Team Name: MAEvil Incorporated

Robot Name: Planned Parenthood

Wilson Moyer, George Kopf, Zachary Andrews, Ella Simmons, Chimwemwe Chinkuyu, Emmanuel Ishola, Joel Boon, Joseph Roberts, Madelyn Smoyer, Sawooly Li, Stanford Cramer

Executive Summary

This report presents "Planned Parenthood", a two-robot Search and Rescue Robot (SaRR) system engineered by the MAEvil Incorporated team to autonomously complete rescue missions in challenging terrain. The marsupial robot pair—Mother Bot and Baby Bot—offers a mechanically robust, highly maneuverable, and sensor-driven solution for precision medkit delivery.

The Mother Bot features a durable steel chassis with a four-wheel, two-axle, skid-steer drivetrain powered by two CIM 2.55 motors and two mini CIM motors. There is a 60:1 belt-driven gear reduction to each wheel. They provide ample torque to transport both the 90 lb Mother Bot and the 30 lb Baby Bot payload over a 15° wooden ramp and through a 3 ft wide twisting chute. It has a lifting mechanism that employs a rigid frame, sliding carriage, and winch assembly to horizontally extend and safely lower the Baby Bot 12 inches beyond the wall edge. The rear-weighted chassis and controlled deployment design preserve stability under shifting loads.

The Baby Bot is a compact 13 in x 14.5 in x 20 in, equipped with dual E-S 12V, 60 RPM DC motors, with a 2:1 gear reduction, for smooth navigation and medkit transport. Its spring-loaded four-bar linkage launches a 1 kg payload at up to 7.2 ft/s for accurate delivery into the receptacle. Lightweight construction and balanced geometry ensure reliable movement and survivability during drop testing.

Both robots use light sensors for autonomous navigation—guiding the Mother Bot up the ramp and through the chute, and the Baby Bot toward the medkit drop point and back to its recovery beacon. Ultrasonic distance sensors and limit switches provide spatial feedback for closed-loop control, managed by Teensy microcontrollers with open-loop testing capabilities.

The full system is under 3 ft wide, weighs approximately 120 lb, and cost $271.05, primarily from the steel frame, wheels, drivetrain components, and lifting mechanism materials on the Mother Bot. During our official robot demonstration, our robot showcased its ability to autonomously traverse the course, breach the wall, and deliver the medkit (though not all during the same run through the course due to an issue with our motherbot motor controllers)—showcasing a well-designed and innovative SaRR design.

I. Introduction

A. Objectives For the Robot

For our SaRR to be successful in accomplishing its task of operating a search and rescue mission in a complex and challenging environment, it must meet a wide variety of objectives spanning multiple mechanical and electrical domains.

Mechanically, the Mother Bot portion of the system is the primary method of navigating through most environments, including sloped ground and enclosed pathways. To do this, Mother Bot's drivetrain will need to be strong enough to carry the full load of the system (including itself, the Baby Bot, and the enclosed medkit) up and down slopes without slipping. Additionally, the chassis should be designed to maintain the overall balance of the system as it navigates these varying slopes, thereby preventing it from tipping. Mother Bot should also have a steering system that enables it to make turns through tight, enclosed spaces without hitting the walls. One final major objective for this portion of the system is mechanical robustness, especially concerning sudden, unexpected loads. When Mother Bot is subjected to an impulse load (such as our drop test), it is responsible for protecting not only its own functionality but also the functionality of the Baby Bot as well. As such, Mother Bot should contain provisions to insulate both itself and Baby Bot from the effects of this impulse.

However, the Baby Bot portion of the system should also be robust enough to operate on its own. While not responsible for as challenging of an environment as Mother Bot, Baby Bot should still be able to navigate while carrying the medkit. In addition, Baby Bot is responsible for the actual delivery of the medkit; as such, it should contain a system by which the medkit is able to be deposited at its final destination at the victim. As recoverability is a key feature in a reusable system, Baby Bot should also have provisions for reuniting with its deployment mechanism on Mother Bot.

Deploying the Baby Bot serves the primary purpose of allowing the system to breach any obstacle that the Mother Bot cannot conquer on its own, such as a vertical wall. As such, the deployment mechanism should be able to lift the Baby Bot up and over the wall, releasing it safely once on the other side. Another significant factor here is the balance of the whole system, as the deployment of the Baby Bot will cause significant changes in the distribution of weight in the system, so the deployment mechanism should also have a method to compensate and ensure machine balance. As noted above, we desire recoverability as well, and thus the deployment process should be entirely mechanically reversible, allowing the system to completely reunite the Baby Bot with the Mother Bot once the Baby Bot has concluded operation. Notably, the deployment mechanism is also the primary connection between the Baby Bot and Mother Bot prior to deployment, meaning that this system should be a significant part of the design to protect Baby Bot from impulses made to Mother Bot.

For electrical objectives, both robots need some way to identify light sources and navigate towards them. We are planning to accomplish this by using two offset photoresistors. Mother Bot needs to be able to navigate the chute and not fall off the ramp. Distance sensors on the side of and below the chassis should allow the robot a spatial understanding of where it is in the course. Both robots also need some way to identify when they arrive at their final destination. If light tracking is effective, a distance sensor at the front of the robots will be able to identify when they reach the wall or the medkit drop point, respectively. Then, Mother Bot needs to be able to detect when Baby Bot has returned for retrieval. The best sensors to use will depend on the final design of the recovery mechanism, but some kind of mechanical limit switch that Baby Bot will press seems to be the simplest way of managing this. Baby Bot can be directed back to Mother Bot by repurposing the light-tracking system. We will mount a light to the lifting arm, so Baby Bot can turn around and navigate back after the medkit is delivered.

The robots must be fully closed loop, which means there is a set of autonomous capabilities that each robot must display. Mother Bot must be able to autonomously track to the start of the course using light sensing, drive over the ramp without falling, navigate through the chute, detect when it encounters the wall, and successfully deploy Baby Bot over the wall. Mother Bot should also be able to autonomously retrieve Baby Bot upon Baby's return. Baby Bot must be able to autonomously track to the medkit delivery point using light sensing, detect when it encounters the basket, and successfully deliver the medkit. Then it must drive back to the Mother Bot, possibly using light tracking again.

The two systems can exist entirely separate from each other. The simplest method is to have both robots with distinct Teensys and batteries operating entirely without communication. Mother and Baby Bot both perform their standard duties, each assuming the other does theirs. Baby Bot can navigate back to Mother Bot by tracking a light fixture on the lifting arm. If the control code is robust, this method should be fine. In testing, if we identify that this method will not suffice, we can connect them. The robots can communicate wirelessly over esp32 or XBee, or they could even be tethered. In an ideal world however, this will be unnecessary.

B. Research

Mother-baby robots are not an altogether new concept; early examples include NASA's 1990 "marsupial rescue robots" initiative, in which large tracked carriers deployed micro-robots into confined rubble zones[1], and more recently, the ETH Zurich and NTNU team in the DARPA Subterranean Challenge developed the BSTAR–RSTAR system, where a four-legged mother robot deployed a rolling daughter robot for subterranean mapping and sample retrieval.[2] While many of these designs were largely successful in achieving their objectives, they also faced challenges that we will take into account when designing our own robot. The BSTAR-RSTAR found that vibrations during motion repeatedly caused the tail-mounted ramp to slip from its rotation axis, undermining deployment reliability. This emphasises the importance for our consideration of rigid mounting, vibration isolation and center of mass control during movement.

Several other accounts reported difficulty for the deployed robots to reliably return to the mother robot. In our case, however, the distance between the mother and baby bot will not be too large, but this will still be a component to be conscious of, and varying methods of return will be assessed, including radio communication and tether.

---

1 https://www.washingtonpost.com/archive/politics/1999/07/05/marsupial-robot-duty-rescue-in-pockets-of-rubble/e7843e08-85f2-496f-9724-cab7f513defa/

2 https://doi.org/10.3390/app10248767

C. *Design Goals and Principles*

To drive our design choices, our team established 4 guiding principles: feasibility, reliability, simplicity, and uniqueness. Feasibility, in this case, refers to how likely it is that our team is able to meet the requirements of a particular design with the limited resources available to us. In practice, this has involved straying away from designs that, while they could in theory work, would require so much extra effort to ensure that they work that it would exceed the amount of labor our team could reasonably provide. Reliability refers to how consistently we anticipate a certain design would perform under varied conditions—our design should be able to complete the required tasks successfully almost every time. With simplicity, our goal is to reduce the number of moving parts (both physically and metaphorically) and thereby the number of available failure modes of our design, allowing us to focus our efforts on fine-tuning what needs to be done. Last, but certainly not least, we use uniqueness as a measure of how much we have the ability to learn from this project. As this project as a whole is such a strong learning opportunity, we seek to take advantage of how much we can stand to learn from incorporating less-common elements in our design, gaining even more experience with the principles of mechanical design in the process.

It is the combination of these factors that led us to the marsupial design that we pursued. With regard to feasibility, the marsupial design (though slightly more difficult in requiring mechanisms to connect the two bots for deployment) can be broken into two otherwise fairly simple robotic systems, allowing the bulk of design effort to be in the transition between these two. Additionally, having a consistent deployment plan can be easily accounted for, leading to operations that should be fairly reliable. While the design of two separable systems does sacrifice some simplicity, each can on its own be quite simple, focusing our complexity once more on the marsupial element of the design. From a brief survey of past project designs for this course, this is also quite an uncommon design, allowing ample opportunity for us to explore new design territory. All in all, then, the Baby Bot/Mother Bot marsupial design is driven by our core design principles, providing a suitable challenge for our team to meet over the course of this project.

D. *Standards*

The team adhered to the ANSI B4.1 standards for tolerancing running fits, sliding fits, and location fits.[3] The team also decided upon a series of best practices to ensure the optimal operation of our robotic system. All components were CADed and added to the overall project assembly before manufacturing to ensure there was no interference between components or subsystems. Additionally, dynamic and static simulations were performed on the components and assemblies to confirm that there were no obvious design flaws before manufacturing and assembly. Components manufactured in the shop were properly deburred and cleaned to ensure a proper fit within the overall system and to avoid any injuries. The system was tested many times, leading up to the final demonstration, in order to identify any potential failure modes resulting from the wearing of components.

[3] https://www.engineersedge.com/mechanical,045tolerances/preffered-mechanical-tolerances.htm

For electrical standards, the team will adhere to a standard set of electrical best practices to ensure the safety of the build. All grounds will be connected together through a standard terminal block to ensure no floating voltages are present in the system. All wiring will be done with appropriate gauge wire, especially regarding connections through the battery. Components will be integrated with appropriate heat sinks if necessary. Most major wiring will be done via twisted pairs to increase robustness and minimize noise. Major power systems will have visible and accessible ON/OFF switches so that the robots can be easily turned off in the event of a malfunction. There will also be a remote kill switch built into the open-loop software. Shutting down the robot will result in the battery being hard-disconnected with a latching switch. Thorough testing of all connections will be done via multimeter. Additionally, extra care will be taken while charging and reattaching the battery to avoid shorting it and causing a fire.

E. Overall System Sketch

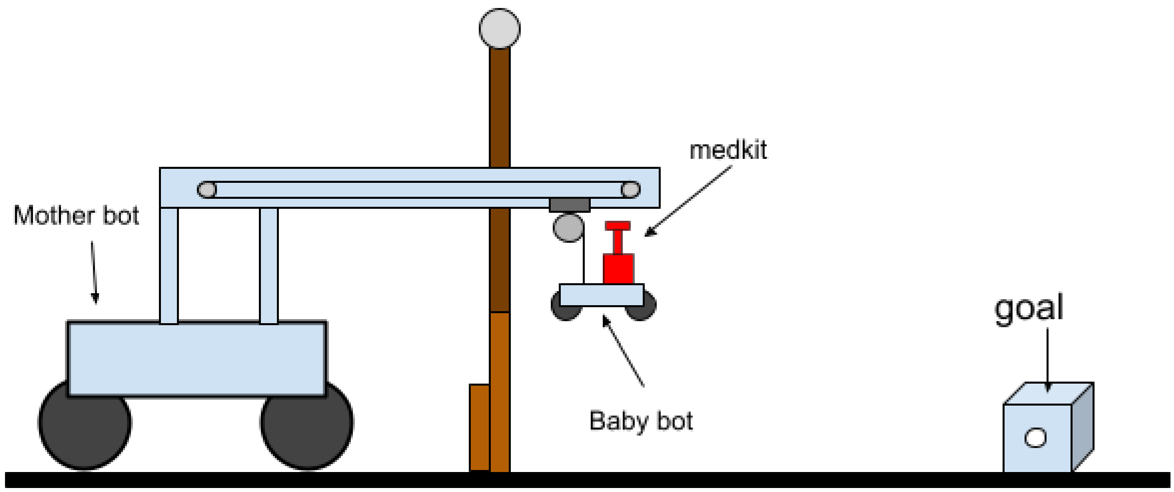
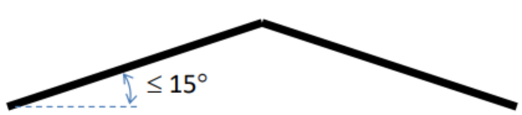


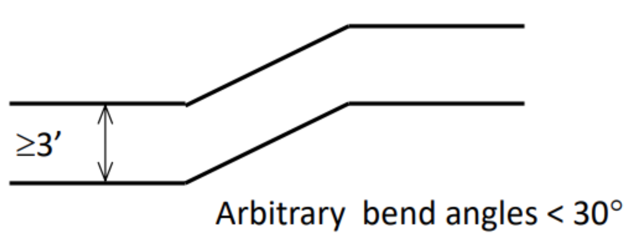
Fig. I.E.1: Sketch of the whole system in the process of deploying the Baby Bot over the wall, demonstrating its overall planned functionality.

II. Specifications

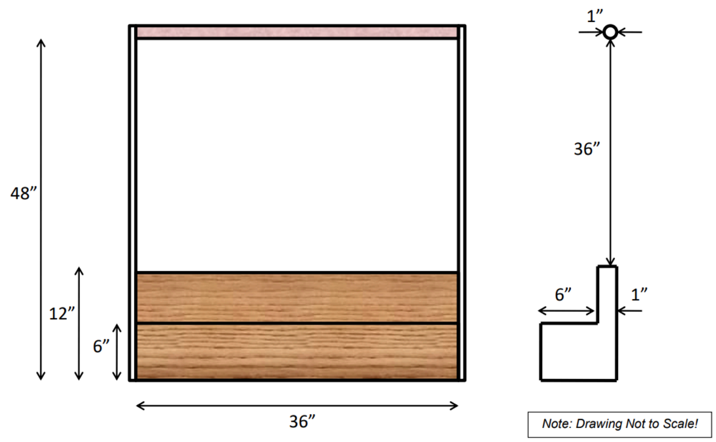
A. Mechanical Specifications



The specifications for our SaRR were developed based on the physical constraints of the course, as well as the functional requirements of the mission. Although there are no explicitly defined size or weight constraints imposed, several practical constraints exist that heavily influence the mechanical design. The Mother Bot must be narrower than the 3-foot-wide ramps, chutes, and wall openings, and be capable of navigating inclines up to 15 degrees and turning through chute angles under 30 degrees. The Motherbot was around 90 pounds including

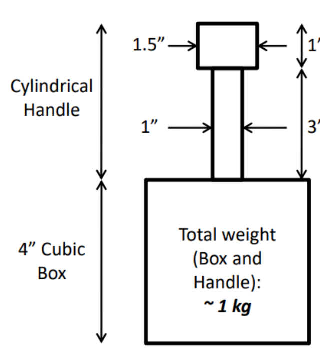


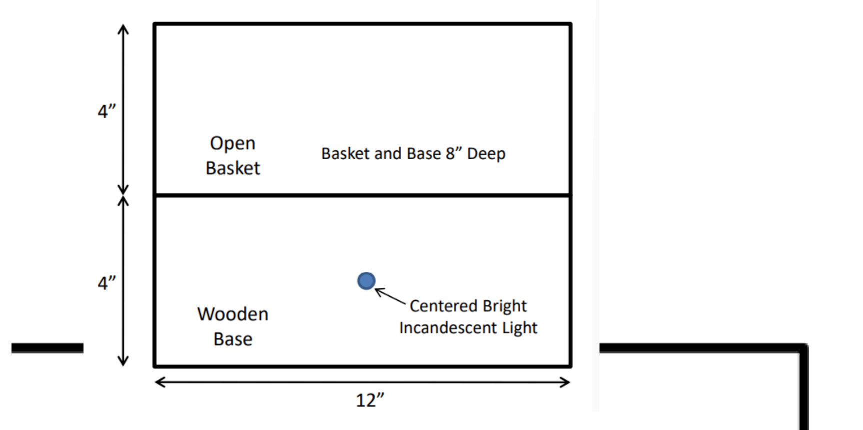
chassis, drivetrain, electronics, and lifting mechanism. The chassis alone was made from about 60 pounds of low carbon steel C channels that were welded together. Overall, Motherbot was 24” wide by 32” long including wheels.



Additionally, the lifting mechanism must operate within a 3ft x 3ft wall gap, capable of raising the baby bot at least 12 inches vertically and 7 inches forward, all while maintaining balance during deployment, especially given the shifting center of mass during lifting. The baby bot must be compact and lightweight enough to carry and adequately deliver the medkit to a designated receptacle. The lifting mechanism had four circular 1.25 inch diameter 36” long aluminum rods for supports, two on each side of the Motherbot chassis. On each side the supports were 20.75 inches apart and the two sides had 16.5 inches between them. The top of the rods were held together with steel plates that supported a 48” long steel rectangular tube. This had a notch cut though it for the ability for a carriage to move Babybot laterally past the wall after it was raised up. The lifting mechanism likely weighed about 25-30 pounds.

The baby bot is designed to weigh approximately 20 pounds, transport and accurately deliver a 1kg, 8’’ tall medkit and survivability during the drop test. It must deploy from the mother bot, traverse a shorter course segment, and deliver the medkit into a receptacle, and then either return autonomously or await recovery. Because both bots must survive drop tests, weight is indirectly constrained by the need to limit impact impulse; high mass would increase the stress during drop landings, so a lightweight design is essential. The delivery system uses a spring-loaded hook coupler mechanism, launching the medkit at 15 degrees with a required velocity between 3.63 ft/s and 7.2 ft/s, depending on the target distance. A spring constant of approximately 8.5 lbf/in is required to achieve this. Ultimately, BabyBot came in just under the target weight of 20 lbs when loaded, fitting in a very compact volume of approximately 20” length x 16” width x 20” height.





There is no fixed time limit to complete the mission; however, traversal time is naturally constrained by the onboard power supply, operational efficiency, and the need to execute all navigation and delivery actions within a reasonable time frame. These constraints inform subsystem sizing and navigation strategies. The full course, including Baby Bot's recovery, spans approximately 110 feet. Mother bot is expected to traverse around 50 feet, while baby bot covers about 60 feet. To allow for course inefficiencies, realignment, and potential reentries, the design range is conservatively estimated at 150 feet per robot (for a total of 300 feet total).

Speed requirements are not strictly specified, but the robot must move at a pace compatible with its sensing and control systems. Thus, speed is constrained by practical limits on sensor accuracy, motor torque, and response time, rather than by an explicit performance goal. Controlled movement, rather than velocity, is prioritized to ensure reliable delivery, accurate navigation, and survival of the medkit payload.

