb/s of data bandwidth. To satisfy this requirement, all TCP messages are encoded into 5-byte blocks which are structured as follow (each field is one byte):

[command, device, value1, value2, checksum]

This message structure gives the earth screw more than four billion different options (256^4) to control the robot which is way more than enough for the robot operations. Although the message length could have been shrunk down to two bytes by encode more data fields in one byte, it is consuming more computing resources and doesn't not gain much benefit for this competition. Lastly, there is also provided a documentation on how to interpret and construct the five-byte message[11].

Lastly, the IIT team will also provides an intuitive graphical user interface (GUI) with the control software for visual enhancement. The purpose of this GUI is to display robot health and its sensors' data such as wheels' speed and the position of the linear actuators. It is also provided input methods to reset the robot or to change robot parameters on the fly. An emergency stop button is also included in case of mechanical failure and danger to human (this will stop all actuators but will not cut off their power; one needs to physically push the E-stop button on the robot to isolate of electrical power from the motors). A screenshot of GUI is presented[12].

2.3.11 – Critical Design Review (CDR):

On January 14, 2018 the team met with the Project Manager for the CDR. The Manager verified the maturity of the designs and the team's allotment of resources. The team's plan for system assembly, integration, test, launch and mission operations was confirmed to be sufficient to progress into Phase D.

2.4 – Phase D: System Assembly, Integration, Test, and Launch (SAITL)

2.4.1 – Mechanical System Testing

Drive:

- **Wheel deflection:** When the max load is applied to the robot, the team will measure difference between position of the frame relative to the ground with load and without.
- **Rotation:** To Determine the driving abilities the team will proceed to have the robot turn 360 degrees about its center to determine if the robot is able to turn.
- **Obstacle:** Crater will be made, and the robot placed in it with full load to test if the robot would be able to drive itself out of a ditch.

Digging:

- **Slide movement:** The first test is to see if sled can move in either direction on the slide rail system.
- **Angle of attack:** The second test is the team must also test to see if the actuators are able to hold and maintain the angle of attack of the drill.

Soil excavation:

- It is broken into two layers the drills ability to drill through the top soil, the top soil is 300mm deep.
- The second layer to determine if the auger can drill through and collect the rock below the soil layer.

If the robot passes all the verification tests stated previously then the integration between the excavation system and the drivetrain. The drill will be tested while mounted to the drive train. An issue foreseen is the torque from the soil drilling may cause the robot drive train to be thrown. If so protocols to counter this force must be implemented by the drivetrain.

Deposition system:

- **Interference:** The depositing system is tested first by itself. The conveyor is running without any load to ensure there is no interference with other components.
- **With load:** The team next verifies it can move the capacity it theoretically could move which is 20 kg. If the subsystem can deposit a 20 kg load of dirt or rock, then it can be integrated with the rest of the system. If not the max load of the system must be determined. Once determined the conveyor will be integrated with the rest of the robot. And field tested as a complete system.

Once the excavation and depositing system are operational individually validating them as a complete system is next. If the drill can successfully deposit soil and rock into the hopper it passes if not modifications and or adjustments must be

---

[11] See https://docs.google.com/document/d/1cUTG8RFGPtx6UG5p6J76NQImksJC-LNIKzTHBX8p66E/edit?usp=sharing

[12] See APPENDIX A, Figure 11.

made. If the robot can drive back approximately 7 meters and deposit its load while being .6 meters off the ground it passes this and is therefore able to function as a complete system.

2.4.2 – Electrical System Testing

Electrical Circuits:

The electrical system has been configured to fit within the electrical enclosure, which will keep all electrical components isolated for the outside environment. The electrical team has created prototype boards to test any electrical circuits that have been made for specific uses of the robot. Circuits for stepping down voltages and for signal processing have been made and tested using breadboards. Once a circuit design has been tested for functionality and reliability on a breadboard the circuit is then moved to a solderable perfboard where it will be in the final stages of being integrated with the entire robot. The perfboard model of the circuit is then assembled with the rest of the electrical assembly and then every connection is checked for any faults. In parallel with the perfboard the circuit design is also being included in a printed circuit board design (PCB). The PCB is an attempt to reduce the footprint of all our assembled circuits onto one board which will increase reliability and ease of use during the competition. Once the design of the PCB is completed the board is then checked by every team lead for completeness, in other words whether all the required components requested by each team is included. Once the design is completed and checked the board is sent out to be manufactured for the test model. Once the board arrives the electrical team is to check the reliability of the board before implementing into the main electrical system. If the board passes testing it can then be integrated into the main electrical assembly, but if it shall fail the design must be corrected and sent out for manufacturing again. If the board fails to be completed and checked the team will resort to using soldered perfboards that were created previously.

Electrical Enclosure:

The electrical enclosure is made to house all the electrical circuits and components. Due to the harsh conditions the robot will face, the enclosure should be durable and completely dust proof to maximize the integrity of the electrical assembly housed inside. The space allocated for the electrical enclosure is quite small and serves as a challenge to fit all the necessary components. The design has been worked with the mechanical team to make sure that the dimensions fit the overall robot design. The preliminary design has also taken into consideration the ease of use and a modular functionality. To allow control signals to enter and exit the enclosure, the use of aviation grade wire connectors have been used because of their dustproof features, and the ease at which to connect and disconnect. While designing the electrical enclosure the components where constantly included in the design and checked to see if their presence fit the required dimensions. If the components do not fit in the design, they shall be arranged to be compliant with the dimensions or the box design must be adjusted to make room for the components. Once the box design has been completed it is manufacture and then the components are included inside the enclosure. Once all the necessary components are included the aviation cable connectors are mounted and the proper sealant added to the enclosure to block off any outside debris.