B. Operational and Navigational Modes

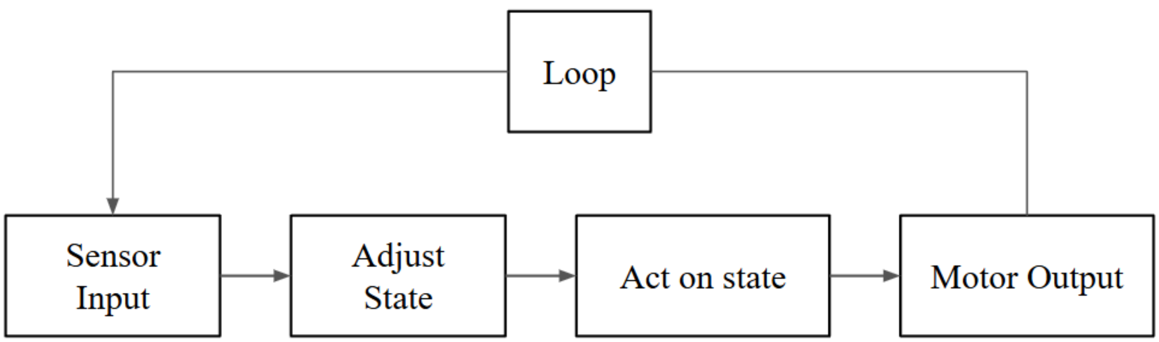


Fig II. B. 1. High-level SARR loop structure based on teensy function

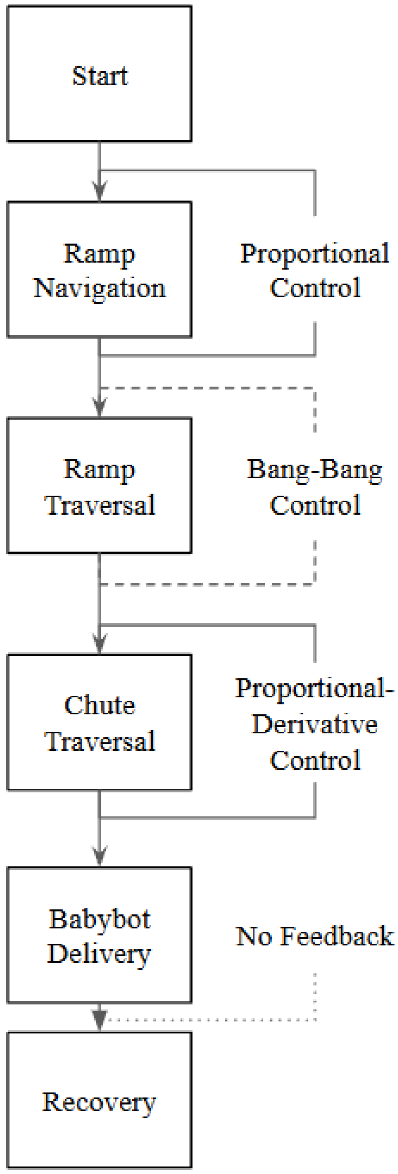


Fig II. B. 2. Finite state diagram of Motherbot from course start to completion

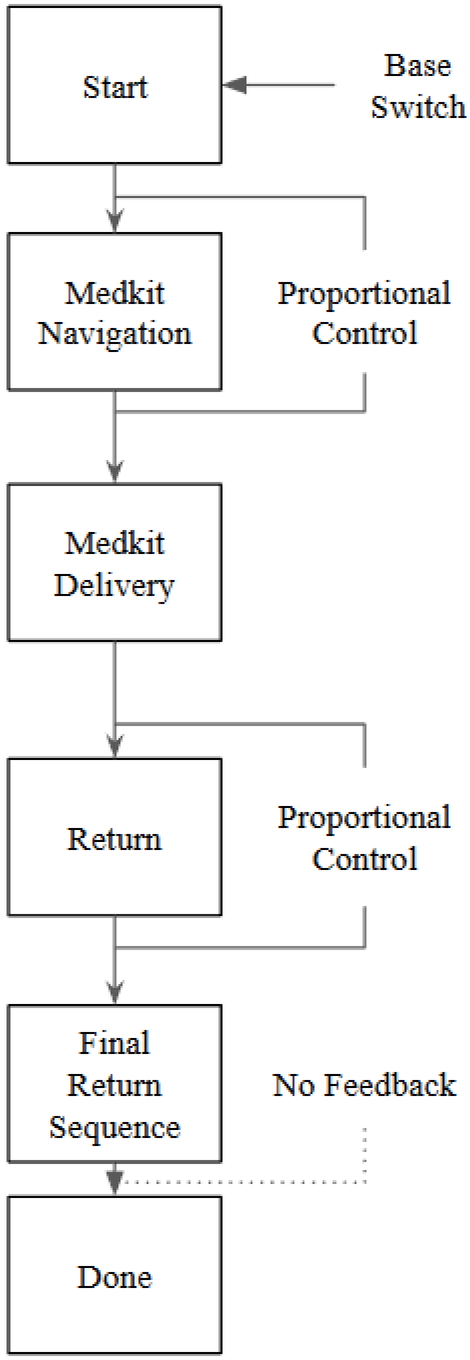


Fig II. B. 3. Finite state diagram of Babybot from course start to completion

The above diagrams illustrate the operational and navigational modes of the Motherbot and Babybot. Because the teensy continually calls the loop() function, the entire functionality of the robots can be encapsulated in a number of “states.” At each call of the loop() function, the robots check which state it is in, and act on instructions specific to each state. Each state also has a specified entry and exit mode.

Both robots have implemented a proportional controller for smooth navigation. This functions by having a degree of turning which is based on how close the SARR is to the wall and how aligned the SARR is towards a navigational light. This could be combined with the current state-based organization, where a single navigation state handles both turning and driving.

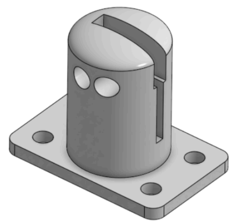


Fig II. B. 4. Babybot limit switch housing

A notable operational mode involves transitioning from control of the larger motherbot to the smaller baby bot. This was accomplished by positioning a limit switch on the bottom of Babybot. Upon making contact with the ground, the limit switch triggers, signaling for Babybot to transition to its autonomous mode.

III. Project Management

A. *Division of Labor*

The group was divided into subteams to better tackle the wide range of work outlined above. Mechanical projects were broken into three subteams: babybot, motherbot, and lifting mechanism. The electrical work for each of these subsystems was primarily led by George Kopf, with programming written by Wilson Moyer. Additional help for electrical and programming was determined on an ad hoc basis, as scope of work was manageable for one other person with mild assistance during bottlenecks throughout the year. For a greater discussion of tasks, timelines, and bottlenecks, see schedule section below. Due to their interest and experience, George Kopf was chosen as 'Electrical Logistics' and Wilson Moyer was chosen as 'Mechanical Logistics'. The role of these individuals was primarily to weigh in on project management discussions from a mechanical and electrical perspective, which provides at least two pairs of eyes and two perspectives on all logistical discussions and decisions (PM + electrical or mechanical logistics lead).

Outside of George Kopf and Ella Simmons, everyone was assigned a mechanical subteam (babybot, motherbot, and lifting mechanism). The scope of work, determined through the primary directives of each subsystem, was established during primary discussions, before groups were chosen. Table IV.A.1 below shows these challenges (and thus, shows the delineation between subsystems). Then, each member of the team selected a subteam to work on. Babybot, due to the larger scope of work, had the largest number of people, 4. The lifting mechanism subgroup had 3 members, and the motherbot subteam had 2 members. See table IV.A.2 below for the specific groups.

Lifting Mechanism	Needs to protect Baby Bot from drop test Needs to lift Baby Bot Watch center of mass shift during lifting Cannot drop Baby Bot early Consider a static mechanism that lowers Baby like a crane Consider using mechanism to alter center of mass
Motherbot	Needs to be maneuverable for chute Needs to handle ramp Needs to survive drop test Watch center of mass shift during lifting
Babybot	Needs to be able to deposit the medkit Needs to be able to navigate to the light source Needs to be able to turn, especially if a cone is put in front of it Needs to be able to return to the main robot Consider a self-righting mechanism or anti-flip design Consider how hard the delivery drop is Consider 3 lb, 5 inch cube of the medkit

Table IV.A.1: Scope of Work for Subteams

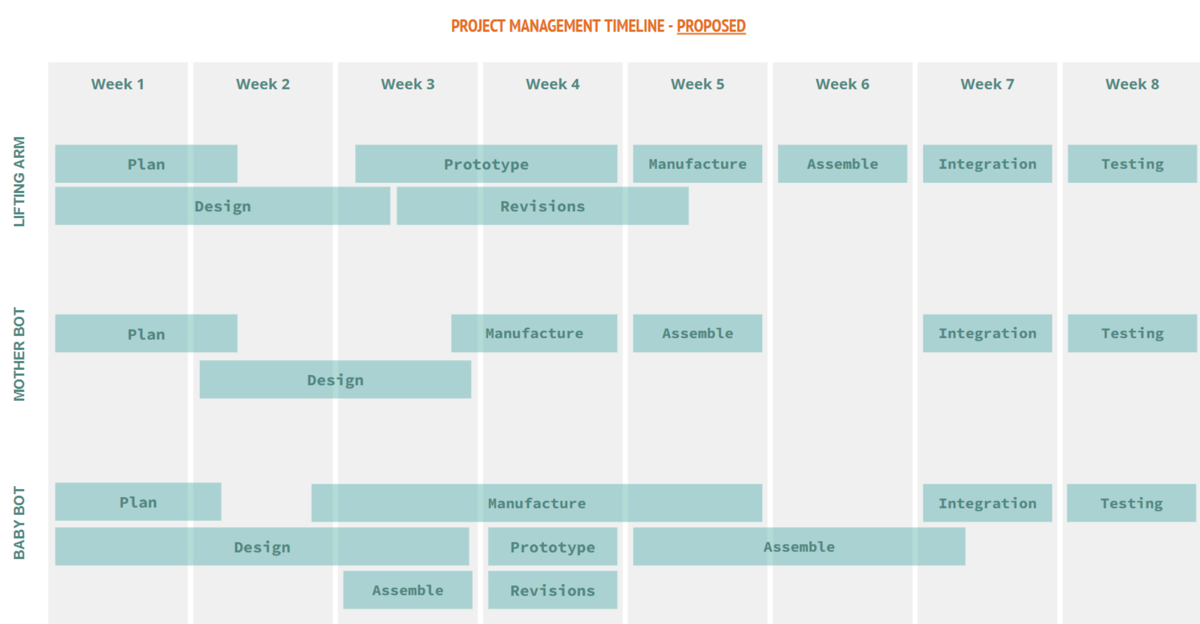
Project Management	Ella Simmons
Electrical	George Kopf
Lifting Mechanism	Joel Boon, Joseph Roberts, Emmanuel Ishola
Motherbot	Stanford Cramer, Chimwemwe Chinkuyu
Babybot	Wilson Moyer, Zachary Andrews, Madelyn Smoyer, Sawooly Li

Table IV.A.2: Members on Each Subteam

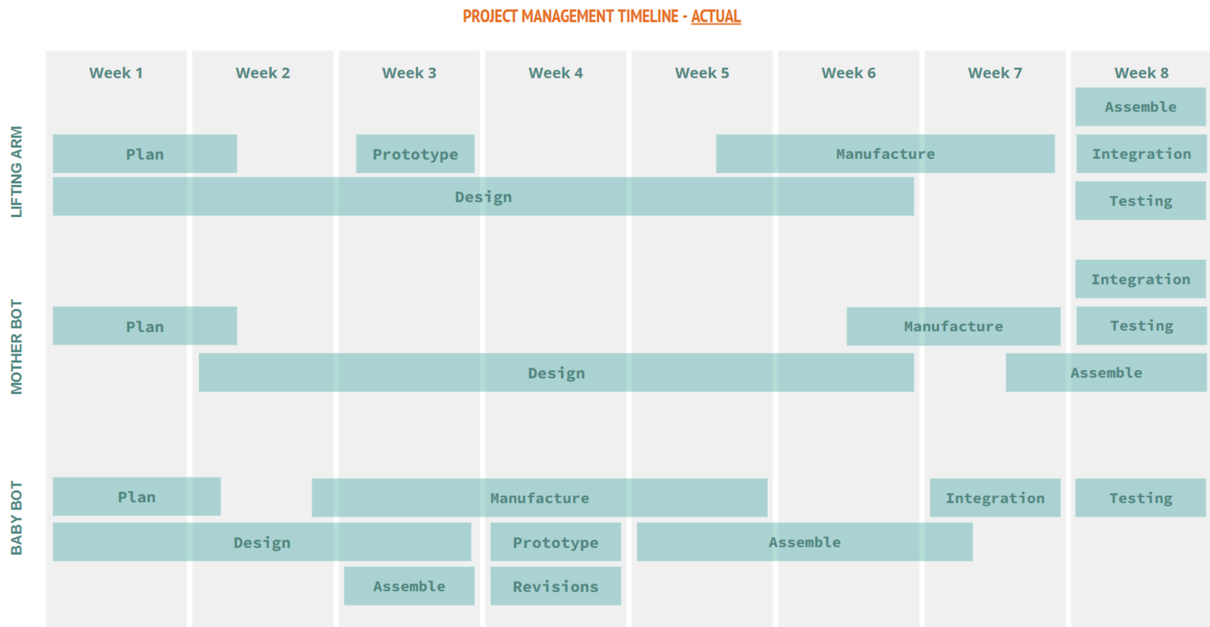
B. Schedule and Key Tasks

After the division of labour was solidified, tasks, milestones, and a timeline were created. See milestone list in Table V.1 (in the appendix). Milestones were created in accordance with a version of the design cycle: planning, prototype, revise, manufacture, integration, test. For each subteam, an evaluation of where the subteam should be during each stage of the cycle was used to make the list. The fact that each subteam has its own milestones makes the list quite long, but it was decided that it was more important to be specific to a team than vague. Milestones were used to inform the task list, which is broken down very similarly. Tasks belong to both a specific subteam and a specific stage in the process, as shown by Table V.2 (in the appendix). Finally, these tasks were placed on a timeline. The timeline was broken down into weeks, which was what we decided was the most specific we can make the timeline while sticking to it. Each subteam was given its own section on the timeline, the rough stages were temporally placed first to determine when each task needed to be completed. Then, an hour estimate was added to each task. If one week had vastly exceeded the hour estimate of other weeks, the tasks were moved around as much as possible. The goal was that each member had roughly 6 hours of work a week, 3 in lab and 3 outside of lab. In particular, the design stage was very compressed in initial layouts, with all CAD complete ~Week 3 according to the schedule. However, there was no way to allow more time to design without cutting other parts of the project off, or not testing, which the team decided not to do. The initial timeline can be reviewed in Table V.3 (in the appendix)

In reality, there were deviations from the timeline. The majority of delays occurred to an extended design time that pushed manufacturing, assembly and testing back farther. The majority of lifting and motherbot manufacturing occurred during the last two weeks. See two figures below which compare the timeline at creation and the actualized timeline. The most time consuming task, as the figure shows, was the design processes. This is due to a number of factors. Firstly, the computer outage halted design for a week in the middle of the main design stage. Additionally, the labor distribution ended up not being completely equal, with babybot having more people than needed and motherbot requiring one or two more people to more evenly spread the workload. Since design was pushed back so far, manufacturing, assembly, integration, and testing had to be compressed into the last 2 weeks, which resulted in an very compressed timeline and less testing than we initially desired.



Project Timeline, Proposed



Project Timeline, Actual

For a greater understanding of the division of labor, see chart below with a general breakdown of who worked on what tasks.

Ella	Lead project management, group communication, and cross-team manufacturing assistance. Electrical and testing assistance.
George	Electrical design and implementation, aided in testing and final debugging.
Wilson	BabyBot medkit delivery, software for BabyBot and MotherBot, led testing and final debugging, general manufacturing
Maddie	BabyBot chassis, MotherBot chassis, aided in testing and final debugging
Sawooly	BabyBot chassis, MotherBot chassis, aided in testing and final debugging
Joe	Lifting mechanism, forging for BabyBot attachment ring, aided in testing and final debugging
Joel	Lifting mechanism, aided in testing and final debugging
Emmanuel	MotherBot chassis, lifting mechanism, and drivetrain
Chimwemwe	MotherBot chassis, drivetrain, lifting mechanism
Zachary	BabyBot hook and holding plate, general manufacturing assistance
Ford	MotherBot chassis and drivetrain, tank drive tensioners, aided in testing and

	final debugging

C. Risk Management

Conflict, albeit tame, did arise throughout the group project. The majority of discourse happened in the first week(s) as different members wanted to go in different directions for the final robot design. To make the process as democratic and efficient as possible, Wilson Moyer created a decision matrix that allowed each member of the team to rank each considered robot design on a variety of criteria: feasibility, uniqueness, reliability, and simplicity. These rankings were compiled, and based on the results, each member voted on the top 3 designs they wanted to move forward with considering. Surprisingly, after the decision matrix, the team was very aligned. Then, in the next meeting (allowing time for consideration), a final design was voted on. The whole process allowed each member to express their opinion and concerns, while still, in the end, unifying around a single decision. This approach early on established a pattern of communication and resilience towards a shared goal, which we carried throughout the rest of the semester. The majority of the semester was very low conflict, with what little occurred happening in the last week or so as stress got higher. The conflict that did arise (often about each other's parts not working) was quickly resolved with communication.

D. Time Estimations

Project management	Ella Simmons				Total
Time (hr)	60 (includes helping individual teams)				60
Electrical	George Kopf	Ella Simmons			Total
Time (hr)	70	10			80
Lifting Mechanism	Joel Boon	Joseph Roberts	Emmanuel Ishola		Total
Time (hr)	55	66			121
MotherBot	Stanford Cramer	Chimwemwe Chinkuyu			Total
Time (hr)	65	56			121

BabyBot	Wilson Moyer	Zachary Andrews	Madelyn Smoyer	Sawooly Li	Total
Time (hr)	80	65	70	55	270
Software	Wilson Moyer				Total
Time (hr)	34				34

E. *Overall Approach*

Communication and organization were the two tenets of this team's project management approach. A Discord server was used for team communication. The server had multiple different channels (ex., 'design-reports', 'babybot', 'electrical-general') to organize communication. Additionally, several early meetings established the primary goals of the team and the robot. These meetings were scheduled during a time when everyone indicated they were free, and an agenda was determined at the beginning of the meeting to streamline the discussion. At every meeting, a member of the team takes notes, and anytime a mechanism is being discussed, a member will create a drawing and put it in these notes. In essence, we were very proactive about communicating effectively and being organized in our meetings thus far.

Throughout the semester, at the start of each week during lecture, Ella went through every task from the previous week and marked it as complete or understood why it was not complete yet. Then, we discussed the tasks for the upcoming week, any risks or concerns and help needed. This allowed everyone to be on the same page and Ella to follow up on risks and reassign members who had smaller tasks for the week. This worked well for the first several weeks; however before Thanksgiving, work ramped up in other classes so people were tired/less responsive. In the last two weeks to make sure everything got done, Ella met with each group and planned out exactly what needed to be done in the last weeks and what the projected timeline was. This ensured that things got completed.

IV. Detailed Design and Analysis

A. *Baby Bot*

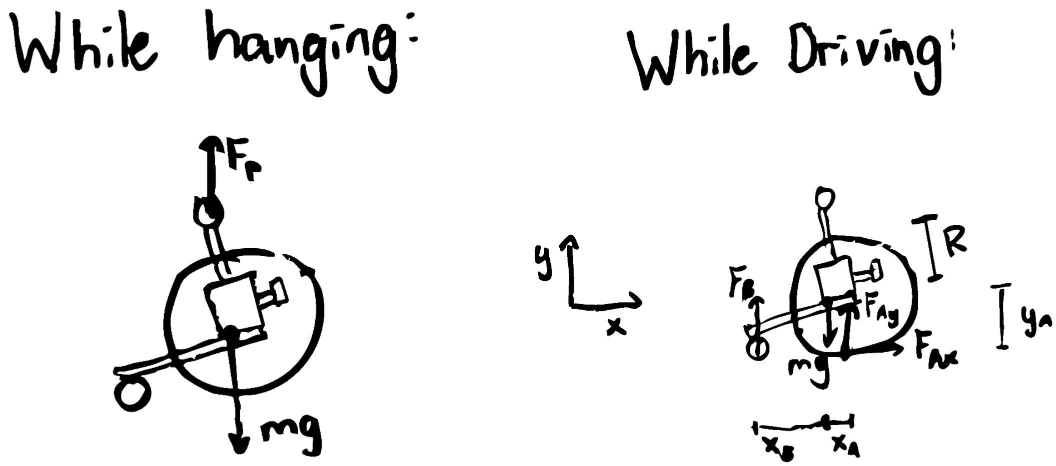


Fig III. A. 1. Babybot FBD in hanging and driving mode

The above figure demonstrates a FBD of what the Baby Bot looks like in its hanging and driving modes. What follows is a detailed analysis of each mode:

<u>Hanging mode:</u> In this situation, the total force on the baby bot should be

$$\Sigma F_y = F_p - mg = ma$$

Where m is the mass of the baby bot, and for a static scenario, $a = 0$, giving $F_p = mg$. However, the primary concern is when the baby bot undergoes the drop test, in which case $a \neq 0$. The upwards force will be provided by the hanging hook. To model this scenario, we will take the moment that the baby hits the ground as an impulse $J = m\Delta v$. To simplify the problem, we can assume that the impulse is instantaneous, giving

$$F_p(t) = m\Delta v\delta(t) + mg$$

Where $\delta(t)$ is the dirac delta function, and Δv can be easily solved using the timeless kinematic equation with the assumption that the initial velocity before drop is $v_0 = 0$, the velocity immediately before the mother bot hits the ground is v_{f-}, and the velocity immediately hitting the ground after is $v_{f+} = 0$. This assumes constant acceleration, which is a fair assumption due to the small drop height of 1 ft. With these equations, we get $v_{f-} =- \sqrt{v_0^2 + 2g\Delta y}$ which simplifies to $v_{f-} =- \sqrt{2g\Delta y}$. This means that $\Delta v = v_{f+} - v_{f-} = \sqrt{2g\Delta y}$, which we can plug back into our original equation, resulting in

$$F_p(t) = m\delta(t)\sqrt{2g\Delta y} + mg$$

In theory, the magnitude of $\delta(0)$ is infinite, which would result in immediate fracture of the babybot. Thankfully, in practice, an impulse never has an infinite force magnitude, as impulses always occur over a discrete period of time. In addition, to prevent Babybot from being damaged while dropped, we constructed a cradle out of L-channel on top of Motherbot which supports Babybot, so it is not always hanging by the hook. Rather, the hook supports Babybot partially, and the cradle supports the rest, so the cradle mode is similar to a combination of the hanging and driving modes.

A small note to keep in mind for the babybot in hanging mode is that if F_p is not aligned with the babybot's center of mass, there will be a net moment, and the babybot will undergo angular acceleration. This is beneficial during wall traversal because it will keep the babybot oriented correctly with the ground, but it is an important consideration for the drop test to ensure that the babybot is situated properly.

<u>Driving mode:</u> This FBD produces forces in two directional components:

$$\Sigma F_y = F_B + F_{Ay} - mg = ma_y$$

$$\Sigma F_x = F_{Ax} = ma_x$$

Where $F_{Ax} = R\tau_M$, where τ_M is the torque applied from the motor on the wheel (after the gear train). In this situation, $a_y = 0$ since the babybot will not levitate or sink into the ground. In general, $a_x \neq 0$, since the baby bot can be speeding up or slowing down. We also recover from the FBD a moment balance equation:

$$\Sigma M = -F_B x_B + F_{Ay} x_A + F_{Ax} y_A = I\alpha$$

Where all of the x and y are positive quantities and we want $\alpha=0$ to prevent the baby bot from tipping over. Based on looking at the FBD, we can observe that the more likely mode of failure is the baby bot tipping forwards since $x_A < x_B$. To ensure that this does not occur, we can shift the wheels forward or shift the weight backwards to increase x_A.

<u>Babybot Motor Calculations</u>

As Babybot does not encounter any terrain obstacles on its path to deliver the medkit, we assumed that it would not be travelling up or down an incline. On flat ground, the total required driving force for Babybot to travel in a straight line is $F = fmg$, where f is the coefficient of friction and m is the total mass. Consequently, the total torque required to drive the wheels will be $T_w = rF = rfmg$, where r is the radius of Baby bot's wheels. Considering the interaction between the wooden and our rubber wheels, we will choose $f = 0.7$ and estimate Babybot to weigh around 30lb. Consequently, we can estimate that the driving torque required will be around $T_w = rfmg = 84\text{in lbf}$.

In order to translate this into the required torque for the motor, we can use $T_m = \frac{T_w}{\eta G} = \frac{rfmg}{\eta G}$, where η is the transmission efficiency and G is the gear ratio. For transmission efficiency, the slides estimate less than 0.8, but to be safe and account for unforeseen inefficiencies, we can use $\eta = 0.7$. For a small robot, we can select a gear ratio of

20:1, which will produce enough torque to meet our margin of 84in lbf while keeping a reasonable driving speed. Thus we can take $T_m = 6$ in lbf.

To find the motor speed, we can use $\omega_m = \frac{Gv}{r}$. Choosing a cruising velocity of around 0.5m/s, we find that $\omega_m = 940$rpm, which is a very comfortable operating speed for most small DC motors. Thus, the total power needed by the motors is $P = T_m \omega_m = 66.7$W, or around 33W for each motor.

While turning, the torque required may increase, assuming a locked or solid axle model. The extra resisting force can be factored in to identify how much more torque would be needed to turn. Assuming a turn radius of around 0.5m and a Baby bot width of around 0.3m to correspond to its 8 inch wheels, we can approximate the slip to ratio to be $s = \frac{b}{2R} = \frac{0.3}{2 \times 0.5}$ and repeat our force and torque calculations with the additional resistance: $F_{turn} = mgf(1 + s)$, and $T_{turn} = rmgf(1 + s)$. By performing the calculations for straight line drive with the additional force, we find that $T_{turn} = 7.8$ in lbf, corresponding to approximately a 30% increase. However, these values still stay well within the constraints of most DC motors.

The actual motor we chose was the E-S 12V, 60 RPM DC motor, paired with an additional 2:1 gear reduction. With this reduction, the effective wheel speed is approximately 30 RPM. For an 8-inch diameter wheel (4-inch radius), this corresponds to a linear ground speed of about 0.3 m/s, which is on the same order as the cruising velocity assumed in the motor calculations above (0.5 m/s) and is appropriate for controlled, stable motion during medkit delivery. From a torque perspective, the added 2:1 reduction doubles the available output torque at the wheels, providing sufficient margin relative to the per motor torque requirement calculated earlier. With two motors sharing the load, the selected motors comfortably meet the required wheel torque even when accounting for drivetrain inefficiencies and additional resistance during turns.

B. Mother Bot

The primary goal of the motherbot is to traverse the ramp and chute and deliver the babybot. Therefore, it is crucial to analyze wheel traction on the ramp and determine the required coefficient of friction μ.

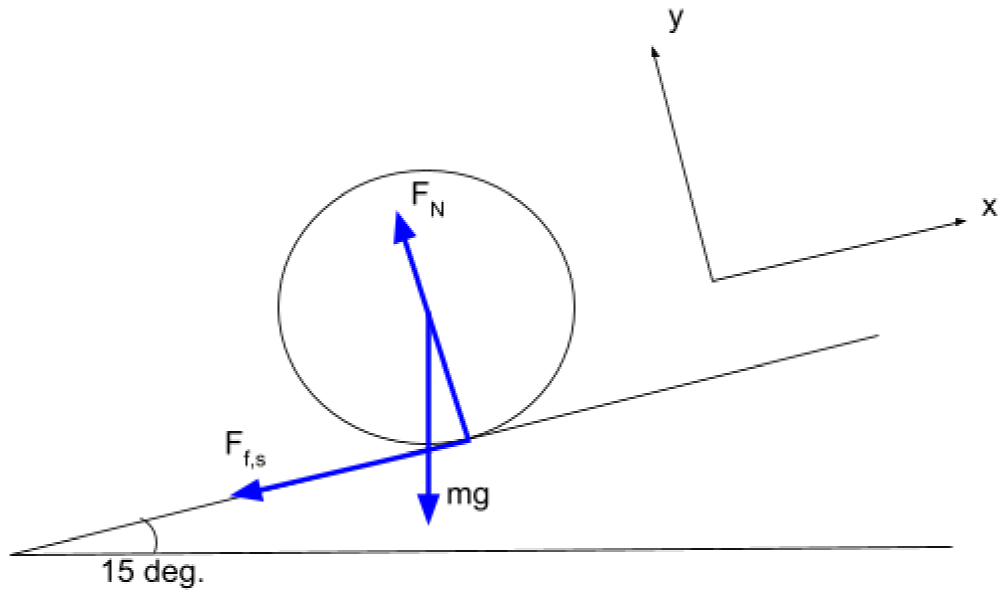


Figure III.B.1 FBD of Motherbot Wheel

In Figure III. B.1, we are analyzing the forces on one wheel of the motherbot while on the ramp. We can do this on one wheel and generalize to the whole robot. Let's now look at the forces present:

$$\Sigma F_y = F_N - mgcos(15) = 0$$

$$\Sigma F_x = mgsin(15) - F_{f,s} = 0$$

Rearranging the equations and using $F_{f,s} \leq \mu_s F_N$ yields:

$$\mu_s \geq tan(15) = 0.268$$

This seems like a reasonable requirement for the static frictional coefficient. The ramp is made out of wood, and our wheels are hard rubber, which corresponds to a coefficient of friction of about 0.7. Traction was not an issue.

The motherbot and babybot weighed 90 lbs and 20 lbs, respectively. With a safety factor of 2, the total robot weight will be about 220 lbs. We used 4 motors: 2 CIM motors and 2 mini CIM motors, to drive the motherbot. Using this, the total required torque (T) to get the robot over the ramp and the torque required from each motor (T_{motor}) were calculated as follows.

$$\Sigma F_x = mgsin(15) - F_{f,s} = 0$$

$$F_{f,s} = 220\, lbf \; * \; sin(15)$$

$$F_{f,s} = 56.94\, lbf$$

$$T = F_{f,s} * \frac{5}{12}\text{ft} = 56.94\ lbf * \frac{5}{12}\text{ft}$$
$$T = 23.725 \text{ lb-ft}$$

$$T_{motor} = T * \frac{1}{4\ motors} = 23.725 \text{ lb-ft} * \frac{1}{4\ motors}$$
$$T_{motor} = 5.931 \text{ lb-ft per motor}$$

The motor curves for the CIM motor list their maximum torque as $T_{max} = 1210.1972 mNm$ and the transmission efficiency as $\eta = 0.66$. To use these motors, the required gear reduction ratio is

$$G = \frac{T_{motor}}{\eta * T_{max}} = \frac{5.931\ lbf}{0.66 * 1210.197\ mNm} * \frac{1355.82\ mNm}{1\ lbf}$$
$$G = 10.08 \approx 10 \rightarrow \text{We needed a gear ratio of 10:1 for each motor.}$$

Likewise, the motor curves for the mini CIM motor list their maximum torque as $T_{max} = 710.39211 mNm$ and the transmission efficiency as $\eta = 0.58$. To use these motors, the required gear reduction ratio is

$$G = \frac{T_{motor}}{\eta * T_{max}} = \frac{5.931\ lbf}{0.58 * 710.392\ mNm} * \frac{1355.82\ mNm}{1\ lbf}$$
$$G = 19.51 \approx 20 \rightarrow \text{We needed a gear ratio of 20:1 for each mini CIM motor.}$$

The both the CIM and mini CIM motors are attached to a 20:1 planetary gearbox, and we are using belts and pulleys that give an additional factor of 3:1 for a total achieved gear ratio of 60:1. Considering this is 6 and 3 times more than required, we were able to run our motors at a much lower amperage than what is needed for max torque output.

With the reduction ratio of 60, the motor output speed will be set to that of the motor with the slower RPM_{max}. Since $RPM_{max,\ CIM} = 2655$ rpm $< RPM_{max,\ mini\ CIM} = 2899$ rpm, then

$$Motor\ Output = \frac{RPM_{max,\ CIM}}{G} = \frac{2655\ rpm}{60} * \frac{1\ min}{60\ s}$$
$$Motor\ Output = 0.738\ rps$$

The wheel speed is $Wheel\ Speed = \frac{Motor\ Output}{2 * \pi * r} = \frac{0.738\ rps}{2 * \pi * 5} = 0.0235\ rps$ on the ramp. While this is slow, we wanted to drive slowly on the ramp to prevent falling off and to allow careful navigation with our light sensors.

In addition to being able to drive, the motherbot needed to withstand the impulse from a 1-ft high drop test without sustaining any damage to either robot. Since the motherbot's weight must also counteract the moment at the end of the lifting mechanism when the babybot is hanging, we decided to manufacture the motherbot's frame from low-carbon steel C-channel. Steel's high density makes it a better material than aluminum, wood, or plastic to meet both the weight and vibration resistance needs. Furthermore, the wheels we selected for the motherbot have a maximum load of 150 lb each and a 1-in-thick rubber tread surrounding a PVC hub. Our own testing, along with the final drop test, revealed that these measures absorbed vibrations well enough to not dislodge or break any motherbot components.

Another concern was bottoming out when crossing the top of the ramp.

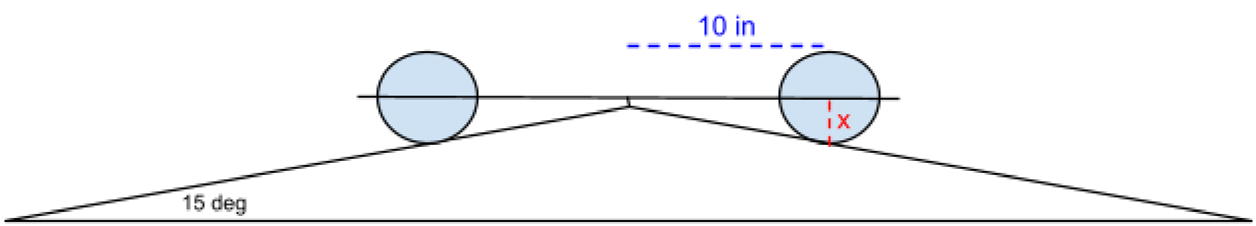


Figure III.B.2 Motherbot Clearance Sketch

We had 20 inches between the wheels, so 10 inches from each wheel to the center (blue dotted line). We can use the law of sines to determine the required clearance from the wheel axle to the ground (red dotted line).

$$\frac{sin15}{x} = \frac{sin75}{10}$$

x = 2.68 inches of clearance needed

We used wheels with a 5-inch radius. The wheels are mounted on an axle 1.5 inches above the bottom of the Motherbot's C-channel frame. So, the 5-in wheel - 1.5 in of robot frame = 3.5 inches of clearance above the floor. This is more than ¾ of an inch greater than we need.

A further concern is the maneuverability of the motherbot in the chute.

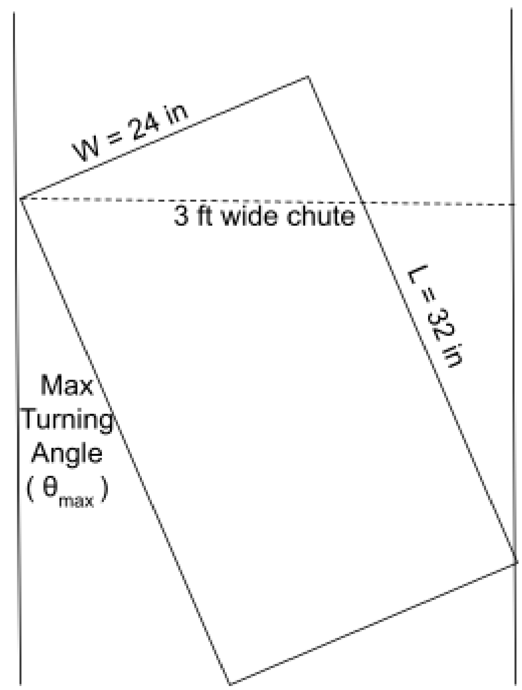


Figure III.B.3 Max Chute Turning Angle

We can solve for the maximum turning angle by using the length (L) and width (W) of the motherbot.

$$W cos\theta_{max} + L sin\theta_{max} \leq chute\ width$$

$$(24 in)cos\theta_{max} + (32\ in)sin\theta_{max} \leq 36 in$$

$$\theta_{max} \leq 27.3\ degrees$$

Considering the maximum angle present in the chute will be 30 degrees, one would expect it will be hard/impossible for our robot to navigate through it. However, on test day, by being careful in the chute and using precise movements, motherbot autonomously navigated through without hitting the wall. This may have been due to the width of the chute being slightly greater than 3 feet.

C. Lifting Arm

Since the main advantage of the motherbot-baby bot design is that the baby bot can be placed over the wall without having to traverse it on its own power, the lifting mechanism is a key part of the overall design. The goal of the lifting mechanism is to reliably and gently place the baby bot on the other side of the wall, and if possible retrieve it as well. Tangentially, a key aspect of the lifting mechanism is controlling the COM of the motherbot, particularly during the ramp traversal, until it is time to lift the baby bot over the wall. In order to achieve this task, several designs were discussed.

The first design considered involved a large rotating arm which would lock the baby bot on top of the motherbot during initial transport, and then would rotate 180 degrees to drop the baby bot over the wall. While this design was simple in theory, and had a robust way of grabbing onto the baby bot, it was discarded due to the large amount of torque required to rotate the arm from its resting position to its final position.

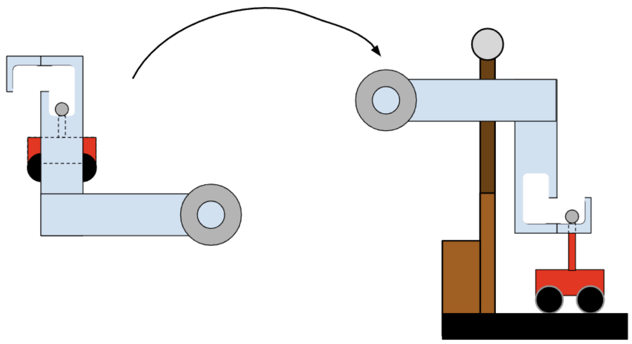


Figure III.C.1 Rotating Arm Lifting Mechanism

The next design considered involved a sort of crane mechanism which would lower slightly near the wall and lower the baby bot down on the other side using a winch. This design had a significantly lower torque requirement out of the motor used to lower the crane arm and baby bot, which was a major upside. However, it was discarded due to its unnecessary complexity.

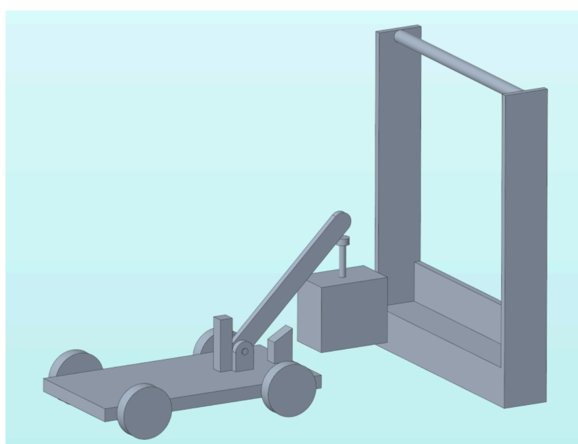


Figure III.C.2 Crane Arm Lifting Mechanism

The last design considered, which is the one that was used, is similar to the crane arm design, but is more static. It involves a long static arm which extends out beyond the front of the motherbot by roughly 22” so that the motherbot can drive up to the wall, and simply lower down the baby bot on the other side of the wall. This arm is secured by a large static structure that is placed on top of the motherbot chassis in order to assure stability and add weight back of the arm in order to move the COM farther back. A key feature of this design is rather than simply having the baby bot stay at the very end of the arm during the ramp and chute traversal stages of the course, the baby bot stays closer to the center of the motherbot and then is slid out to the end of the arm once the motherbot reaches the wall. This allows the COM of the system to stay farther back during the stages of the course when the robot is more likely to tip over. Once the baby bot is at the end of the arm, it is lowered down to the ground on the other side of the wall. Below are sections pertaining to the calculations for the torque required for the winch system and the sliding mechanism, as well as calculations about the tipping point of the system.

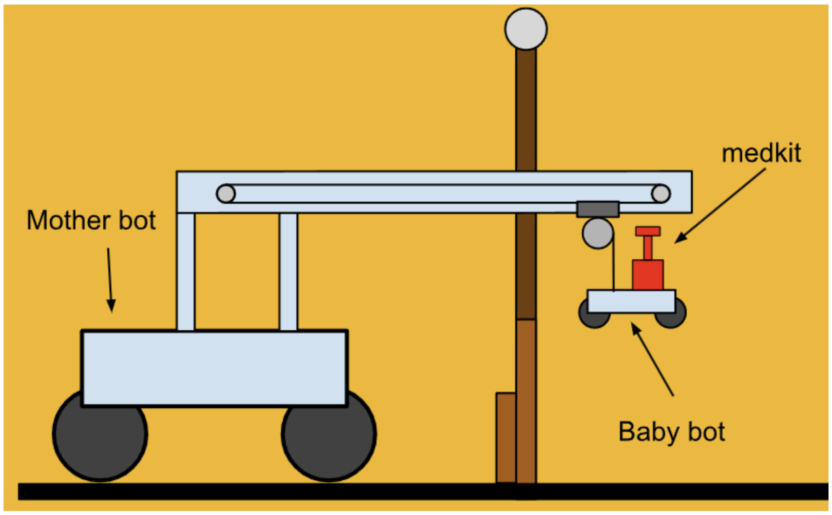


Figure III.C.3 Final Lifting Mechanism

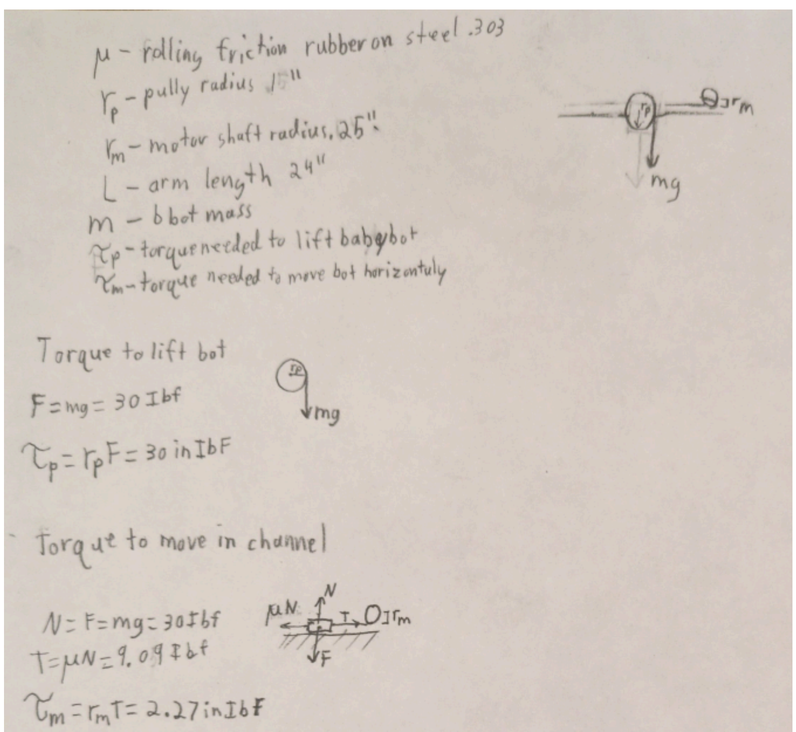


Figure III.C.4 Lifting Mechanism Winch Torque Analysis

This image above shows the calculations made to determine what motors will be needed for the lifting mechanism. To move the bot vertically a winch mechanism was used. Assuming that the baby bot would be around 30 pounds, a 2 inch diameter spool on the motor and assuming that the weight of the cable would be negligible. It would require a motor capable of at least 30 inch pounds of torque to lift the baby bot. To move the bot horizontally we needed a motor capable of at least 2.27 inch pounds of torque. This calculation was done assuming a motor with a ½ inch diameter shaft, rolling without slipping, that the weight of the first motor is negligible, and that the carriage would roll on bearings.