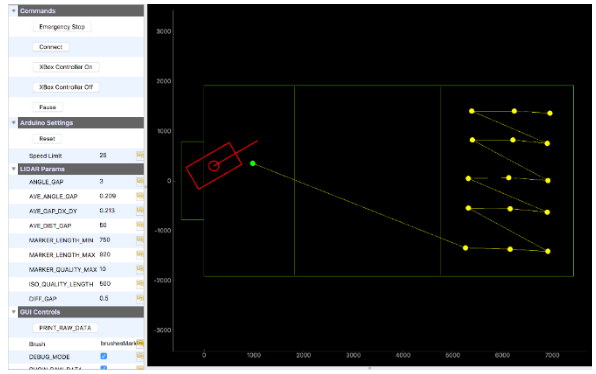
2.4.3 – Autonomy System Testing

Manual Control using Microsoft Xbox 360 Controller:

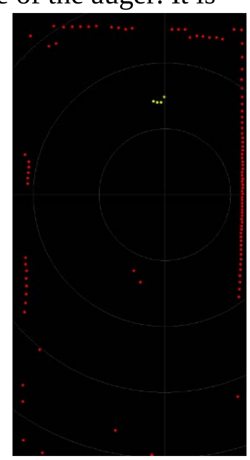
Xbox 360 controller is used as a fall back in case of autonomy failure. A prototype robot has been directly controlled by an Xbox 360 controller, and it’s proven to be reliable and responsive. Basics robot maneuver such as going forward, backward, and spinning on spot about its center was tested on the robot, in which the drive system performs superior to it procedure, a tank drive design. In addition, multiple team members also were also asked to control the robot using the Xbox 360 controller to make sure it is intuitive to most users.

Autonomous Operations:

- **Localization:** The localization program detects the navigation aid consistently at near distance. The output of the localization program allows the robot to rotate itself to a set angle or go any arbitrary position or the combination of the two. Since the LIDAR used in these testings, not the LIDAR will be used for final design, has low angular resolution. The localization program worked less effectively when the robot is too far away from the navigation. Finally, a PID loop for positioning the robot at a set angle or position needs to be tuned with better parameters to achieve smooth and accurate movements. Figure to the right shows a screenshot of the LIDAR testing program, which is built on the Qt library.



- **Self-alignment:** Self-alignment was tested and proven to be working. The robot was placed at an angle in front of a flat surface such as a wall and the navigation aid itself. When the self-alignment program was executed, the robot rotate itself so it is perpendicular to the surface and at a set distance away from the surface, which insure a successful dumping operation.
- **Path finding:** Path finding was tested on a computer simulation where it returns a path to a destination if the coordinate of the robot, destination and the obstacles are given. The reason pathfinding has not been tested on the actual robot is because it is unsafe to drive the robot on a bad path returned by pathfinding algorithm, which might harm human or damage the robot itself. As soon as pathfinding is proven to be reliable in the simulation, it will be tested on the actual robot. As far as the results concern, pathfinding returns good path about 40% of the times.
- **Angle of attack:** Two linear actuators are attached to the slide rails to adjust the drilling angle of the auger. It is essential to have the two linear actuators' position always stay in sync with each other so it will not damage the slide rails. Therefore, in addition to the PID loop in their motion controller, there is also another PID loop implemented on the Arduino Due, which periodically checks for difference in position of the two linear actuators and correct it. As a result, the linear pair never want out of sync more than 0.5 cm.
- **Obstacle detection:** Obstacle detection is of most concern now because no working demo has been presented. The team is moving forward with a LIDAR being used as the data collecting device, and distance sensors in the front and back of the robot will be used as the last safety layer, which inform the main program about the obstacle situation before the robot hits an obstacle. The picture to the left shows one of our successful tests of obstacle detection codes.



Tasks to be done before the competition days:

Below is a list of robot features that was not completed due to the lack of components, manpower, and the physical robot assembly. However, they are planed be done before the competition days.

- **Hopper:** Break beam IR sensors are on the opening of the hopper and the exit of the conveyor and will be detecting whether the material is coming into or out of the hopper. This feature helps the main program to determine whether it should change the drilling angle or if the conveyor should change its delivery speed.
- **Speed control and position control for the dumping conveyor:** An incremental encoder will be attached to the conveyor's motor to obtain speed feedback for a motion controller, Kangaroo X2. This implementation is quite straightforward and similar to the speed and position control in the linear actuators.
- **Speed control and position control for the slide rails:** Similarly, the slide rail system will also be implemented with speed and position control with the help of an incremental encoder attached to the sled screw. The purpose of this is for the main program to know how deep the auger has drilled down the ground and whether it is being stuck while drilling.

2.4.4 – Launch

Once all systems have been fully integrated and tested to operate within mission operable parameters, the robot will be packaged for transport. The Scarlet Space Hawks team will deliver the robot to the Kennedy Space Center where it will be field tested on a Martian simulated terrain. As described in the Concept of Operations, the robot will be evaluated on its ability to excavate frozen regolith autonomously. If autonomous operations fail, Mission control will take over, and operate the robot in manual mode.

2.5 – Technical Management

2.5.1 – Interface Management

To achieve full systems integration, sub-team leaders would meet weekly to discuss their team's progress. The primary goal of these meetings was to ensure that all sub-teams understand and executed their tasks in a timely fashion. Unification of sub-team designs was also tracked to ensure that all sub-team modules are compatible with each other. A project management spreadsheet was used during the design process to allow all members of the team to see in real-time the current progress of each sub-team and their respective tasks.