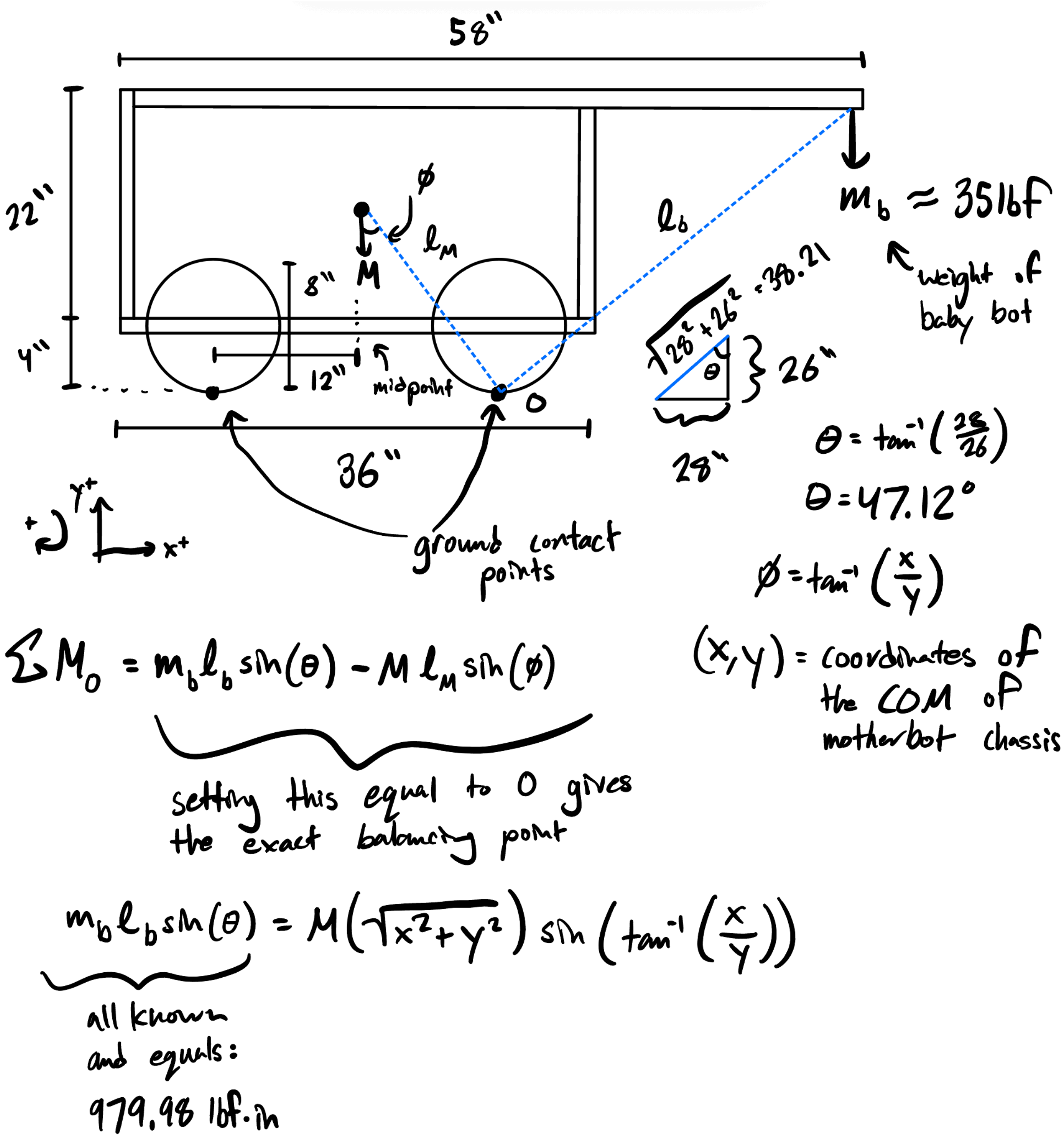


Figure III.C.5 Lifting Mechanism Moment Analysis

In addition to the winch calculations, an essential part of designing the lifting mechanism was to ensure that at no point would the baby bot cause the motherbot to tip over. Therefore, to get a rough idea of how to combat that, a moment analysis was performed on the robot. This analysis is shown in Figure III.C.5. The analysis is set up such that the final equation in the analysis gives a function based on the weight of the motherbot (not including the weight of the baby bot) and the coordinates of the motherbot's COM. Solutions to this equation are when the system is perfectly balanced over the front wheel of the motherbot. Therefore, it was desired that the system be made more conservative so that a safety factor could be built into the robot.

In order to solve this equation, one or two of the variables could be arbitrarily chosen, and the other variable(s) can be solved for. This was done using Desmos and by choosing the x-coordinate of the COM and solving for the y coordinate of the COM and the weight of the motherbot. During this Desmos analysis, it was found that the y-coordinate of the COM has no impact on the weight needed to balance the system for a given x value. Therefore, only the x-coordinate of the COM and the weight of the motherbot will be considered.

For an x value of -12" (which would place the COM in the middle of the wheels), the weight of the motherbot that balances the system is about 82 lbf. By inspection, any weight above this will allow the system to be more stable. A COM x-coordinate in between the wheels is very reasonable since the vast majority of the chassis is around the midpoint of the wheels. A weight of more than 82 lbf is less reasonable. If, however, the COM x-coordinate is shifted farther back from the front of the robot, a lower weight is needed to balance the system. An x value of -24" (which would place the COM over the back wheel) would require a motherbot weight of just 41 lbf to balance the system. In this case, the weight requirement is quite reasonable as this weight is just slightly more than the estimated weight of the baby bot, and the motherbot is substantially larger than that. However, it would be a lot harder to place the COM over the back wheel. A compromise between the two extremes would certainly be possible, however, since the motherbot can be intentionally back-weighted, which would move the COM farther back, and the final weight of the motherbot will likely be close to 80 lbf anyway due to the metal comprising the chassis and lifting mechanism.

Additionally, this is the reason that the sliding deployment mechanism for the lifting mechanism was used since it allowed the adverse moment about the front wheel to be lower until the baby bot needs to be deployed over the wall, at which point the motherbot didn't have to traverse the ramp or encounter other perturbations, so the stability was less of a concern. We anticipated that the time required to deploy the arm and for the Baby Bot to safely detach on the other side of the wall would be no more than 1 minute.

Figure III.C.6 Final Lifting Mechanism Design

The final design used and made in CAD is shown above in Figure III.C.6. The four standoffs were bolted on top of the chassis to form a rigid structure. Essentially, the final design involved a rigid frame, an arm with a slot, a carriage, and a winch mechanism. There were separate motors for the winch mechanism and the carriage movement so that controlling them independently would be easier. The carriage was the part that slid back and forth inside of the arm and was connected to the arm motor via a 3D printer timing belt. The winch mechanism was screwed into the carriage so that the winch position could be controlled by the carriage position. This system allowed for complete planar control of the location of the baby bot. Additionally, the materials used for the lifting mechanism were generally overkill, but it allowed the lifting mechanism to be very rigid and stable, which helped ensure consistency with the deployment of the baby bot.

D. Medkit Placement System

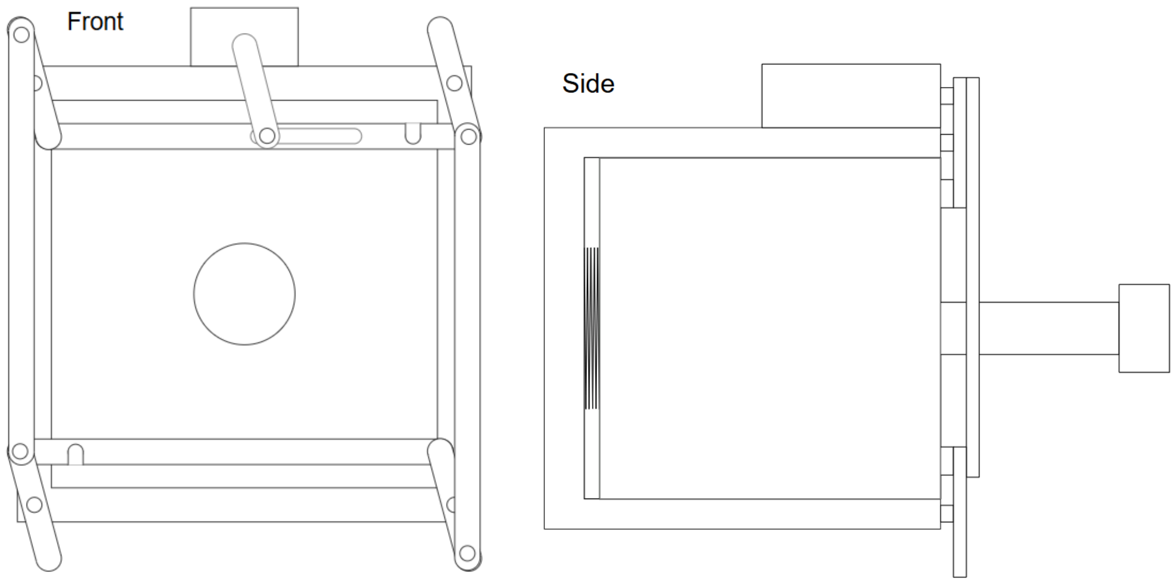


Fig III. D. 1. Med kit in secured configuration

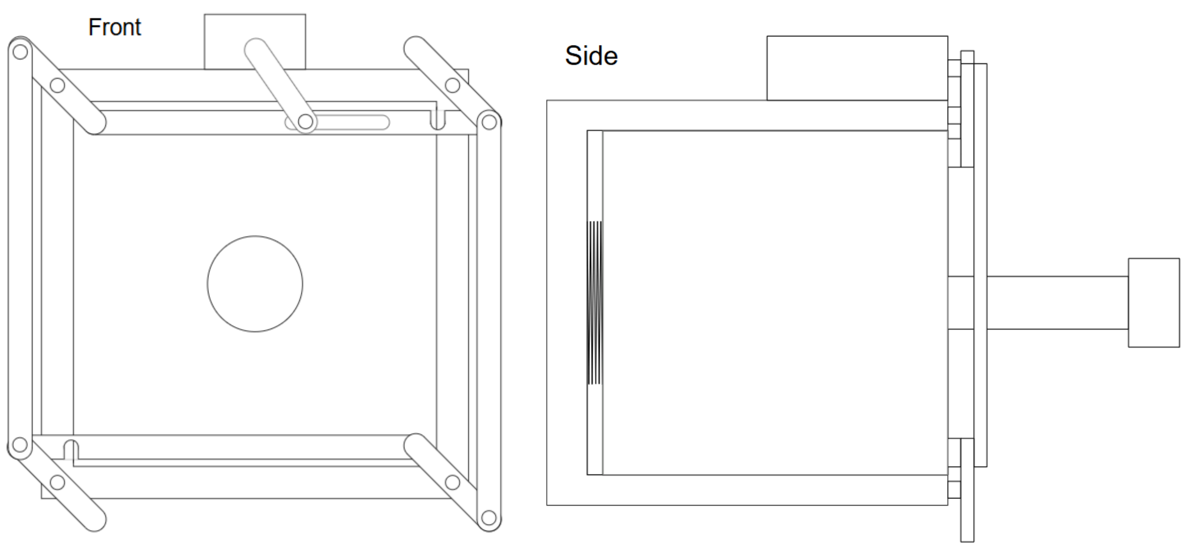


Fig III. D. 2. Med kit mid-deployment

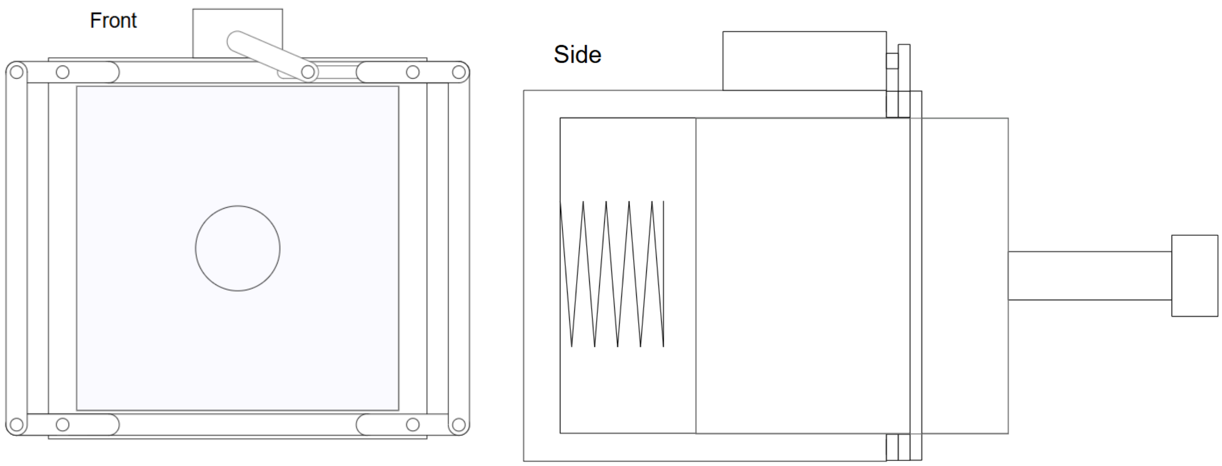


Fig III. D. 3. Med kit fully deployed (blue-shifted to represent velocity)

Figures III. D. 1-3. Demonstrate a preliminary deployment mechanism for the medical kit that can be used with the Baby Bot. The mechanism involves four coupled four-bar linkages. Two of the couplers of the linkages constrain the top and the bottom of the med kit. When the servo on the top is actuated, all of the linkages move in unison, having one DOF. Once the linkages stop constraining the medical kit, a spring in the back propels it outwards towards the medical kit reception box. This design has the advantages of only requiring one servo to actuate and holding the medical kit securely in place before releasing it. In addition, it takes up a very small footprint (not much larger than the medical kit itself), which is crucial for the baby bot, which should be as small and lightweight as possible. However, the main issue with this mechanism is that when the spring presses on the four bar linkages, the linkages have significantly more friction, leading to binding and large torques required to actuate the system. As a result, we decided to pivot to a different mechanism shown below.

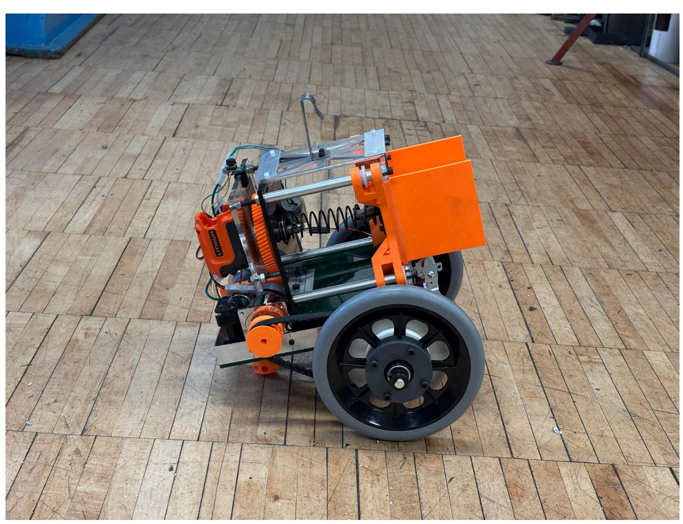
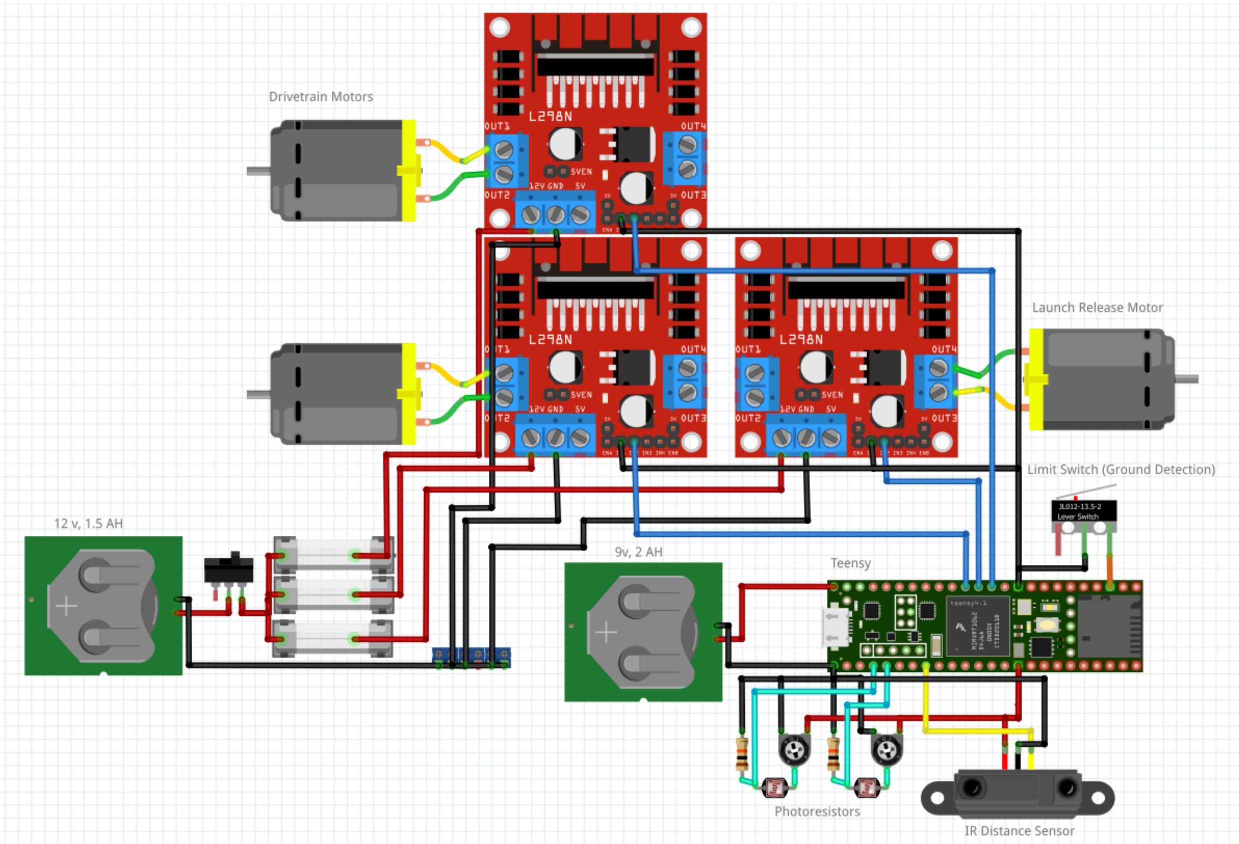


Fig III. D. 4. Final medkit deployment system

Fig III. D. 4 shows a final design for the medkit deployment system in a Creo assembly. The design involves having the medkit mounted in a carriage pressed against a spring, just like the original design. The main difference is that the constraint is in the back, with a rotational key mechanism that a motor drives, releasing the carriage and allowing the medkit to be propelled forward into the carriage.

The design for BabyBot involved a spring-loaded medkit delivery system oriented at a 15 degree angle from the horizontal. The calculations were performed under the assumption that Baby Bot leaves no extra space between the edge of its front wheels and the medkit receptacle and resulted in a desired spring constant of 3 lbf/in. When it came time to select an actual spring for Baby Bot, there were very few options available that had an appropriate spring constant and the necessary geometry for the medkit deployment system. The medkit deployment system was initially designed for a spring of at least a half inch inner diameter, resting length of 5 inches, and 4 inches of compression. Thus, the decision was made to choose a spring that best balanced the geometry requirements without requiring too much extra force. The final spring selected had a spring constant of 8.5 lbf/in, inner diameter of 1.033 inches, resting length of 5 in, and maximum compression of 3.85 in, as this was the spring that would require the fewest changes to the medkit deployment design. The extra force requirement was within the limits of what the deployment system could bear, and the team agreed that having more potential energy in the spring would be valuable, so that if there were any redesigns causing extra mass or friction in the deployment system, a new spring would not have to be ordered. Additionally, any extra energy in the deployment could be countered by simply programming BabyBot to stop farther away from the medkit receptacle. Testing validated the effectiveness of these measures, as the spring was sufficient to propel the medkit into the receptacle. The spring was more strong than needed, but this was adjusted in code by tuning the distance that Babybot stopped from the receptacle. In the end, this method of actuation ended up being very successful and reliable.

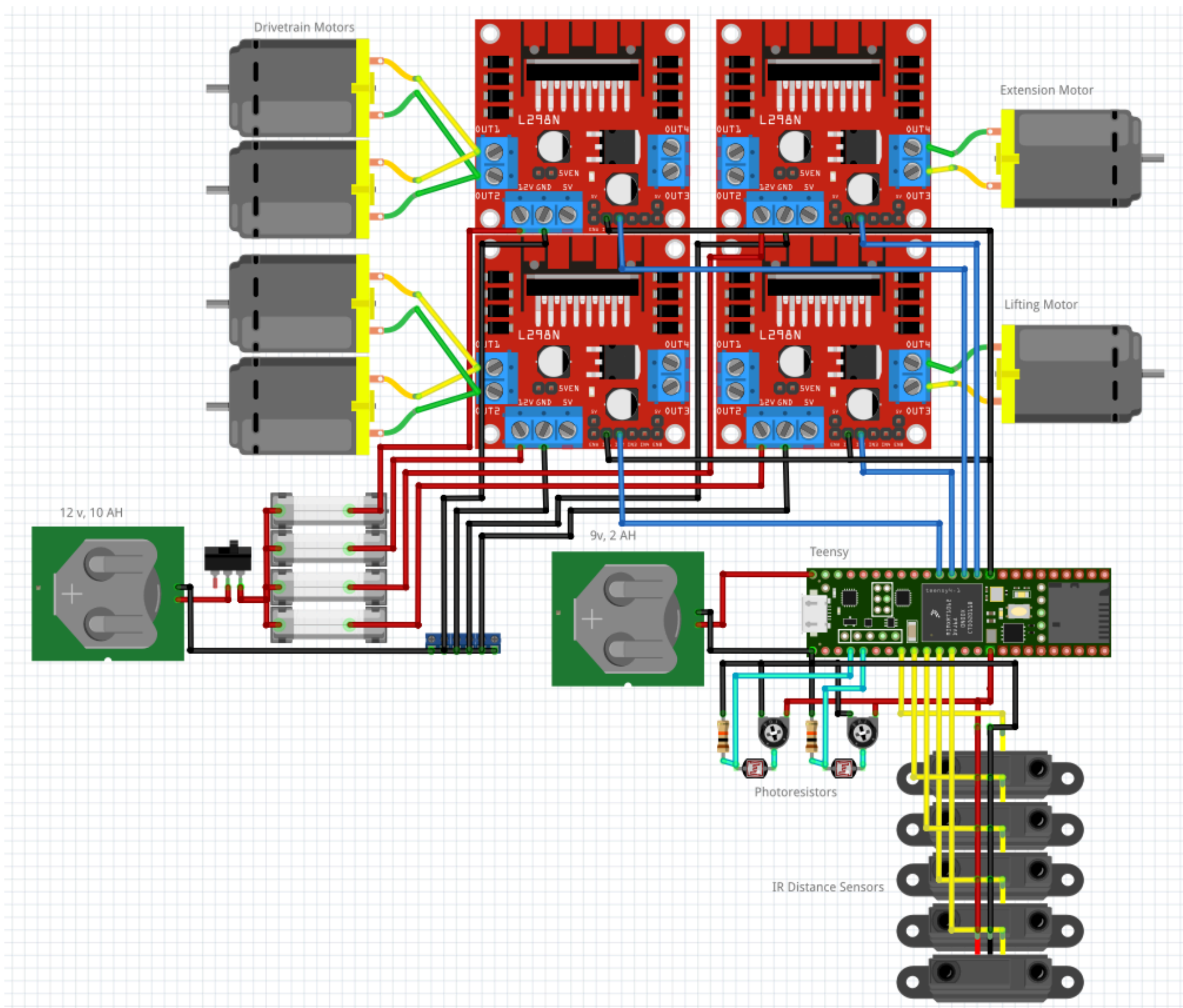
E. Electrical Schematic



Electrical Schematic of BabyBot

BabyBot's electrical design follows a very similar layout to the sample robot. Includes two motors for drivetrain, two photoresistors, a forwards-facing distance sensor for medkit delivery, and an additional motor for triggering the medkit firing mechanism. We also added in a limit switch placed on the bottom of the BabyBot so that it can automatically detect when it is placed on the ground by MotherBot. A small 1.5 Ah drill battery powers the drivetrain motors (which are refitted DC drill motors) and the small motor that releases the medkit launcher spring.

MotherBot's electrical design again follows a similar layout, but with more of everything. Particularly, MotherBot was outfitted with an array of 5 distance sensors; one forwards, two out, and two down, which allowed it to have a strong positional sense of where it was in the course. This allowed us to navigate efficiently without the need for an IMU or other more complex sensor systems. The drivetrain runs four CIM motors, one drill motor for the lifting mechanism, and one smaller DC motor for the extension mechanism. One interesting design decision was to link the drivetrain motors in pairs, so four motors run through two motor controllers. In testing, this was determined to be a viable option, so the electrical system could be simplified without dipping below our required torque levels. However, at the actual demo we found that the motor controllers would shut off after several minutes of continuous use. Likely, what was happening was that we were pulling peak currents of around 60 - 80 amps, not long enough to pop the fuses but definitely enough to overheat the MOSFETs inside the motor controllers. They cycled into a protection stall and refused to reactivate until they cooled down. It was very unfortunate that we didn't catch this during testing, because we did not run multiple tests consecutively (i.e. there was a consistent gap for the motor controllers to cool down after each run as we debugged mechanics and code), we did not realize our motor controllers (and not our fuses as we assumed) were likely the limiting factor on MotherBot's ability to run continuously. Adding in the additional pair of motor controllers to the drivetrain likely would have greatly helped reduce the impact of this issue.



Electrical Schematic of MotherBot

F. *Autonomous Operation*

For autonomous operation, Babybot and Motherbot used states as shown in Fig II. B. 2. and Fig II. B. 3. These states alongside both robots' sensor suites permitted for autonomous operation on the obstacle course. To begin, Motherbot used the photoresistor light sensors to navigate to the ramp, implementing a proportional feedback controller. The forward-facing distance sensor informed Motherbot when it arrived at the ramp. At this point, Motherbot implemented “bang bang” control using downwards facing distance sensors mounted on the sides in order to prevent Motherbot from falling off. Upon sensing the chute with the side sensors, Motherbot would enter the chute traversal mode, in which it implemented proportional-derivative feedback control using the two sideways facing distance sensors to stay centered in the chute. After making it through the chute, the forward-facing distance sensor informed Motherbot when to stop and deploy Babybot. The deployment of Babybot was done without feedback, with set delays for each motor actuation.

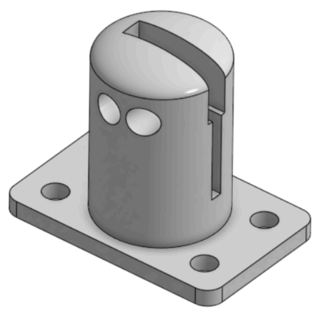
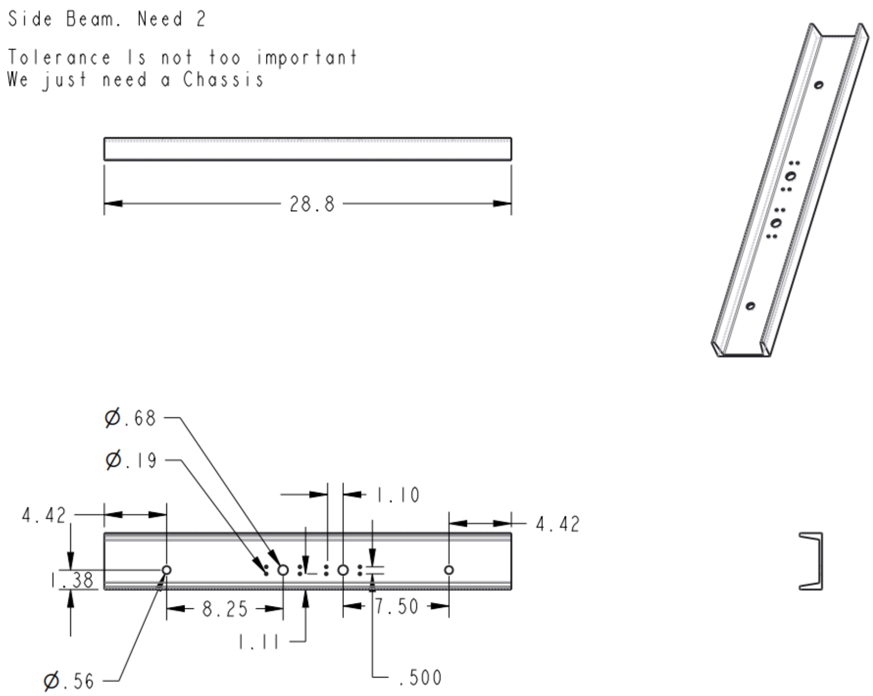


Fig. III. F. 1. Babybot peg with slot integrated for limit switch

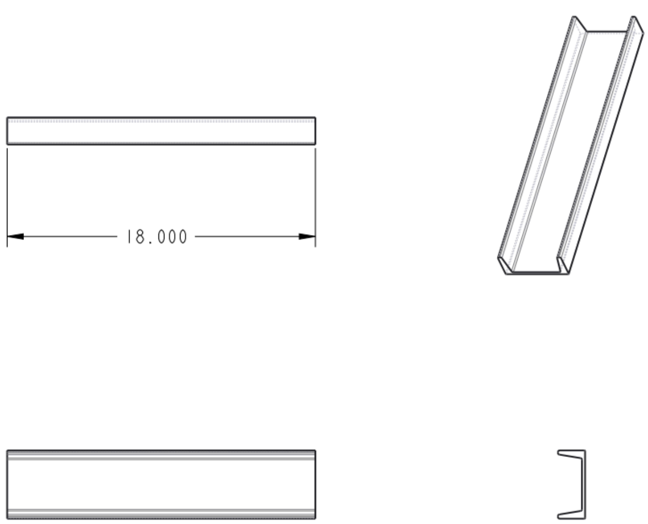
Babybot had a peg with a limit switch, as shown above in Fig III. F. 1. When Babybot touched the ground, the switch triggered activating its autonomous states. This first began with Babybot rotating to locate the light source of the deposition box. After locating the light, Babybot traveled towards the light, implementing proportional control. Upon reaching the deposition box, Babybot would measure the distance to the box. Once it reached the correctly calibrated state, Babybot activated the deployment mechanism by rotating a motor connected to the key-launch system. This projected the Medkit into the deposition box, after which Babybot rotated to locate Motherbot’s light source to return to Motherbot.

V. Drawings (Creo)

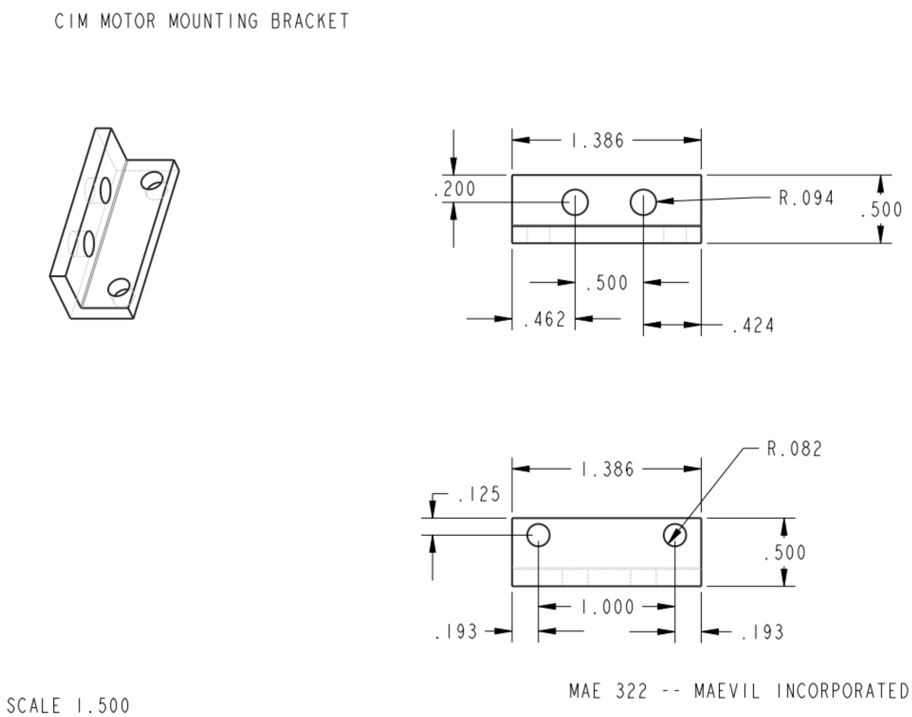
A. Critical Parts and Subassemmblies



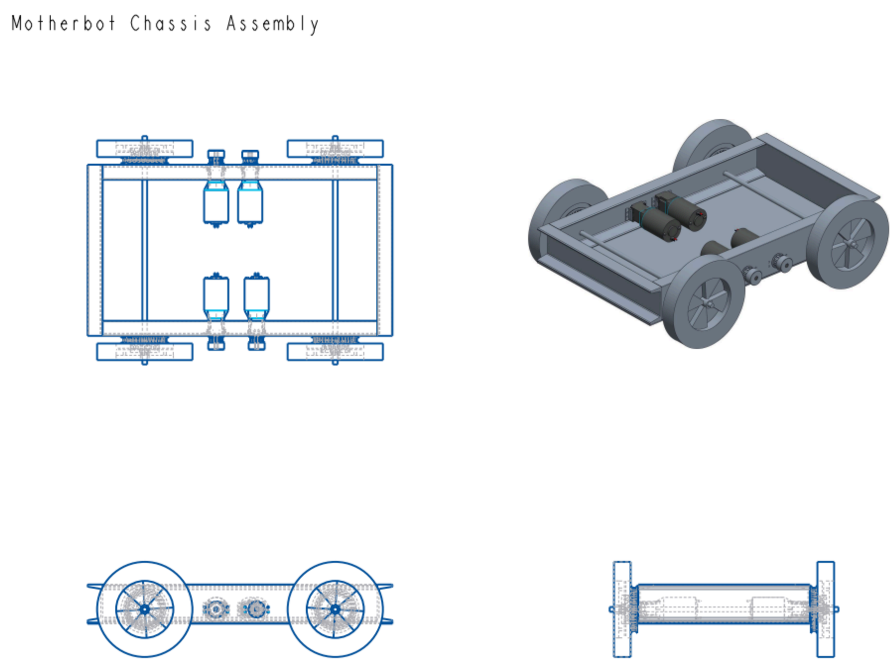
Main Motherbot Chassis Beam



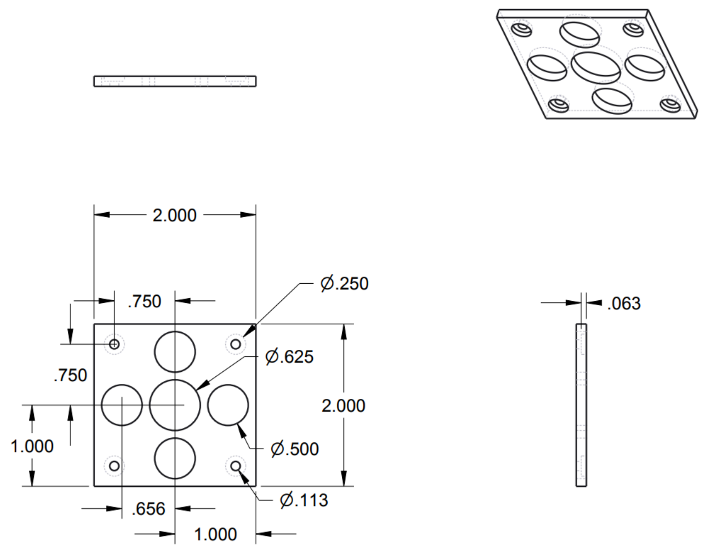
Motherbot Chassis Side Beam



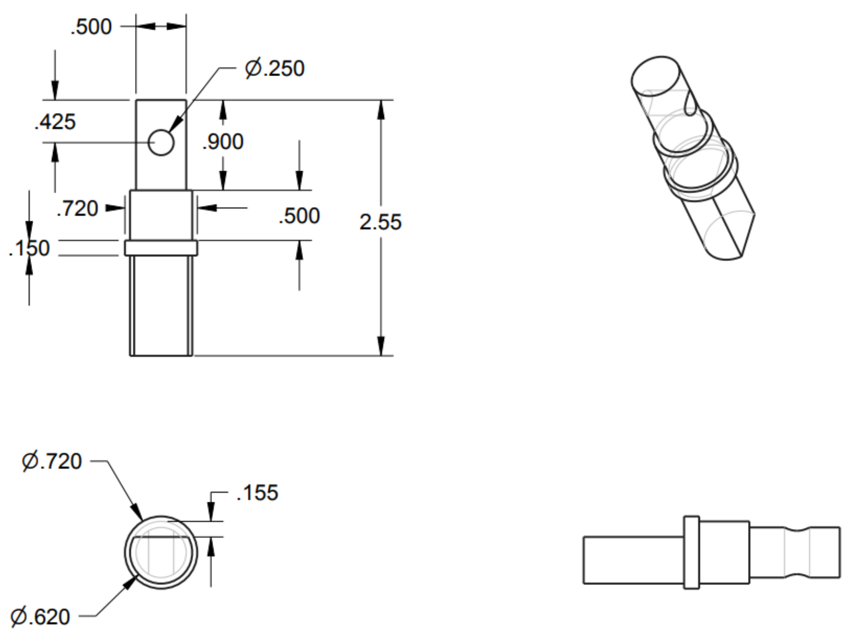
Motherbot Motor Mounting Bracket



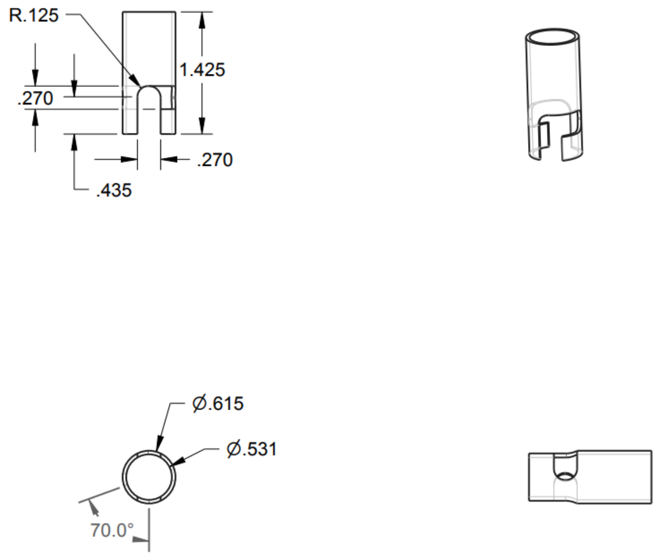
Motherbot Chassis and Drivetrain Assembly



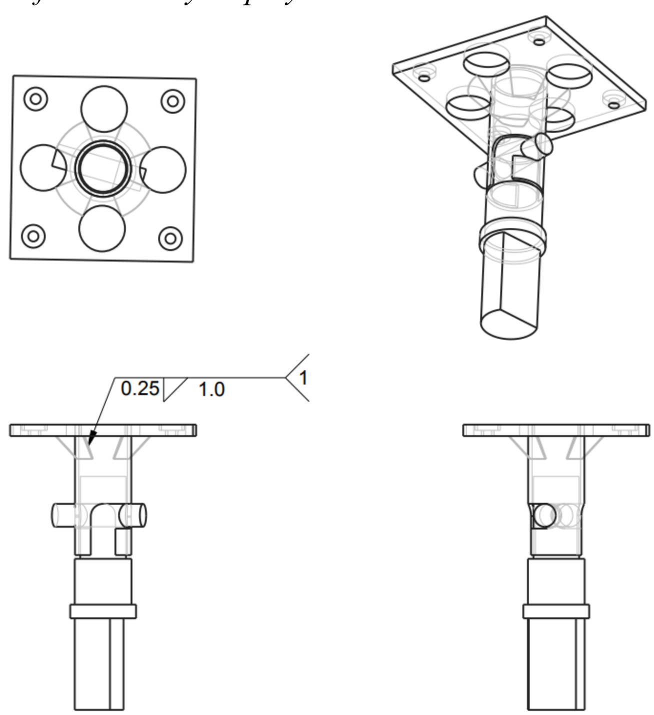
Part 1 of 3 of Medkit Key Deployment Mechanism



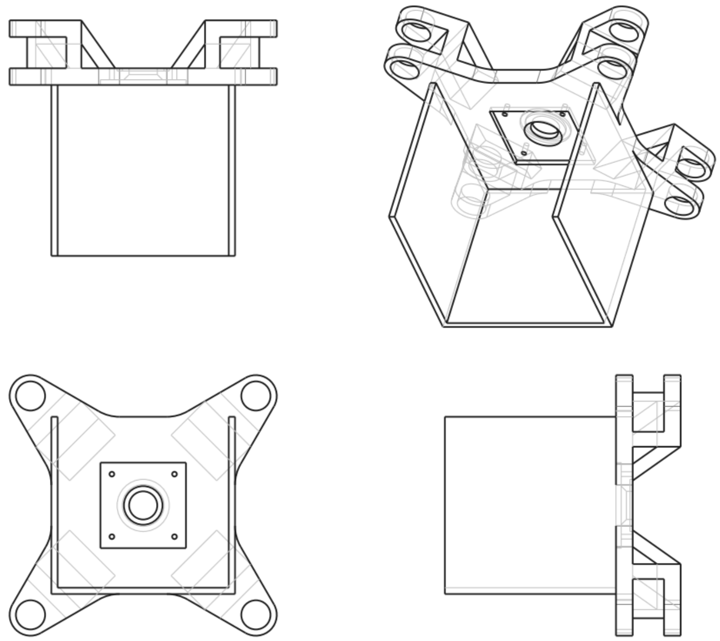
Part 2 of 3 of Medkit Key Deployment Mechanism



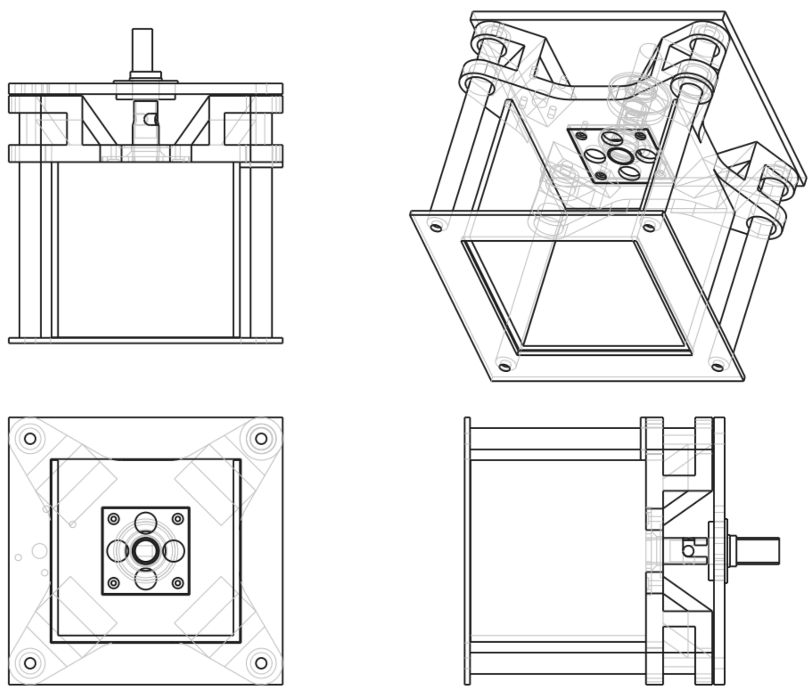
Part 3 of 3 of Medkit Key Deployment mechanism