2.5.2 – Technical Resource Budget

Technological resources monitored during the design process were defined to be the mass of the robot, power usage, and transmission bandwidth. With weight constraints in mind, the mass of the robot is restricted to being no more than 80kg. Figure 30 shows a breakdown of mass by modules. Additionally, the nature of operating an extraterrestrial robotic vehicle requires that power consumption is monitored and kept to a minimum. The table below outlines overall estimated electrical power consumption. In the same realm as the previous two resources, transmission bandwidth is a vital resource to manage. Transmissions between Martian rovers and Mission Control here on Earth are not instant. This makes the relaying of movement commands, or system status delayed. Thus, transmission bandwidth must be monitored.

Power Usage	Subsystem Components	Arduino Due	Asus Tinker	Distance sensors	Drivetrain (motors x 4)	Linear Actuators x 2	Slide Rail Motor	Auger	Conveyor motor	Total	Power Usage per 10 min Trial Run
	Estimated (W)	<1	10	<1	480	60	60	200	60	870	145 Wh
Projected Mass by System	Module	Drivetrain	Excavator	Delivery System	Electrical Enclosure and Sensors	Overall Projected Mass					
	Estimated (kg)	20	30	5	5	60 kg					

2.5.3 – Risk Management

Risk management is extremely vital in the engineering process. All potential system failure risks, whether mission critical or not, must be identified and mitigated. Many failures are easily identifiable early in the system design. However, as testing and integration takes place, additional failure points may become clearer. The table below presents a way for the System Engineer to easily plot out and manage risks as they present themselves.

Class	Severity	Description
4	Complete Mission Failure	The mission can no longer be completed.
3	Greatly Reduced Capabilites	The mission is not lost, but probability of success is lowered. Extreme care must be taken thereafter to prevent further system failure.
2	Reduced Capabilites	Not immediately detrimental to mission success. Redundent systems or methods will be utilized
1	Non-Critical	Negligable endagerment of mission success. Redundent systems or methods are not needed.

			Failure Classification			
			1	2	3	4
	Likelihood	4		Autonomy Failure		
High Risk		3				
Medium Risk		2	Sensor Failure	Extraction Method Failure		
Low Risk		1	Wheel Slippage			Delivery System Failure

2.6 – Project Management

2.6.1 – Schedule

To more effectively monitor the team's progress, sub-team leaders made use of cell spreadsheets as a schedule. With this, each team leader could assign tasks and deadlines to their team members. As can be seen in the picture[13], specific tasks were broken down into further subsystems and a student engineer was assigned to each task. Team members could update their progress as a percentage. The spreadsheet also displayed deadlines set by the team leaders. Additionally, a color-coded status bar defined the current state of the task whether it be in progress or waiting on another task for example. This simplified Gantt chart allowed the Systems Engineer and team leaders, closely monitor the overall team progress.

2.6.2 – Management Structure

Managing a multidisciplinary team has required the team to structure itself accordingly where their multiple sub teams with leaders to over watch the day to day activities within the project. The financial responsibility is held by the team treasurer who is responsible for obtaining the funds required for building and for travel. The outreach chair has the responsibility planning events that educate children, and adults, about S.T.E.M and its possibilities. To document the progress and standing of the team there is a social media representative that will log our progress and standing on social media pages such as Facebook, Instagram, and SnapChat. To oversee the sub team leaders and all other activities throughout the team there is an overall team leader who checks in with each sub team leader and see if the project developments are on course or if there must be changes made to the projected plan.

2.6.3 – Configuration Management

The team has based all its operations in a laboratory provided by the team's advisor, Dr. Mahesh Krishnamurthy. The team's tools, materials and testing equipment are housed in the power systems lab provided by the advisor. The team also has access to on campus machinist workshops where the robot frame and other necessary items have been manufactured. For preliminary designs our sponsors have provided software such as Altium and SolidWorks, which without none of our accomplishments would have been possible. To share work progress and communicate with members Slack, Google Drive, and GitHub have been used.

2.6.4 – Financial Budget

Funding for the robot came from the Illinois Institute of Technology. At the beginning of the year, the team's Treasurer and President submitted a funding request to the school indicating construction materials, tools and equipment, and manufacturing costs which were approximated based upon the initial Phase A design of the robot. It was projected that the overall material costs of construction were to be approximately, $15,000. At this point in the project, the overall construction costs are under par at approximately $11,000.

3 – CONCLUSIONS

Using the Systems Engineering approach combined with previous work and experience, the 2018 IIT RMC team has developed a modular system capable of performing multiple tasks depending on its module configuration. The final system validation and system launch will be the completion of a successful mission at the 2018 Robotic Mining Competition in May 2018.

The team expects this design to continue to evolve in the future as in previous years. As we inherited designs from last year, it is expected for this year to be further developed and polished in support of next year's evolution. It is the team's hope that NASA will find this design to be a viable option for future space missions and that this work will aid in the goal of utilizing space resources.

[13] See APPENDIX A, Figure 12.

APPENDIX A:

Figure 1

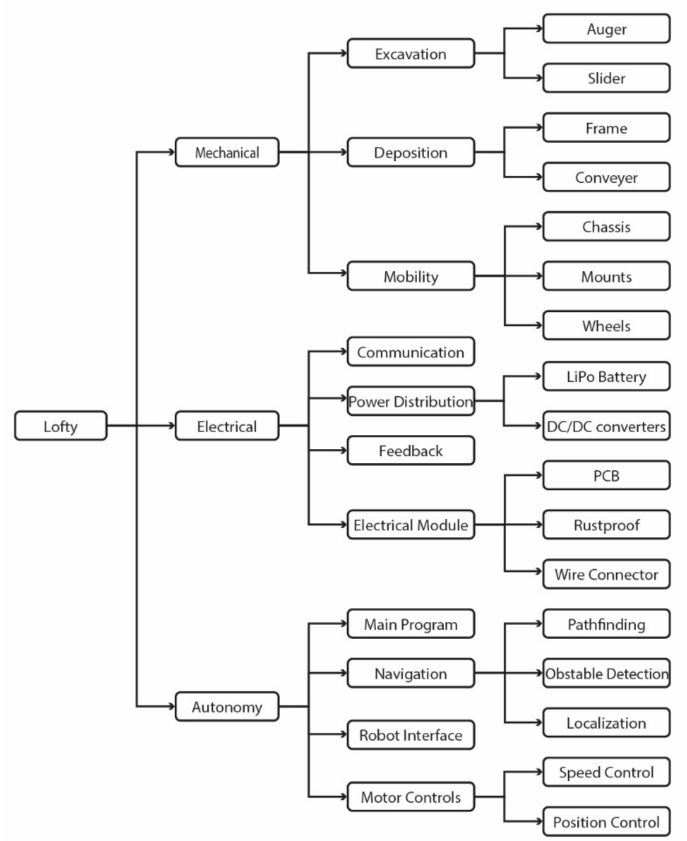


Figure 2

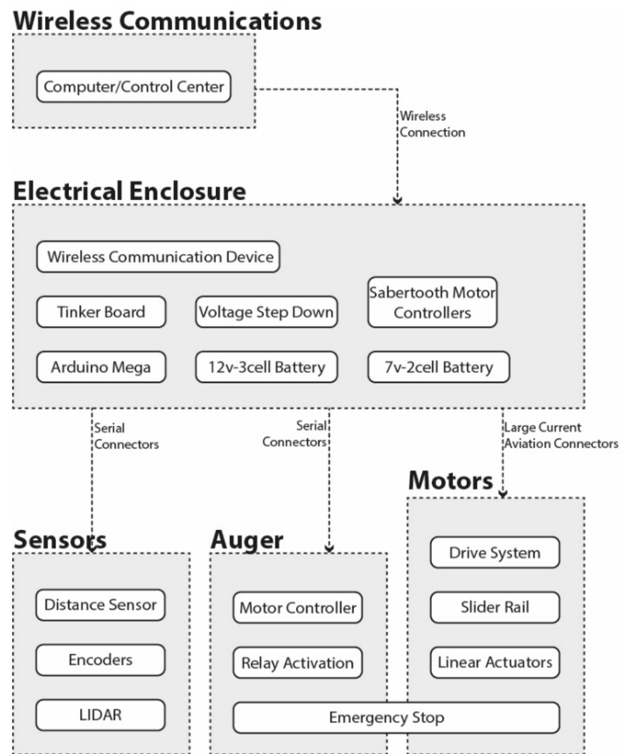


Figure 3

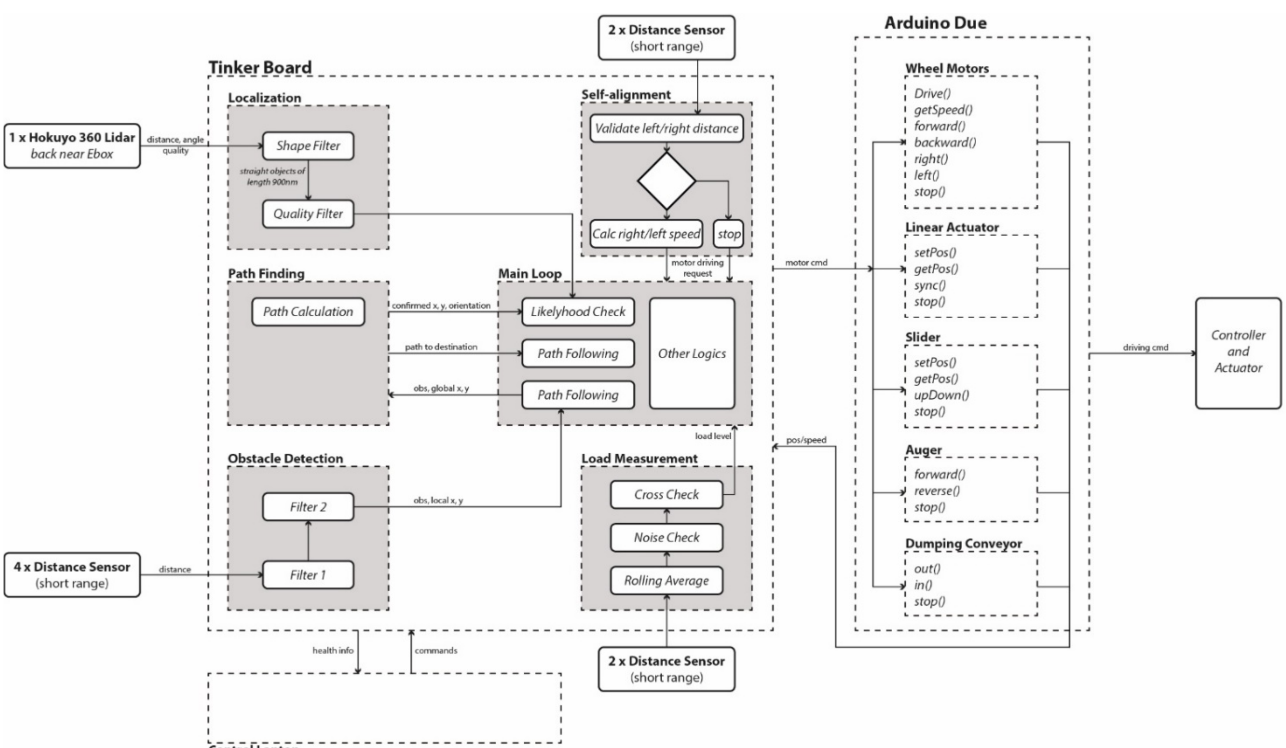


Figure 4

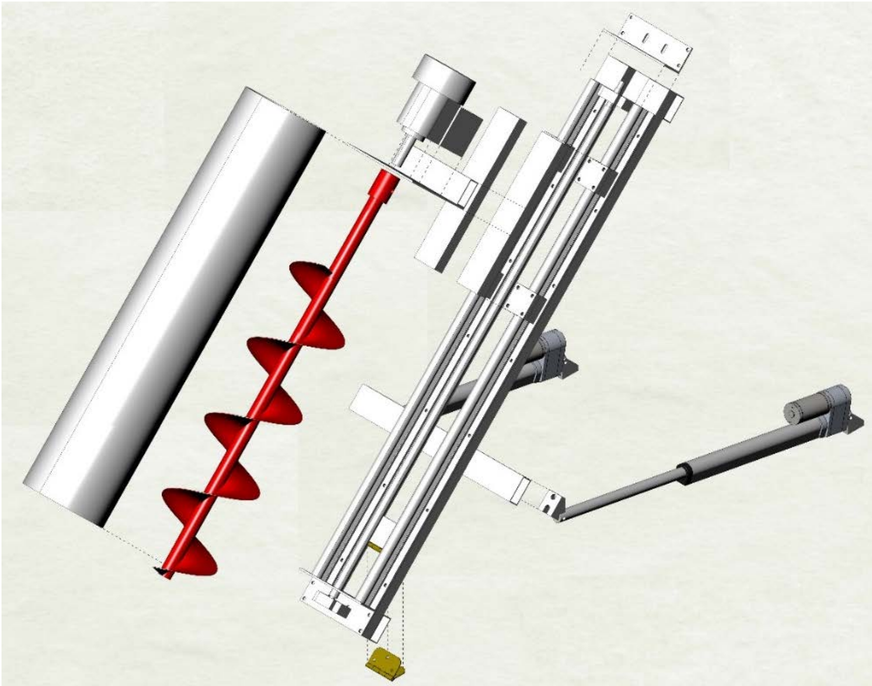


Figure 5

Figure 6

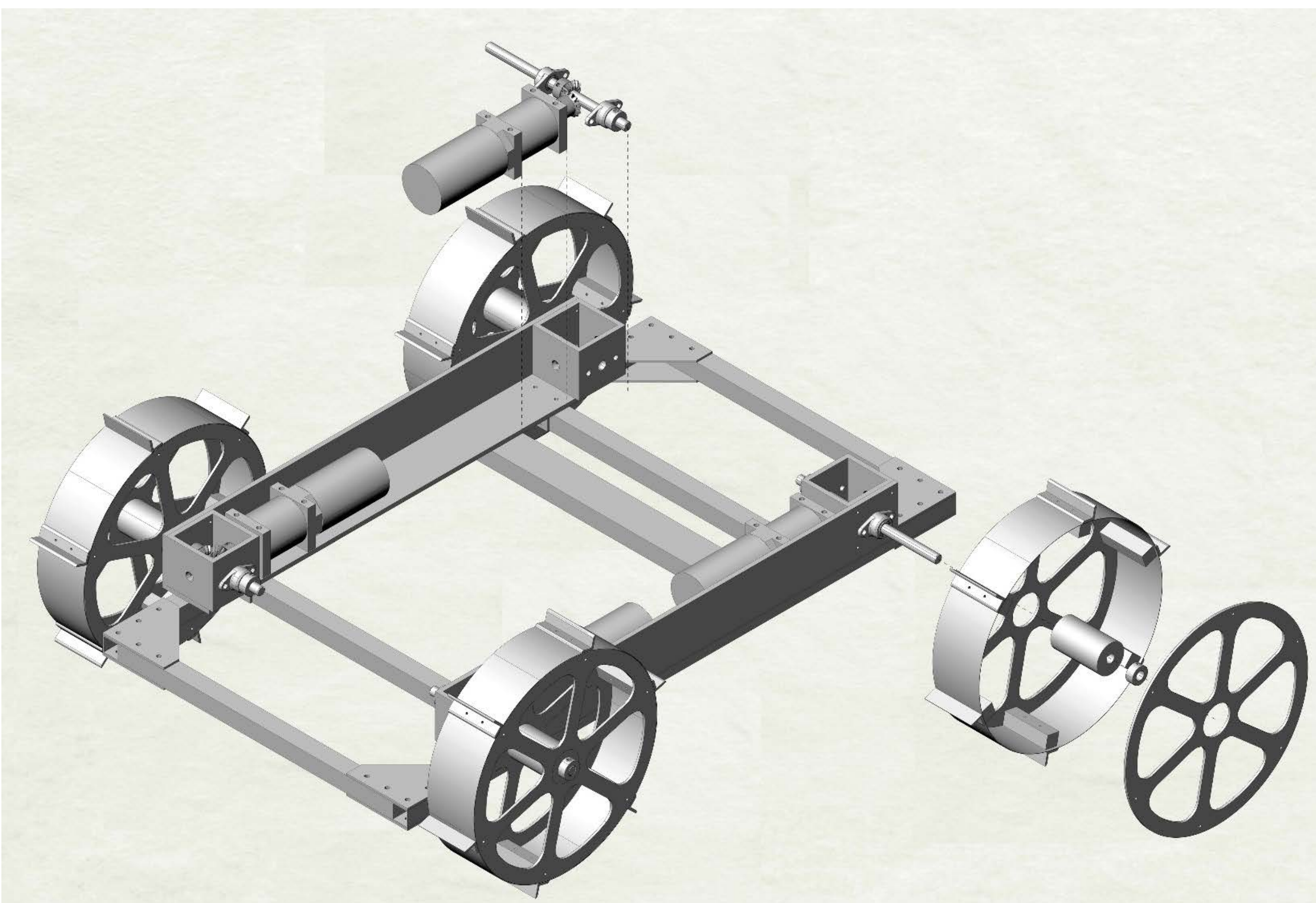


Figure 7

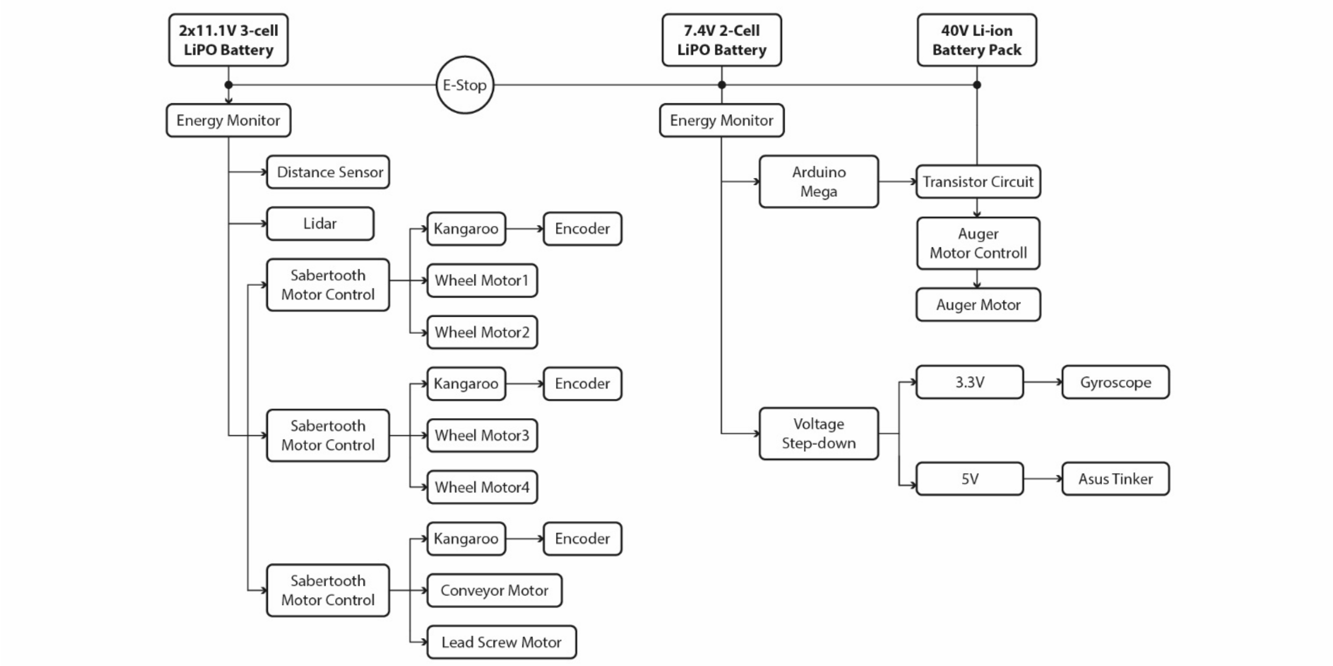


Figure 8

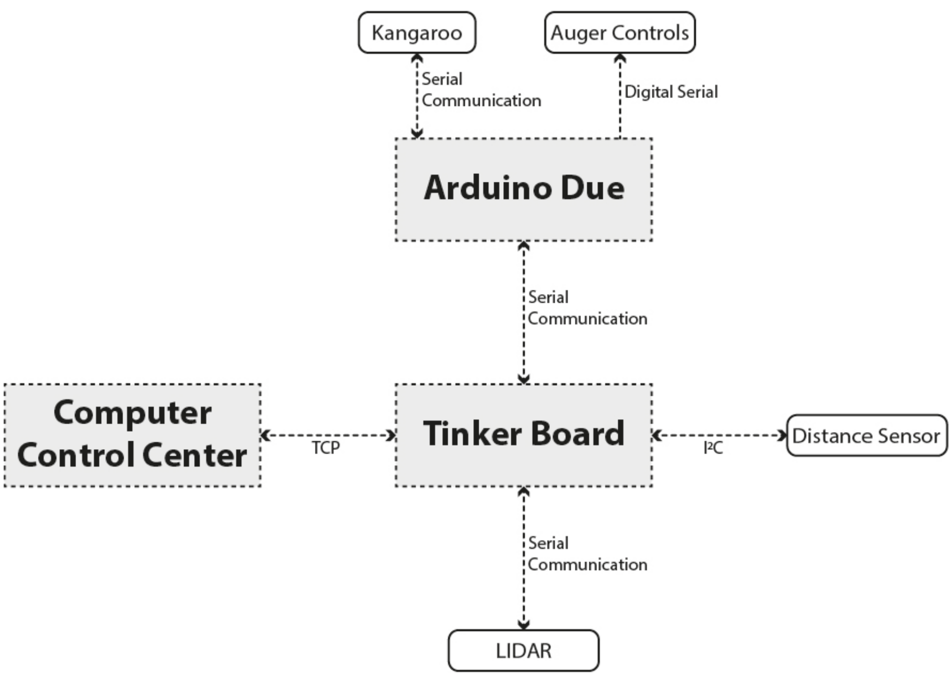


Figure 9

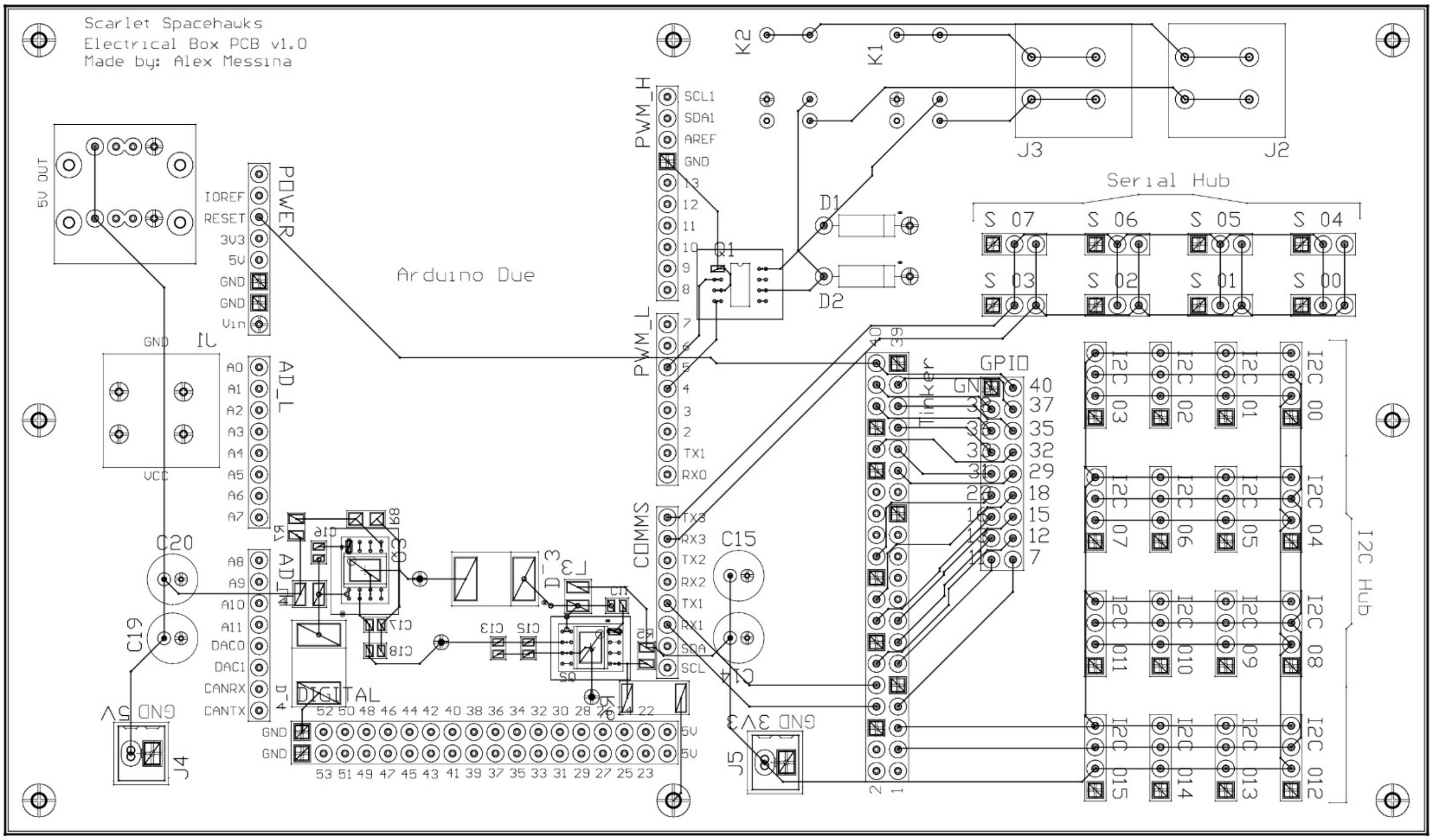