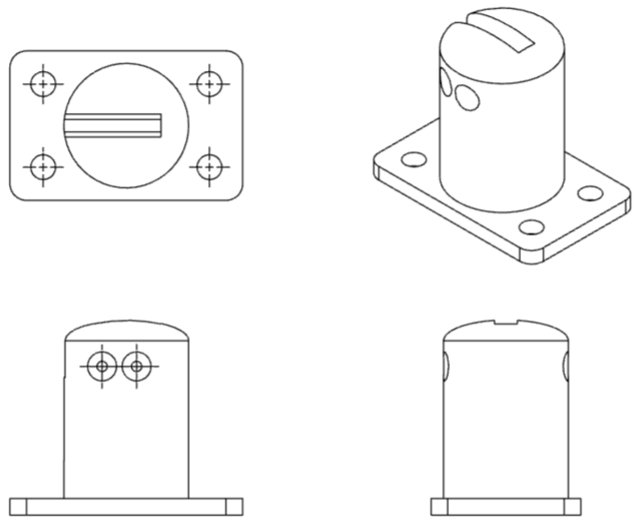
Assembled Medkit Key Deployment mechanism



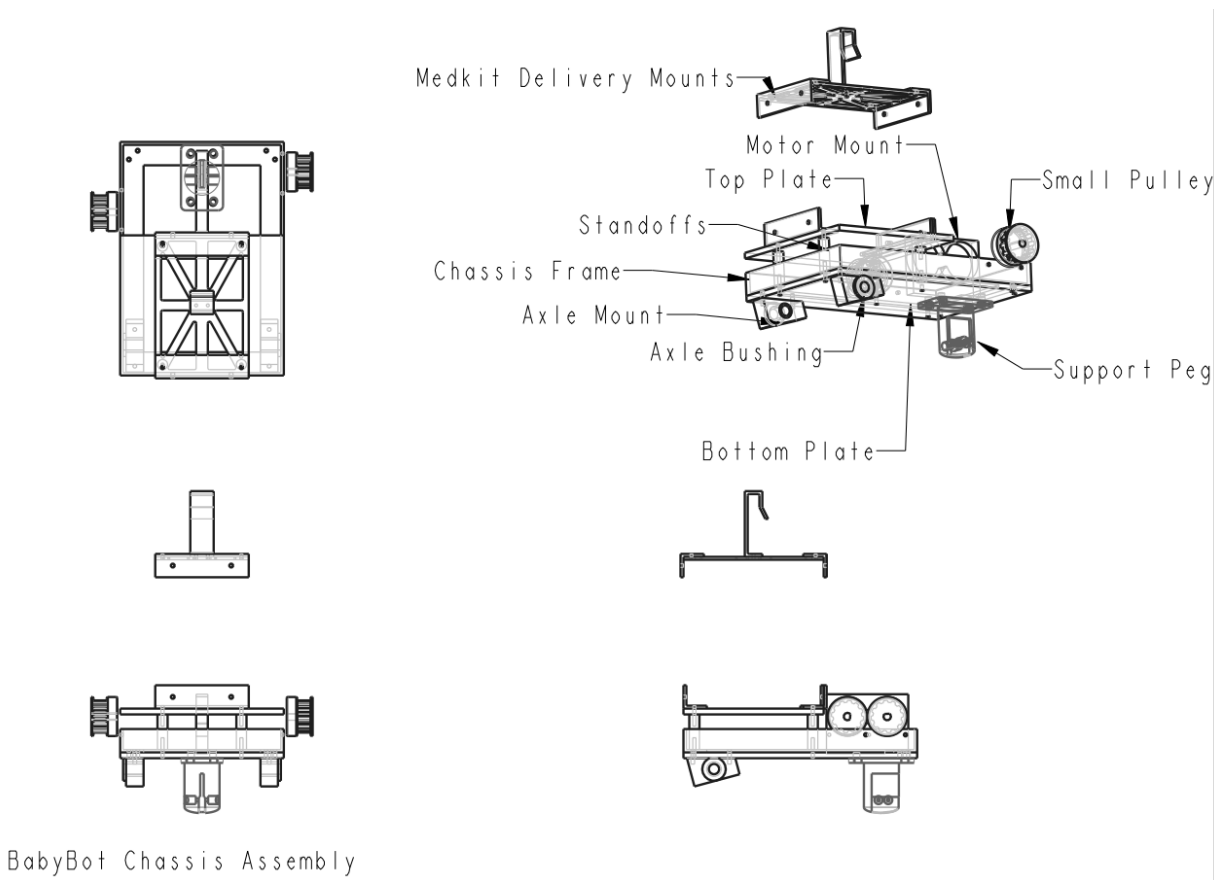
Carriage Drawing (No dimensions shown as this part was 3D printed)



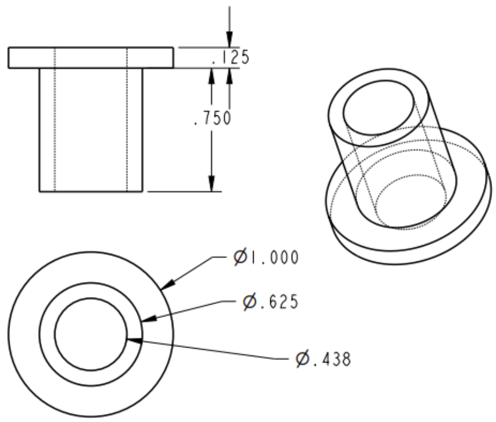
Full Medkit Deployment assembly



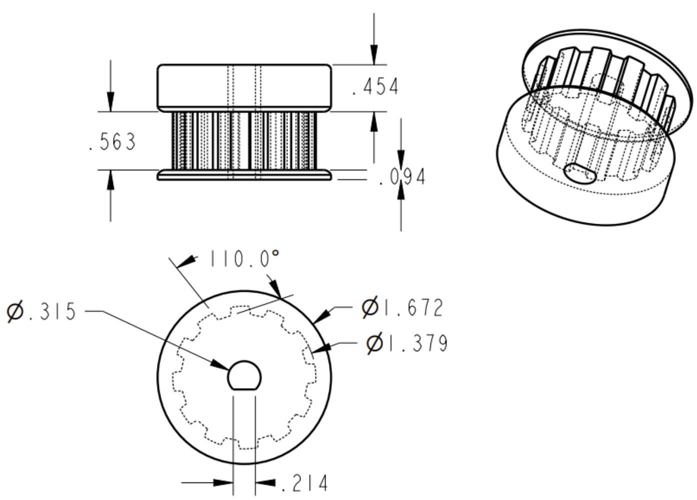
Babybot Foot Drawing (No dimensions shown as this part was 3D printed)



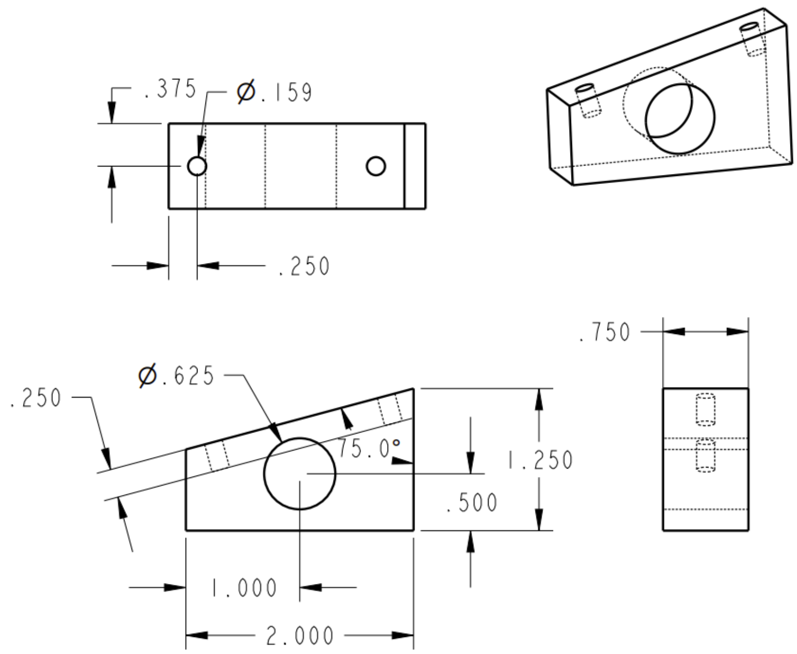
BabyBot Chassis



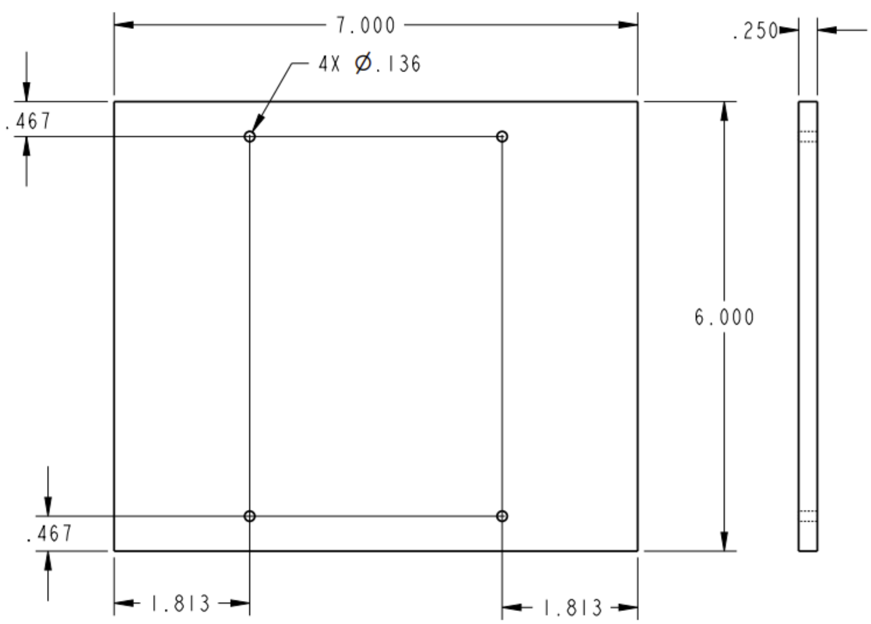
BabyBot Axle Bushing
Note: All dimesnsions held to ±0.005



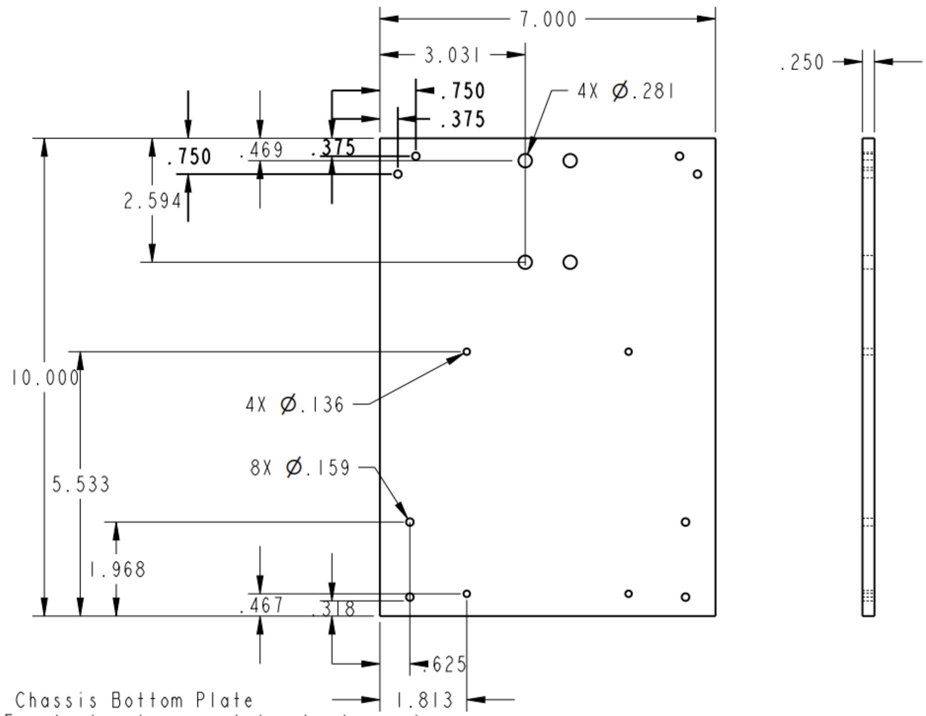
BabyBot Drivetrain Small Pulley



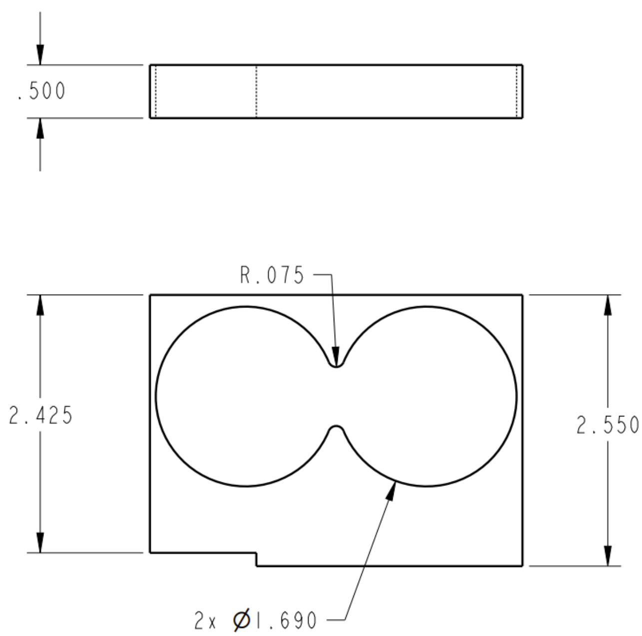
BabyBot Chassis Axle Mount
Note: All dimensions to be held to ±0.005



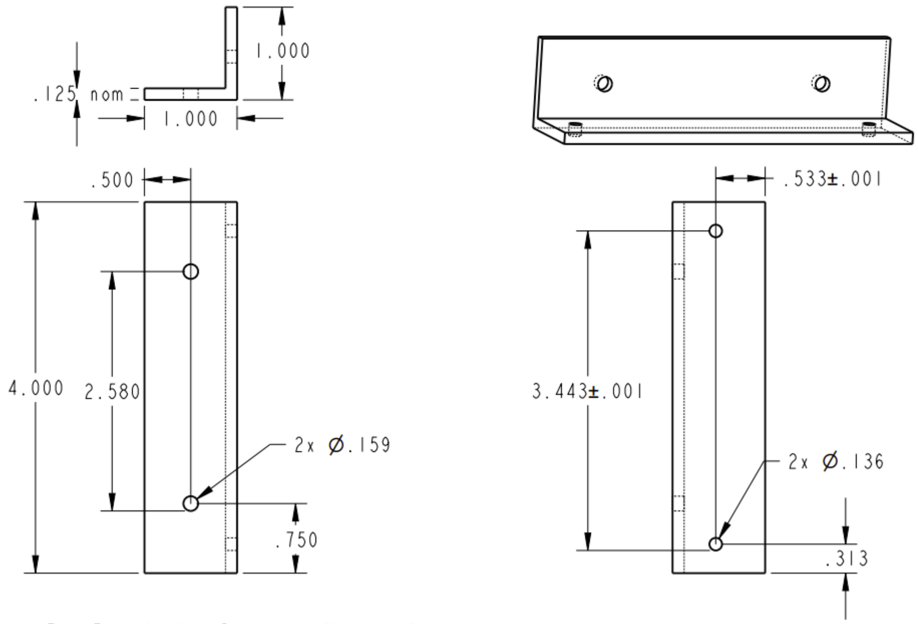
BabyBot Chassis Top Plate
Notes: All hole diameters to be held to +0.005
All distances to be held to ±0.001



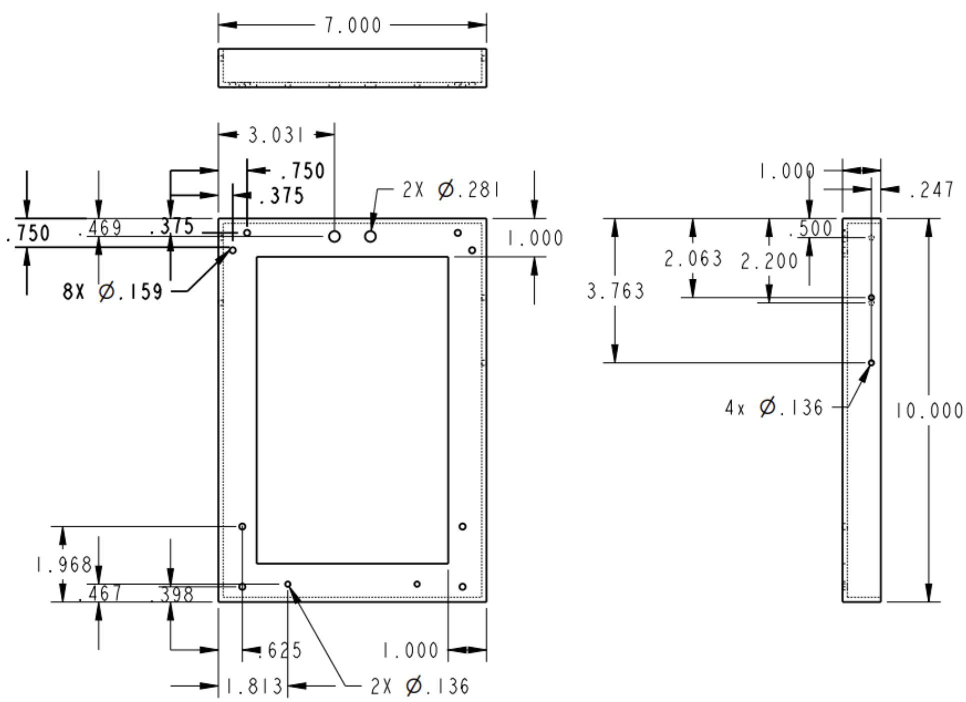
BabyBot Chassis Bottom Plate
Notes: Front view is symmetric about y-axis
All hole diameters to be held to +0.005
All distances to be held to ±0.001



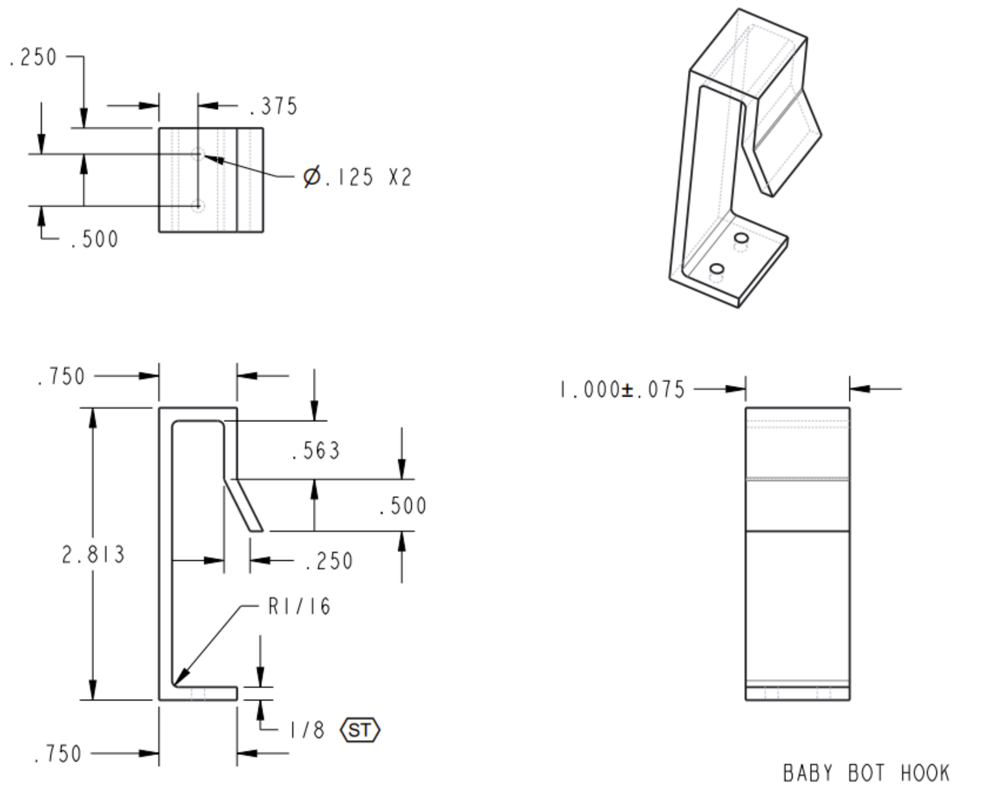
BabyBot Motor Mount
Note: All dimensions held to ±0.005



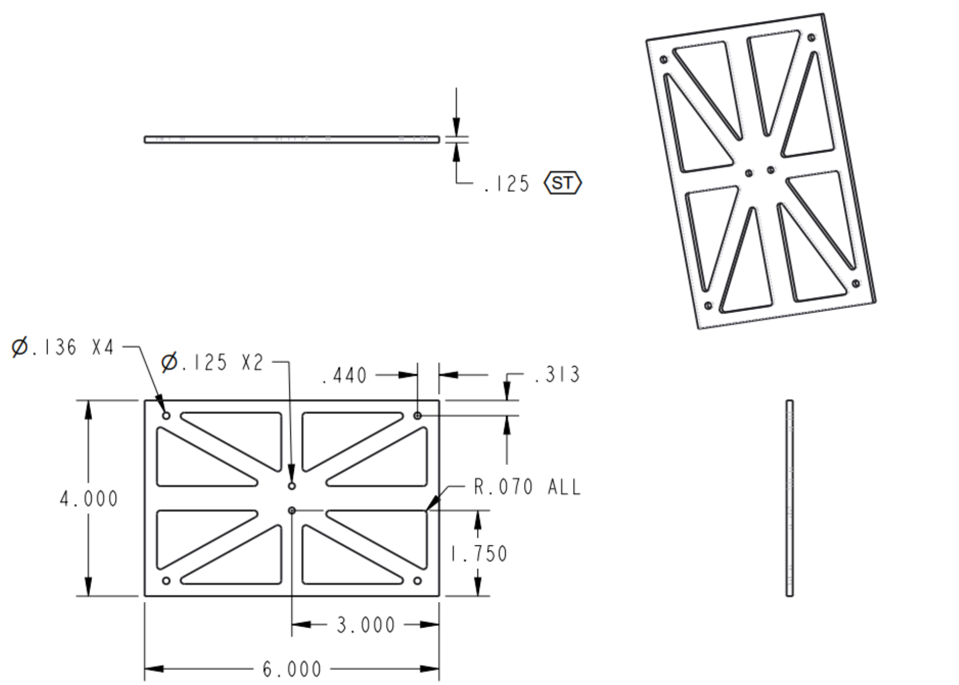
BabyBot MedKit Delivery Mount x4
Notes: Unless otherwise noted, all
dimensions to be toleranced to ±0.005



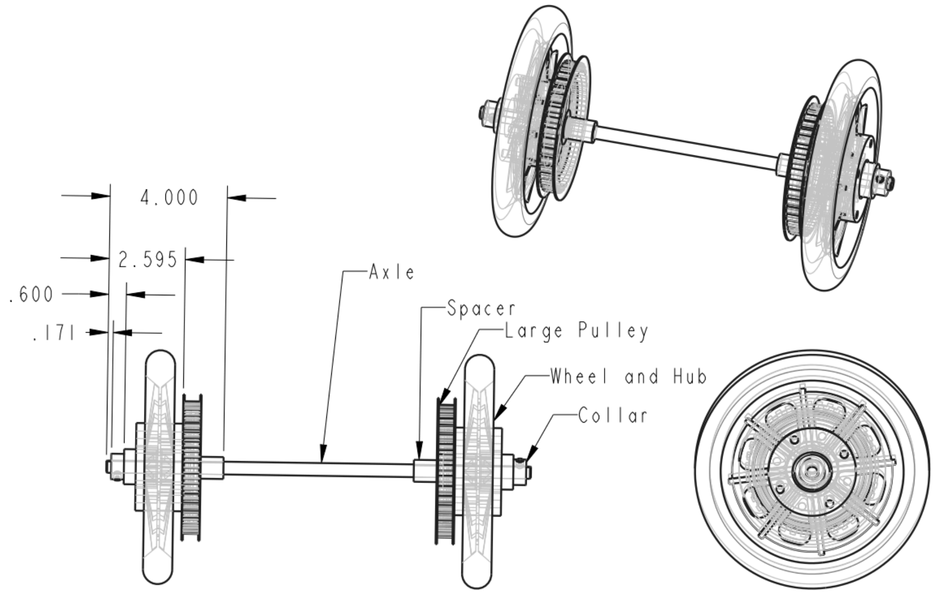
BabyBot Chassis Frame
Note: Front view is symmetric about y-axis
All hole diameters to be held to +0.005
All distances to be held to ±0.001



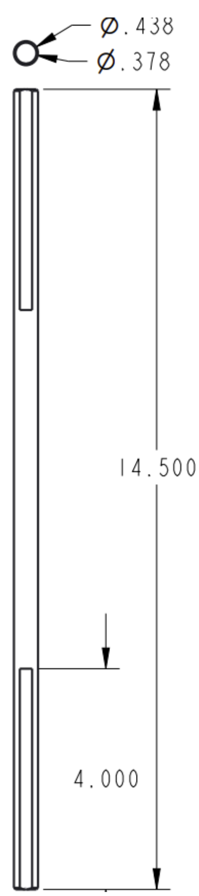
BABY BOT HOOK
(to be hand-bent from stock plate)



BABY BOT HOOK MOUNTING PLATE
(To be manufactured with a CNC mill)

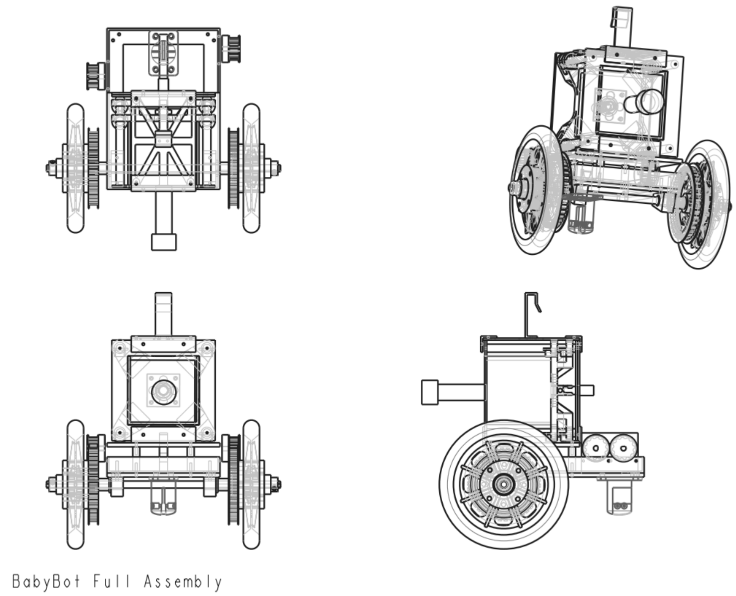


BabyBot Drivetrain
Notes: With exception of axle, all parts taken directly from sample robot



BabyBot Axle

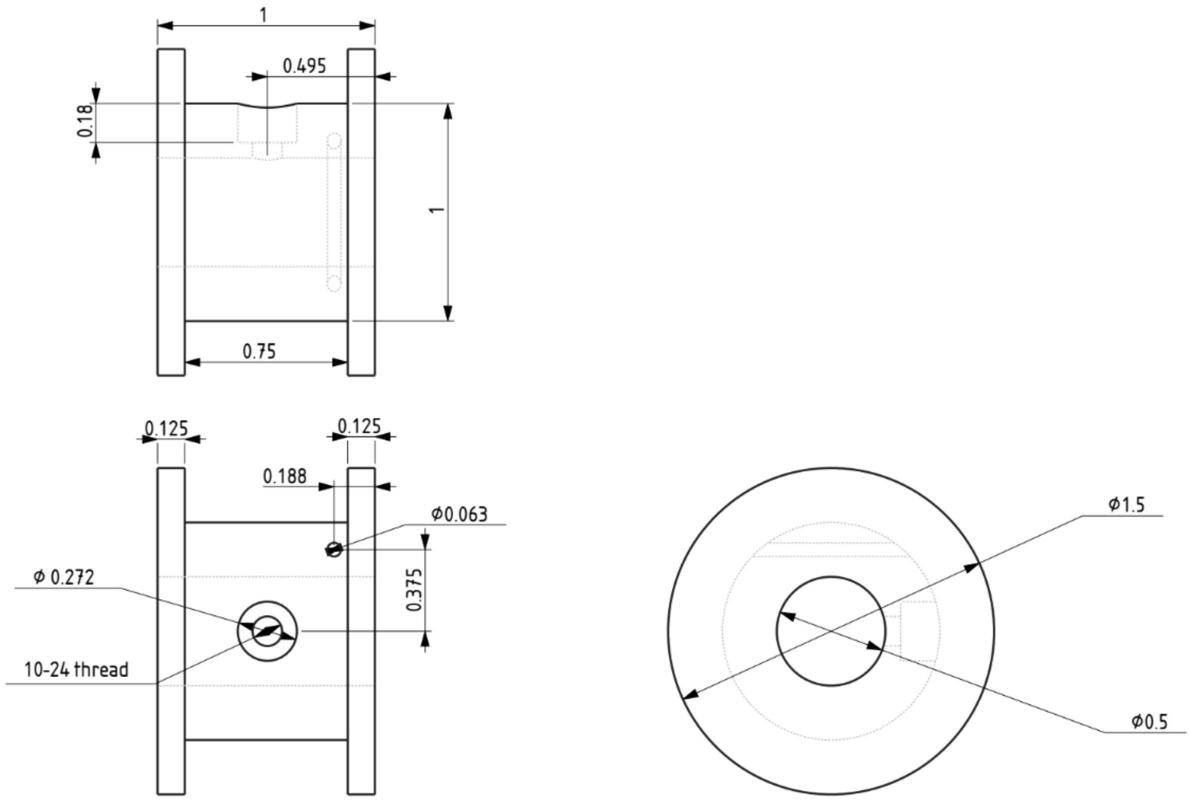
B. Main Assembly



C.

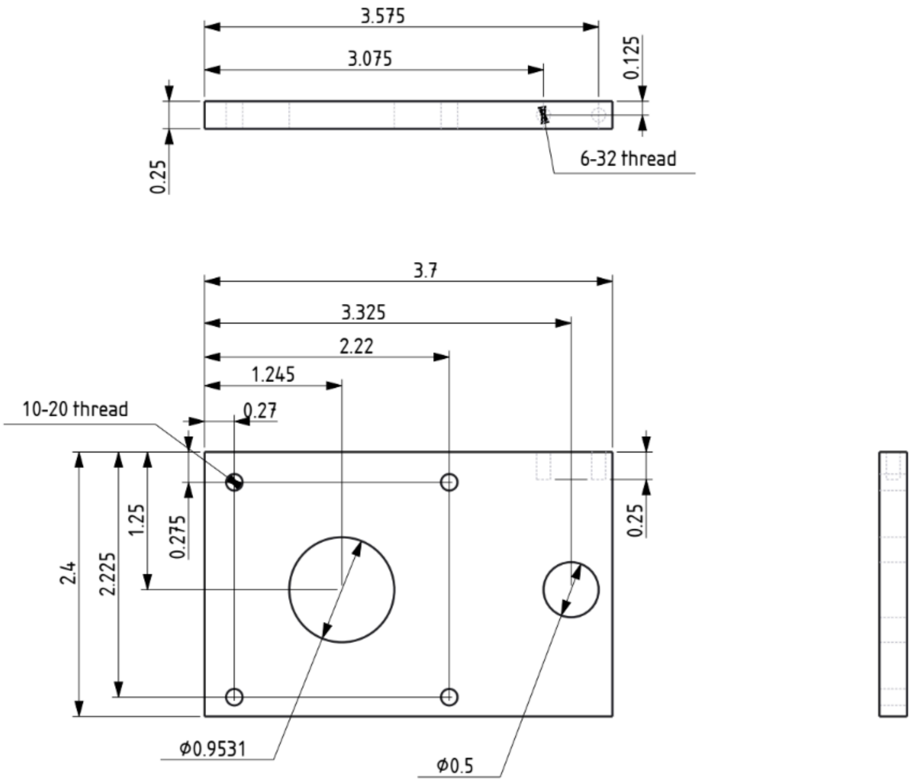
Lifting arm

Drum



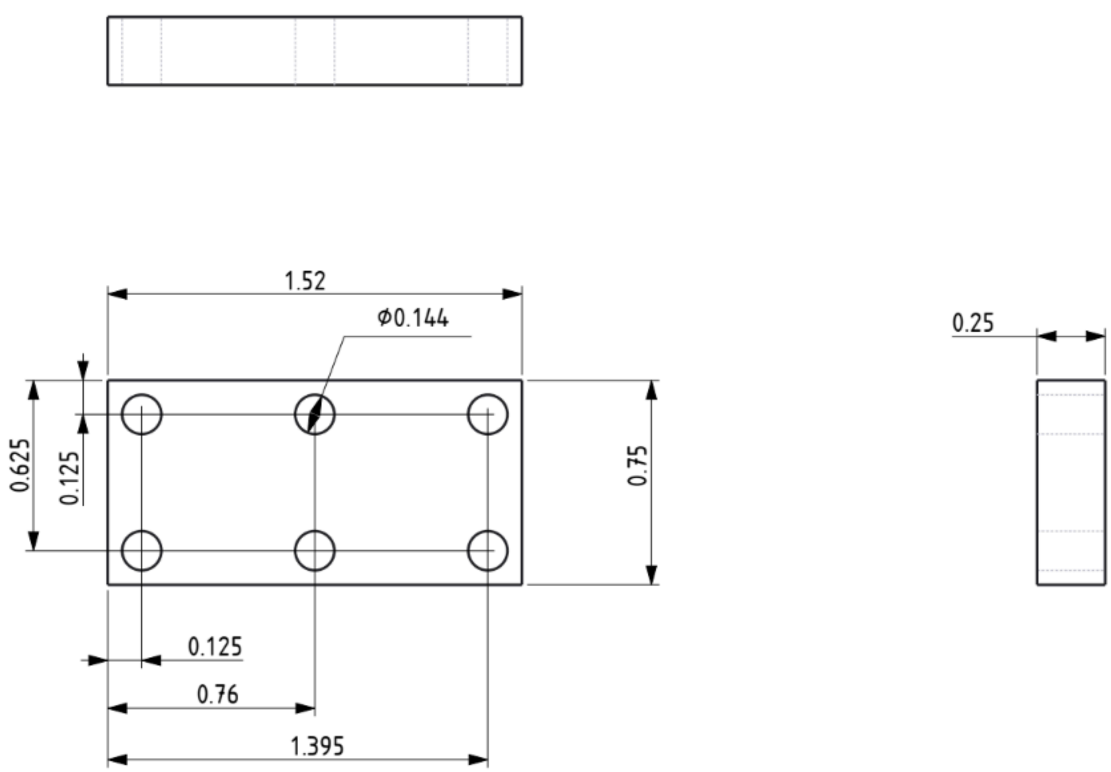
1
0.495
0.18
1
0.75
0.125
0.125
0.188
Φ0.063
Φ 0.272
0.375
10-24 thread
Φ1.5
Φ0.5

Motor mount

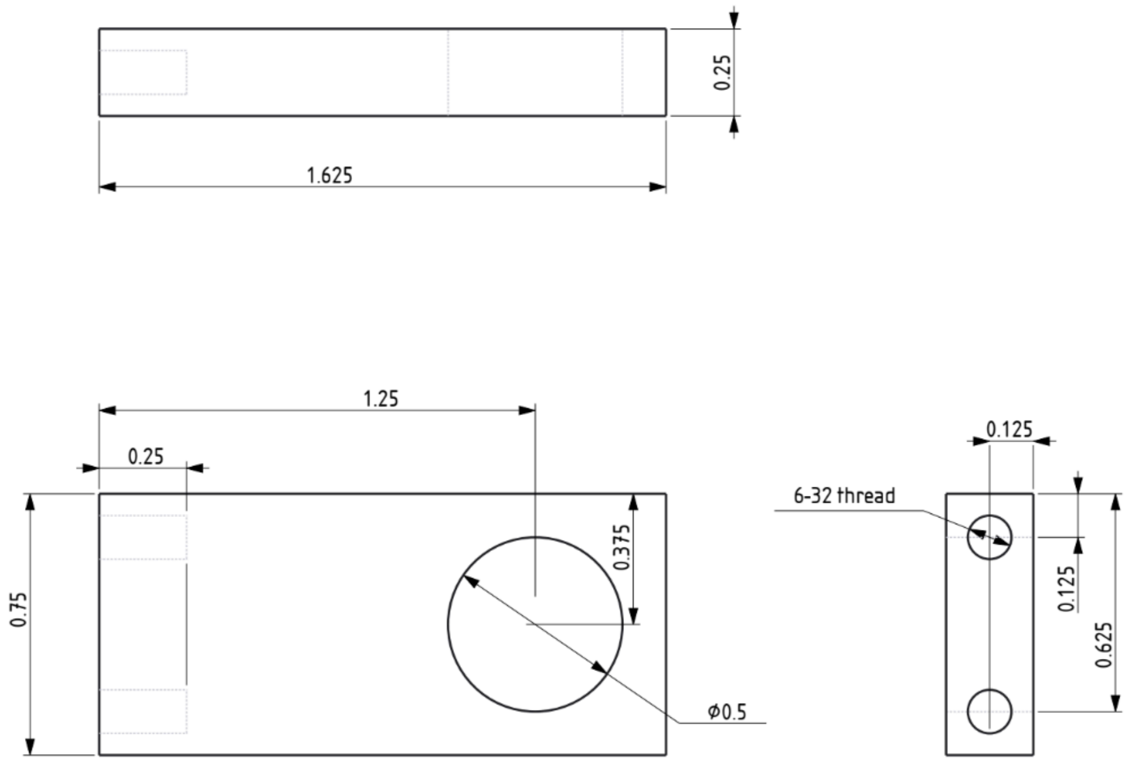


3.575
3.075
0.125
0.25
6-32 thread
3.7
3.325
2.22
1.245
10-20 thread
0.27
1.25
0.275
2.4
2.225
0.25
ϕ0.9531
ϕ0.5