Figure 10

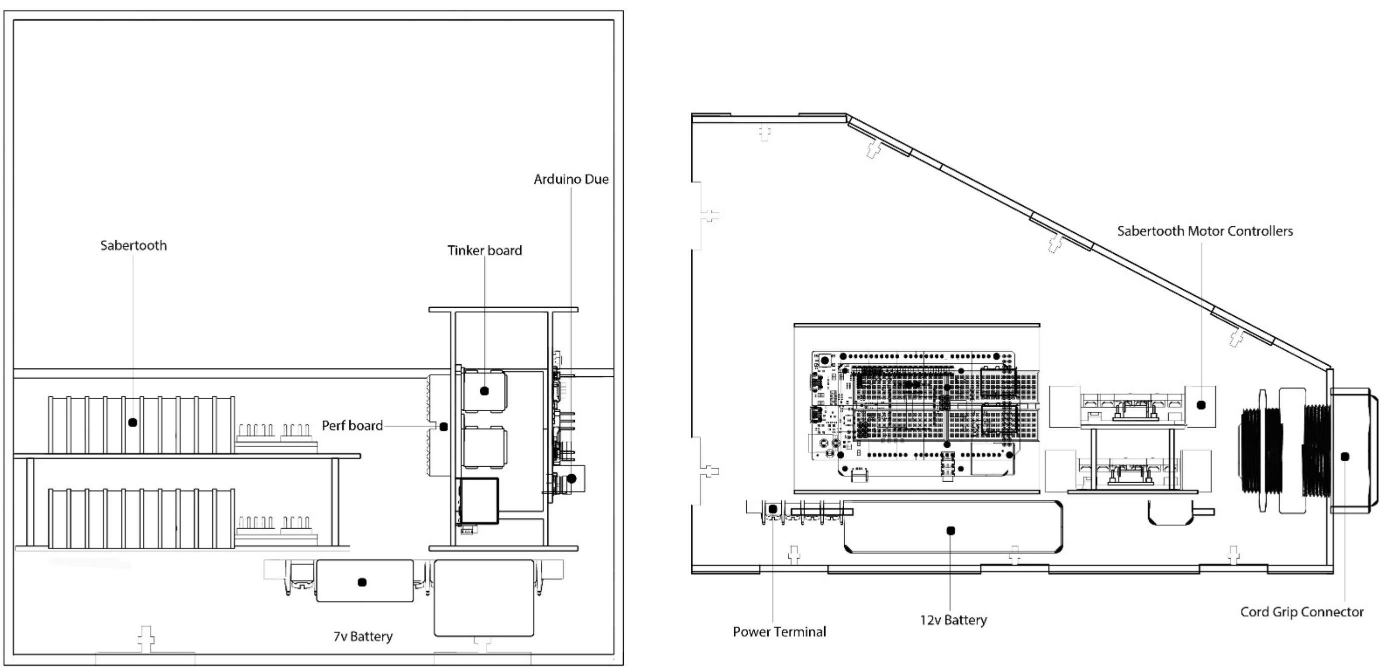


Figure 11

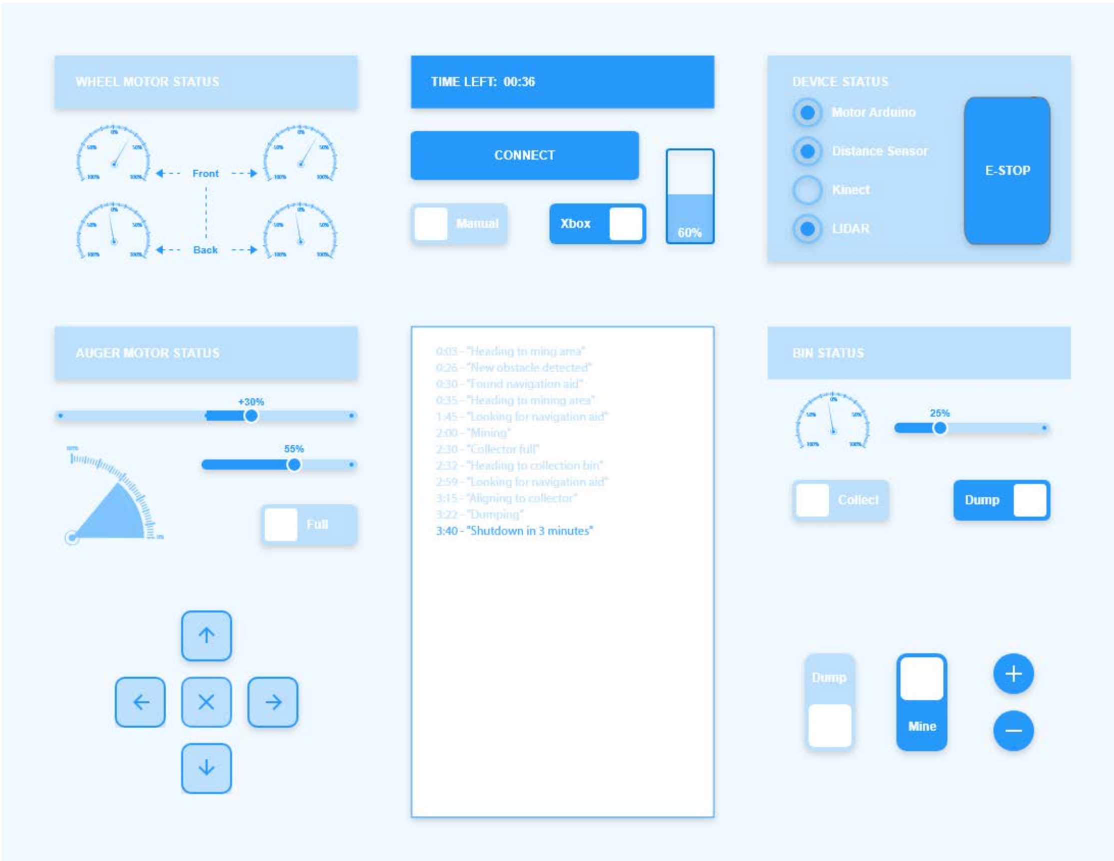


Figure 12

Autonomy Team			Team Lead: Hieu Nguyen		Percent Complete:	
Status	**Activity**	**POC**	**% Complete**	**Deadline**	**Comments**	**# days left**
	TCP Communication		**43%**			
	Write a Python class for sender and receiver	Hieu Nguyen	100%	1/1/2018		Done
	Tcp in C++	Clive	50%	**2/28/2018**		-28