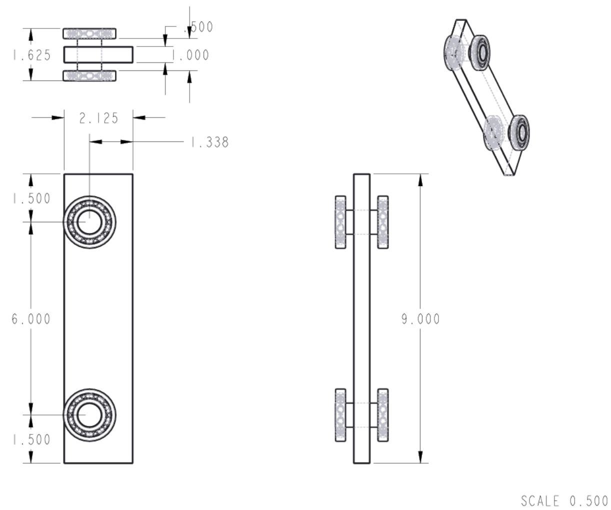
Connecting arm



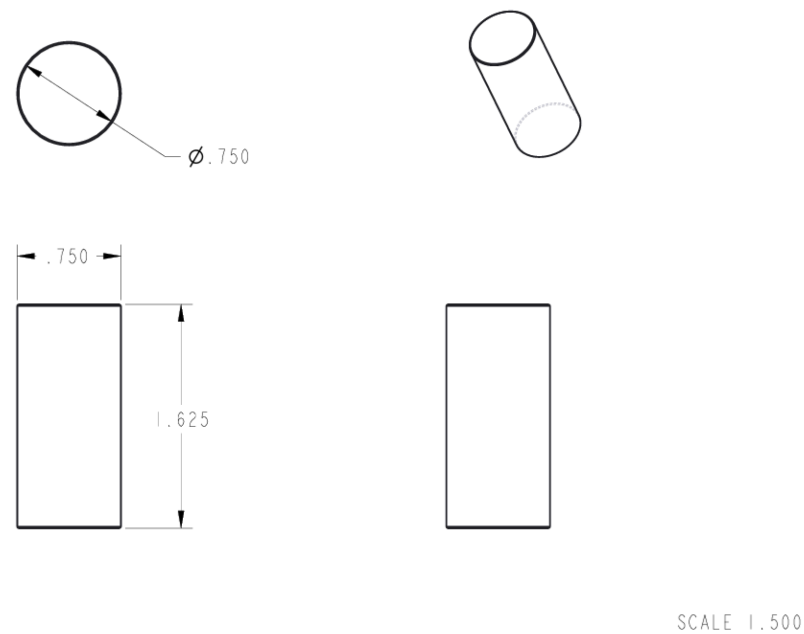
Motor plate 2



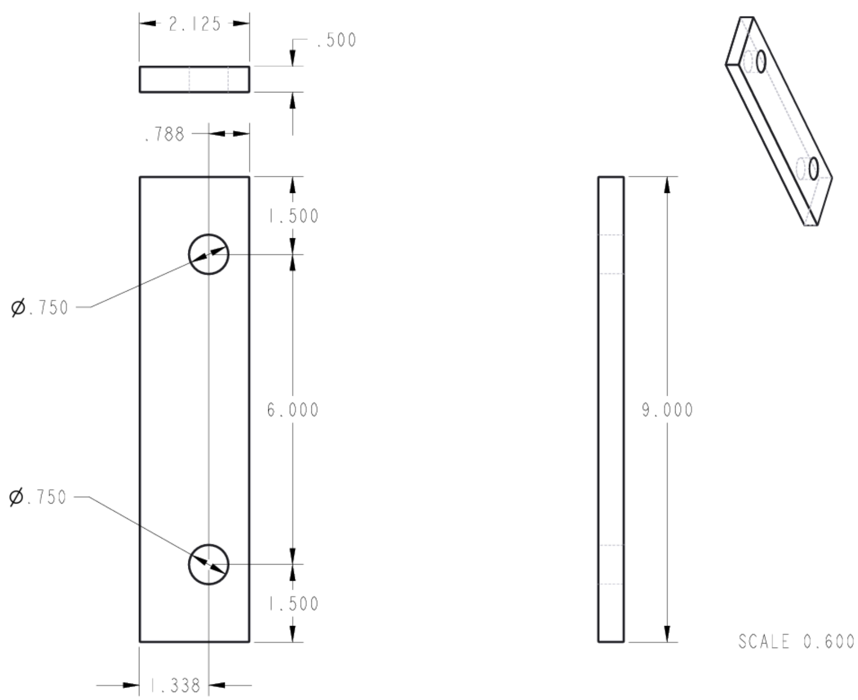
Carriage Assembly



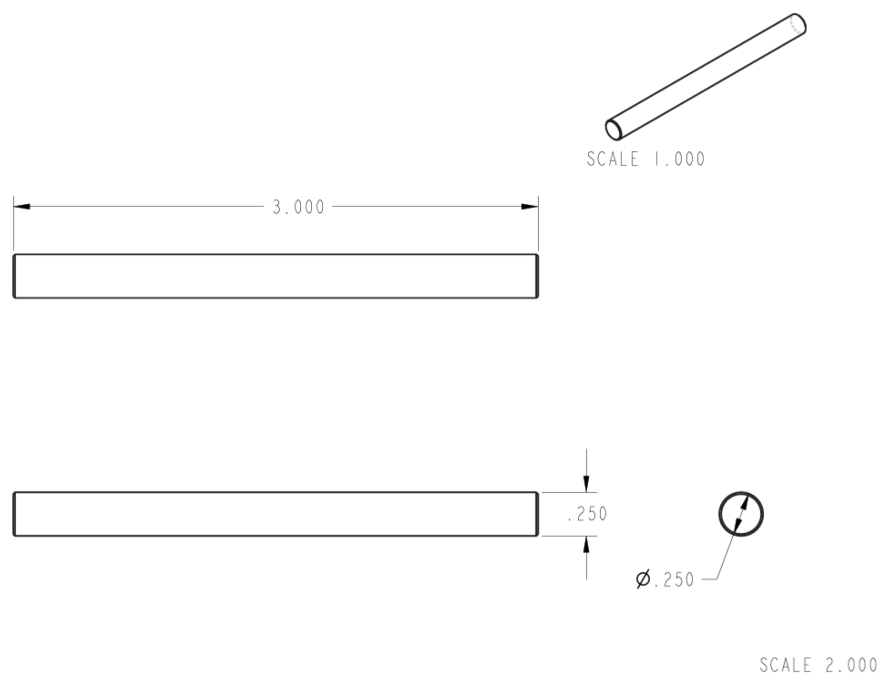
Carriage Axle



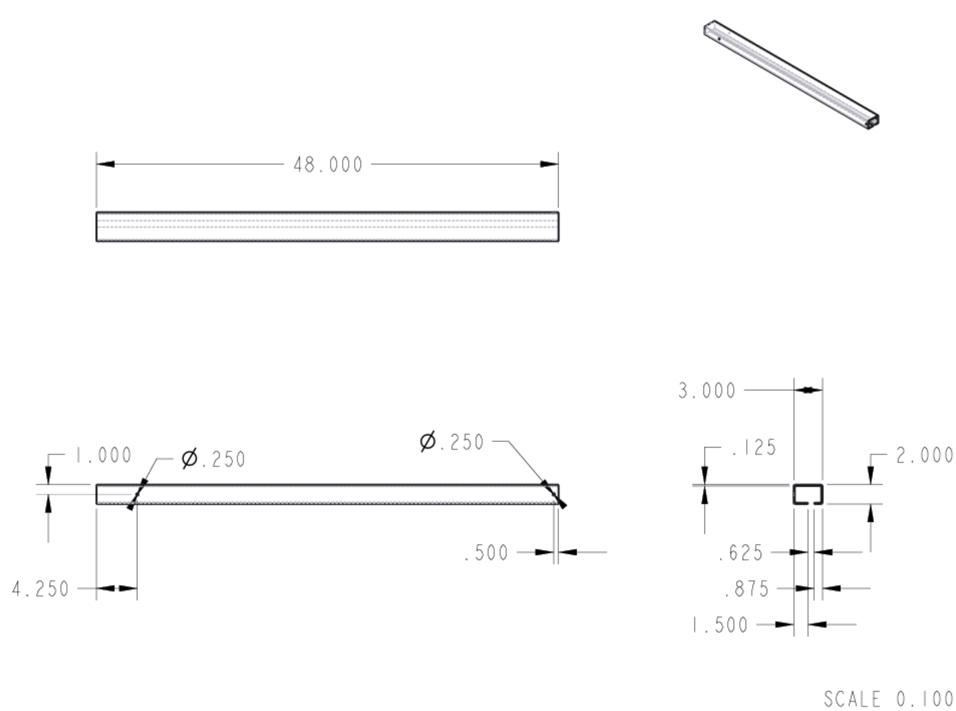
Carriage Body



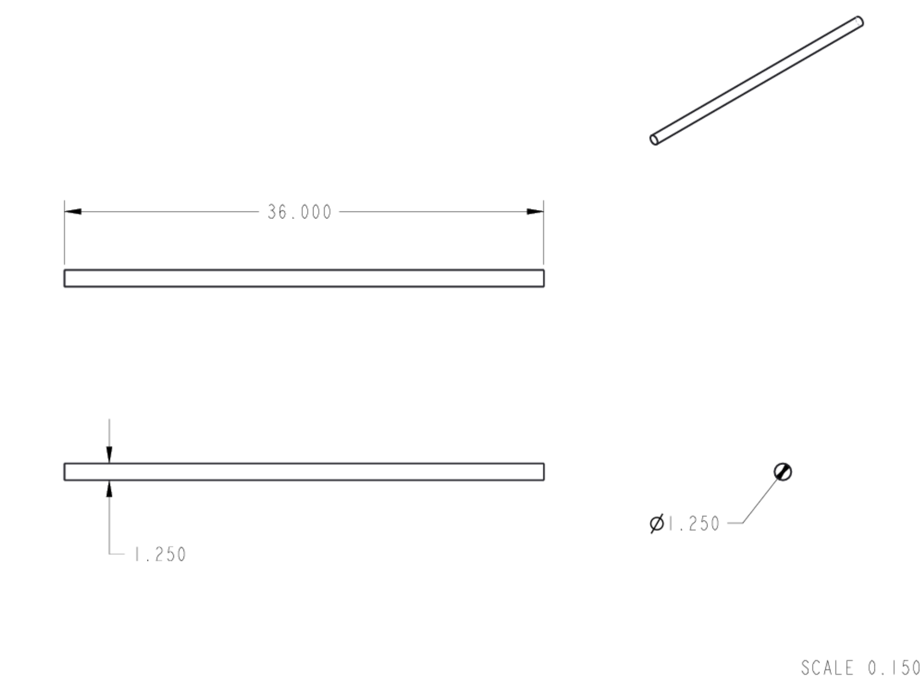
Lifting Mechanism Standoff



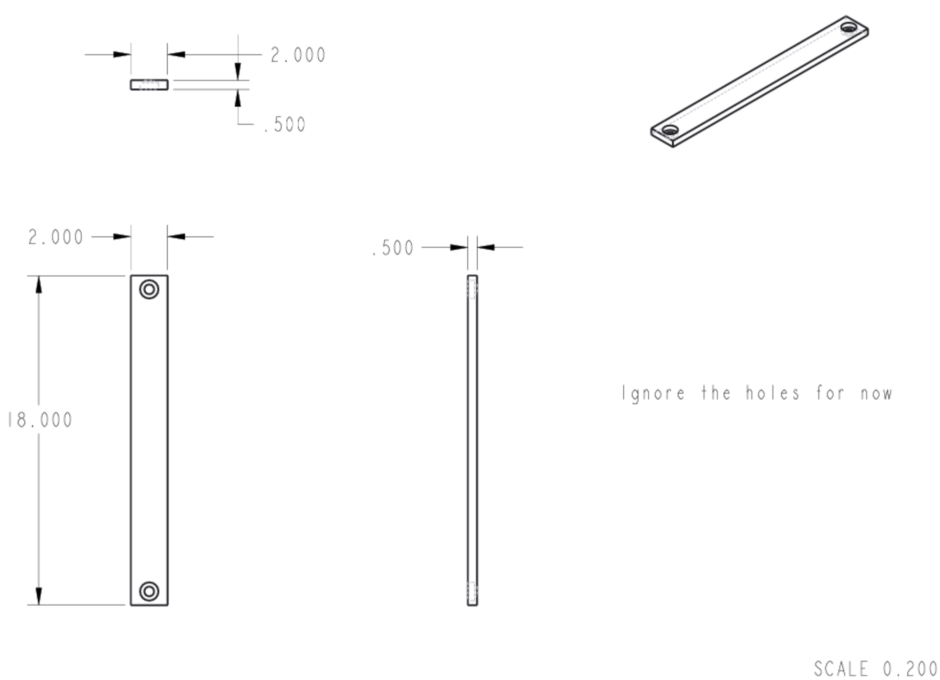
Lifting Mechanism Arm



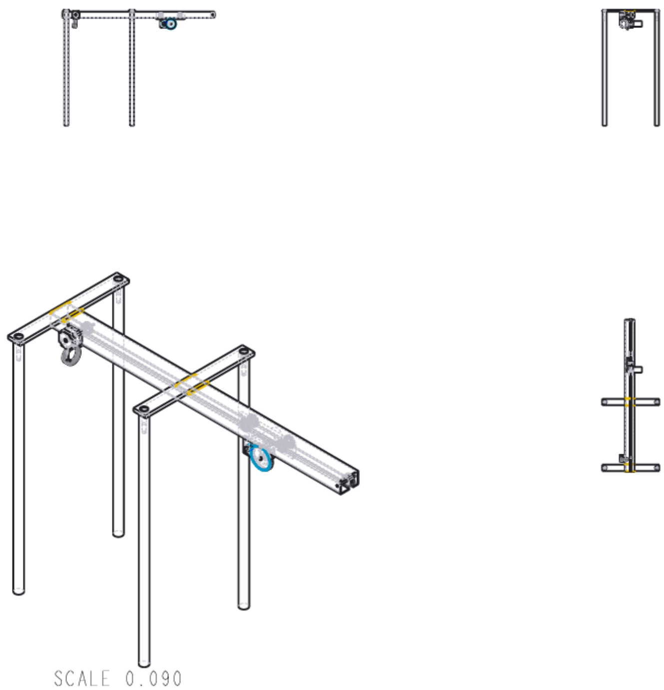
Lifting Mechanism Pulley Axle



Lifting Mechanism Cross-Beam



Lifting Mechanism Full Assembly



VI. Test Results

A. Ramp and Chute Navigation

Motherbot was able to navigate over the ramp and through the shoot while carrying baby bot. It would take around a minute and a half for motherbot to navigate to the wall. The size and weight of mother bot did cause some issues with the ramp. Mother bot would shift the ramp when trying to drive over it leading to the ramp being misaligned with the chute, during one of the tests mother bot drove into the wall as a result of the ramp being shifted over. To mitigate this issue we ended up holding the ramp in place during tests.

B. Wall-Breaching

When we tested the wall breaching mechanism we found that it was able to raise and lower baby bot in a controlled manner. It took about 18 seconds for the carriage to lift baby bot and around 37 seconds to lower baby bot.The breaching mechanism was successfully able to move baby bot to the other side of the wall. It took around 10 seconds for the carriage to move to the other side of the wall and start lowering baby bot. After several tests we found that the carriage stopped moving, which led to us noticing that the shaft connecting the motor to the belt pulley had bent. We believe that the shaft bending had relieved the tension on the belt allowing for the belt to skip on the pulley. To fix this we manufactured a thicker shaft, we were unable to test with the new shaft because we ran out of time.

C. Medkit Deposition

When testing baby bot, our method of medkit deposition, it was able to work well and constantly. It was able to consistently navigate from the wall to the drop off location using the guide light. The launching mechanism also worked well and constantly, being able to deposit the medkit into the designated area. We did not have to make any major modifications to baby bot as a result of the medkit deposition testing.

D. Autonomous Operations

Fully autonomous navigation of the entire course was never realized during testing due to time constraints as well as electrical and mechanical failures. However, each individual element of the course was able to be successfully completed in autonomous mode during both the days leading up to the final presentation as well as on the testing day. With proper time to calibrate the software to the conditions of the testing location, as well as hardware repairs, complete autonomous navigation of the course would be possible.

VII. Conclusions and Further Work

In this project, we set out to design and build an autonomous search and rescue robot capable of navigating complex terrain, breaching a vertical obstacle, and delivering a medical payload with minimal human intervention. Our system used a marsupial design consisting of a large Mother Bot and a smaller Baby Bot, enabling terrain traversal and precise medical kit delivery. The Mother Bot used a steel chassis and a skid-steer drivetrain with high gear reduction to transport both robots up a ramp, through a narrow chute, and to deploy the Baby Bot over a wall using a controlled lifting mechanism. The Baby Bot independently navigated to the target and delivered a 1 kg medkit using a spring-powered deployment system, then returned for recovery. Both robots relied on light and distance sensors for autonomous navigation, controlled by Teensy microcontrollers. The full system met size, weight, and cost constraints and successfully demonstrated autonomous navigation, wall breaching, and medkit delivery during testing.

During our final testing day, our robot was capable of autonomously completing all required course obstacles and each subsystem functioned as intended when evaluated individually, demonstrating that the mechanical, electrical, and control architectures were fundamentally sound. However, the system did not complete the entire course cohesively in a single uninterrupted autonomous run. This limitation was not due to a failure of the overall design concept, but rather due to practical issues that could have been resolved with more testing that was unfortunately unable to be conducted due to the constraints of a compressed testing timeline. The most significant issue encountered during testing was a failure within the motor system, which prevented the team from attempting a final cohesive autonomous run of the course. As we did not have the opportunity to drive Mother Bot for an extended period of time during the testing period, we were unable to identify this issue before the final test day. Post-test analysis suggested that, although the drivetrain was protected by 40 A fuses, these fuses allowed brief current spikes of approximately 80–100 A to pass through during high-load events such as ramp traversal and turning. These spikes caused the motor controllers to temporarily shut down to protect themselves from overheating, requiring cooldown periods between runs.

Future work would focus on improving the system reliability and integration to enable fully continuous autonomous operation. The most immediate improvement would be redesigning the electrical system to better handle high current demands, such as distributing the drivetrain load across additional motor controllers or implementing current limiting to reduce peak loads. Additional closed loop feedback for the lifting and deployment mechanisms would increase robustness during the wall breaching and recovery stages. Finally, further testing over longer continuous runs would help identify and mitigate integration level issues earlier in the development process. Together, these improvements would build on the demonstrated strengths of the current design and move the system closer to a fully cohesive and field ready search and rescue platform.

VIII. Appendix

Planning	Design for Lifting Mechanism complete
	Design for Chassis complete
	Design for Baby Bot: complete
	Electrical Schematics complete

Prototype	Lifting Mechanism Prototype complete
	Prototype Baby Bot frame for lifting mechanism to grab
	Lifting Mechanism Prototype actually grabs Baby Bot frame
	CdS Sensors work
	Distance sensors work
Revisions	Approve all final prototypes
Manufacturing	Lifting Mechanism Built
	Motherbot Chassis built
	Baby bot chassis built
	Baby bot medkit delivery system built
Integration I	Mother bot chassis is powered and driving
	Baby bot chassis is powered and driving
	Lifting mechanism is powered and lifts Baby
	Lifting mechanism is attached to Mother bot and works
	Babybot delivers medkit
	Stick CdS sensors on Mother bot and get readings on teensy
	Stick CdS sensors on Baby bot and get readings on teensy
	Stick distance sensors on Baby bot and get readings on teensy
	Stick distance sensors on Mother bot and get readings on teensy
Integration II	Mother bot navigates to the light
	Baby bot navigates to the light
	Mother bot navigates the ramp and chute
	Mother bot lifts Baby bot over wall
	Mother bot autonomously releases Baby bot
Testing Stage	Mother bot navigates ramp and chute fully loaded with arm and Baby
	Baby delivers medkit to target successfully
	Full Course Test Complete

Table V.1: Milestone List

Stages	Lifting Mechanism	Motherbot (Chassis/Drivetrain)	Babybot	Integrated Electrical Systems
	Finalize design of arm itself	Finalize design of drivetrain	Finalize design of drivetrain	Make schematics
	Finalize design of Baby gripper	Finalize design of chassis/lifting mechanism attachment point	Finalize design of chassis	
	Calculate dynamics of arm	Do dynamics calculations (PDR)	Finalize design of grip-attachment thingy	
	Calculate motor requirements	Calculate motor requirements	Finalize design of medkit delivery system	
	Select Motors/actuators	Select Motors	Do dynamic calculations for drivetrain	
	Calculate center of mass	DO THE CAD	Do dynamic calculations for arm-attachment system	
	DO THE CAD		Do dynamic calculations for medkit delivery system	
			Calculate motors for drivetrain	
			Calculate motors for arm-attachment system	
			Calculate motors for medkit delivery system	
			Select motors/actuators for everything	
			DO THE CAD	
Planning				
	Prototype arm (static or dynamic arm)		Protoype medkit delievery system	Get CdS sensors cooking
Prototype	Prototupe baby gripper		Prototype attachement point to baby gripper	Get distance sensors cooking
	Revise arm design and attachment point		Revise medkit delievery system design	
Revisions	Revise baby gripper		Revise attachement point to baby gipper	
	Build arm with attachment point	Build chassis	Build babybot chassis	
	Build gripper	Build drivetrain in chassis	Build babybot attachement point to baby gripper	
Man! You facturin'?			Build medkit delievery system	
	Add gripper to main arm assembly	Add motors to chassis	Assemble parts of babybot together	Figure out how to contorl motors
	Add motors/actuation to gripper	Set up electrical system on motherbot	Add drivetrain motors	
	Add motors to main arm	Get chassis driving	Add medkit delievery motors/actuation	
	Get lifting mechanism electricalized	Attach CdS sensors	Add arm attachement actuation (if needed)	
	Get lifting mechanism to move Baby	Attach distance sensors	Get babybot moving open loop	
	Get lifting mechanism to deliver Baby		Get delievery system working	
	Get lifting mechanism to recover Baby		Attach CdS sensors	
Integration I			Attach distance sensors	
Integration II	Get Mother to bot lift Baby bot over the wall	Get light navigation working on Mother bot	Get light navigation working on Baby bot	
	Control autonomous release of Baby bot	Implement ramp navigation using distance sensors		
	Control autonomous recovery of Baby bot	Implement chute navigation using distance sensors		
	Test mother bot navigating the ramp and chute with arm holding Baby fully attached			
	Test mother bot in the drop test whilt Baby fully attached			
Testing		Motherbot navigates chute and ramp with BabyBot	Baby bot navigates to light and delivers Med Kit	
	Full test course complete			

Table V.2: Task List

For better viewing this table, see it here under 'Tasks': [PDR] Timeline and Milestones

	Week 1 (October 19-25)		Week 2 (October 26 - Nov 1)		Week 3 (Nov 2-8)		Week 4 (Nov 9-15)	
	Task	Hours	Task	Hours	Task	Hours	Task	Hours
	Schematics of electrical systems (dependent on designs being fi	5	Finish schematics	4	Continue programming logic for sensors	2	Get Baby Bot driving open loop	4
			Get CdS sensors cooking	2	Figure out how to control motors	2	Complete wiring harness for mother	3
			Get distance sensors cooking	2	Start wiring harness for Mother	2		
					Complete wiring harness for baby	4		
Electrical								
	Finalize design of arm itself	3	Select motors/actuators	2	Prototype baby gripper	4	Revise baby gripper CAD	4
	Finalize design of Baby gripper	3	Finish CAD of baby gripper	6	Finish arm cad	7	Prototype arm desing	5
	Calculate dynamics of arm	2	Start CAD of arm	4			Start revising arm CAD	3
	Calculate motor requirements	2						
	Calculate Center of Mass	2						
Lifting Mechanism								
	Finalize design of drivetrain	2	Select motors	2	Finish CAD of chassis	5	Manufacture drivetrain	5
	Finalize design of chassis/lifting mechanism attachment point	3	Start CAD of drivetrain	5	Finish CAD of drivetrain	5	Manufacture chassis	5
	Do dynamics calculations (PDR)	3	Start CAD of chassis/lifting mech connection point	5	Plan drivetrain manufacturing	2		
	Calculate motor requirements	2			Plan chassis manufacuturing	2		
MotherBot								
	Finalize design of drivetrain	2	Select motors/actuators for everything	3	Manufacture chassis/drivetrain	6	Prototype medkit delievery	5
	Finalize design of chassis	2	Finish CAD of chassis/drivetrain	9	Assemble chassis/drivetrain	4	Revise baby gripper attachment point	4
	Finalize design of grip-attachment thingy	3	Start CAD of baby gripper attachement point	3	Add motors to drivetrain	2	Revise medkit delievery system	6
	Finalize design of medkit delivery system	4	Start CAD of medkit delievery system	5	Finish CAD of gripper attachment point	4	Manufacture baby gripper attachment point	5
	Do dynamic calculations for drivetrain	3	Prep for manufacturing of chassis/drivetrain	2	Finish CAD of medkit delievery system	8	Prototype baby gripper attachment point	4
	Do dynamic calculations for arm-attachment system	2						
	Do dynamic calculations for medkit delivery system	2						
	Calculate motors for drivetrain	2						
	Calculate motors for arm-attachment system	2						
BabyBot	Calculate motors for medkit delivery system	2						
	PRELIMINARY DESIGN REPORT & PRESENTATION		ALL ORDERS FINALIZED					
Logistics								
		47		54		59		53

Week 6 (Nov 23-29) (TG Break)		Week 7 (Nov 30-Dec 6)		Week 8 (Dec 7-13)	
Task	Hours	Task	Hours	Task	Hours
Attach electricals to Baby bot chassis	2	Get light tracking mechanism working in Mother	5	Implement ramp navigation using distance sensors	5
Get Baby Bot grippers working (IF THEY EXIST)	4	Get light tracking mechanism working in Baby	5	implement chute navigation using distance sensors	5
Get Baby medkit delivery working autonomously	3	Get lifting mechanism working open loop	3	Get lifting mechanism to autonomously deliver Baby over wall	7
Attach CdS sensors to baby	1	Get lifting mechanism to deliver Baby open loop	3	Get lifting mechanism to autonomously recover Baby	3 (low importance)
Attach distance sensors to baby	1	Get lifting mechanism to recover Baby open loop	2	Get Baby to autonomously navigate back to wall	3
Add motors to main arm	2	Test motherbot lifting babybot over wall	4	Test complete lifting mechanism	7
Add motors to gripper	2				
Add gripper ot main arm assembly	2				
**help arm manufacture/electrical				Test complete motherbot	7
**help arm manufacture/electrical		Assemble all parts of babybot together	4	Test complete babybot	7
				Test full course	10
	17		26		41

Table V.3: Timeline

For better viewing this table, see it here: [PDR] Timeline and Milestones