	Optimize the class to increase performance	Hieu Nguyen	0%	3/17/2018		-11
	Add methods for robot and comtroller to interface with each other	Hieu Nguyen	20%	**4/27/2018**	**Ready for proof of li**	30
	Motor control / Prototype		**63%**			
	Positional controll for a single linear actuator	Yuzhe	100%	1/1/2018		Done
	Synchonization of a linear actuator pair	Yuzhe	100%	1/1/2018		Done
	Speed control for a single motor	Yuzhe	100%	1/1/2018		Done
	Speed control for the drive system. (4 independent motors)	Yuzhe	100%	1/28/2018	Prototype is ready	Done
	Skidding prevention	Yuzhe	30%	2/4/2018	Waiting for intergrator c	-52
	~~Auto brake~~	~~Yuzhe~~	~~80%~~	~~2/4/2018~~		~~-52~~
	Dumping positioning (distance sensors)	Yuzhe	50%	**4/27/2018**	complete C++ codde on Arduino Mega. PROBL	
	Incremental position tracking	Yuzhe	10%	2/4/2018		-52
	Path following using PID	Yuzhe	0%	2/11/2018		-45
	Navigation		**44%**			
	Path finding algorithms: found (A*)	Jackson	100%	1/1/2018		Done
	Simulating A*	Jackson	90%	1/28/2018		-59
	Localization using 360 lidar	Patrick	50%	2/24/2018	**Testing on old lidar,**	-32
	Kinect simulation on PC	Tung/Tuan/Trung	75%	2/28/2018		-28
	Implement kinect simulation on Raspberry Pi	Tung/Tuan/Trung	10%	3/7/2018		
	Obstacle avoidance using camera/kinect on Robot	Tung/Tuan/Trung	15%	3/17/2018		-11