Project 3 Global Emergency Autonomous Response System (GEARS)

Team 03

Lincoln Alicea, Emily Kalin, Amanda Pan, Katherine Pesetski

Cover Letter

Team 03

1101 Third St. West Lafayette, IN 47906

May 2, 2025

Dr. Lendra Wilson
Lead Engineer for L3Harris Disaster Response Division

1025 W. NASA Boulevard, Melbourne, Florida 32919

Dear Dr. Wilson,

The team's Global Emergency Autonomous Response System (GEARS) consists of a simple design consisting of a variety of sensors and a simple but effective chassis and cargo system combined with traditional map-solving algorithms. This prototype had some struggles navigating the maze without contacting debris, but it is believed that this could be solved with a simple feedback control system.

The team decided to go with a two-motor, four-wheel design. The motors were located on the front left, front right, back left, and back right of the robot. The wheels on the back were 68.8 centimeters in diameter and were connected directly to the axle of the motors. The wheels on the front of the robot were connected to the back axles by a train of four gears. This was an important decision because this allows the prototype to have pinpoint turning, crucial for a tight maze. When turning, the wheels on the inside of the turn will turn backwards, and the wheels on the outside of the turn will turn forwards. This was proven to work, as the robot was able to turn without moving laterally in testing and in the final demonstration.

A battery holder was constructed on the pi case on the top of the robot. This case is a rectangular prism laid on its long side with an open top. This gave an opening for the battery to insert into the holder. The case was centered on the robot, increasing the robot's stability, as it never fell over during testing.

The cargo holder was a rectangular prism that was 5.5 centimeters in width and height and 7.5 centimeters in length. There was a slit, approximately half a centimeter in height, cut into a cross section of the length of the cargo, about one centimeter from the edge. A plate was slid into this hole that acted as the face for this side of the rectangular prism. This allowed the container to carry a lot of cargo and be relatively easy to access while ensuring the contents of the cargo stayed safe during transportation. As shown at the second PoC and the final demonstration, the cargo was successfully able to hold the bead cargo even when the container was disturbed.

The system to drop the cargo container at the end of the maze was a combination of a sweeping door and a ledge. At the back of the robot, there was a ledge that was made from pins and long straight Lego pieces. The container resting just on this ledge would fall off, so to ensure that the cargo stays on during transportation, a motor was attached to the top of the robot on the back of the cargo container. A small axle was attached to the motor, and a long rod was attached to the axis. This rod spun in a plane that was parallel to the back of the robot. This rod would keep the cargo on the cargo ledge until the robot exited the maze, when the rod would sweep up, releasing the cargo. This was proven to be successful at the final demonstration as the robot was able to deliver the cargo at the end of the maze.

All the GEARS systems were designed to complete the tasks at hand and should be able to operate to their fullest capabilities. However, due to the lack of a feedback control system, one that would control the distance the robot is from the right wall, the GEARS had trouble with drifting to the left or the right and not turning at 90-degree turns. This caused the prototype to only achieve two out of the six tasks at the final demonstration, those being symbol identification and mapping. However, the team believes that the implementation of such software would allow the robot to utilize its design to the maximum and successfully complete all the tasks. Therefore, the prototype should be selected, as it has all the capabilities to complete all tasks; it just needs a simple feedback control algorithm to iron out any inconsistencies in driving or turning.

Sincerely,
Team 03

Executive Summary

Every year, there are many man-made and natural disasters that devastate the infrastructure, dislocate populations, and cause mass chaos. Providing aid to areas of the world that have suffered from such disasters is an immense challenge, as such areas can be scattered with debris and lack several resources, such as water and electricity. It is also very dangerous for first responders to attend to these areas. In response to such a problem, the team was tasked with the development of the global emergency autonomous response system (GEARS). GEARS is a robot that can navigate through the damaged terrain of an area affected by a natural disaster, carrying supplies and communicating with the local inhabitants.

The prototype developed has several unique aspects to its design. To begin, the prototype used a four-wheeled design that allowed for pinpoint and accurate turning. This was achieved by attaching two wheels of diameter 68.8 centimeters to the back of the robot. Their axles were powered by two large Lego spike motors. The front wheels were connected to the back wheels by a chain of gears. When turning, the inside wheels would turn backwards and the outside wheels would turn forward, allowing the robot to turn in place. The prototype also had a unique cargo drop-off design that combined a ledge and a sweeping door to deploy the cargo at the correct time and orientation.

At the final demonstration, the team's GEARS prototype did not perform to the standard that it was expected to perform, only completing two of the six tasks. However, given the current knowledge gained from the final demonstration, it is believed that updated software would solve all the problems that the robot experienced at the demonstration. The prototype successfully created a map of the maze, indicating hazards and safe areas. The symbols were also approved by the instructional team. The robot failed to navigate the maze properly due to the tendency to drift left or right and turn not exactly 90 degrees. This would cause the robot to crash into the wall. A simple feedback control mechanism that would turn the robot closer to the right wall when it is further away from the wall and turn it away from the wall when it is closer would eliminate the drifting issue and prevent the robot from smashing into the maze wall.

Design Considerations

With the rough terrain present in areas around the world struck by disaster and the need for supplies and resources, the GEARS needed to meet five main requirements. It needed to accurately move through the rumble and terrain, being able to navigate one hundred percent of the maze, reliably carry cargo full of useful resources, not dropping the container once, avoid dangerous areas, without hitting a single hazard, make a map of one hundred percent of the land and hazards to give first responders a better idea of the unfamiliar area, and based on survey results be well received by people of all cultures and backgrounds. Based on these objectives, Team 3 went through multiple iterations of both the hardware and software aspects of the GEARS. There were three main components of the robot that made up the hardware of the robot: the chassis and wheel system, the battery holder, and the cargo holder and dropping mechanism.

The first component designed was the chassis and wheel system, as it was the basis of the structure of the robot. When deciding on the design, three possible wheel options were determined. The first was a four-wheel drive where there are two motors, one connecting to the two front wheels and the other connecting to the front wheels. The second idea was a two-wheel drive in which two motors are used to power the two back wheels in a rear wheel drive. The last idea was the gear wheel system. This would utilize two centralized motors, one to power the left wheels and one to power the right wheels, and there would be a gear system connecting the two wheels on each side. With these ideas, a decision matrix, as seen in Figure 1, was created to determine the best idea based on what the team deemed most important for the GEARS: accuracy, simplicity, and power.

	Weight	Gear Wheel System	Two Wheel Drive (RWD)	Four Wheel Drive
Accuracy	15	13	10	11
Simplicity	10	5	7	5
Power	10	7	5	9
Score		31.5	27	30.5

Figure 1 - Decision Matrix for Wheel System

For accuracy, the team gave this factor the most weight because the GEARS most important goal is navigating unknown terrain that could be dangerous or rough. Therefore, having accurate movement, especially with making tight turns, would be critical in its completion of this goal. The gear wheel system would be best suited to complete these zero points turns and maneuvering these areas. With the setup of the motors and gears, it will allow the wheels on the inside of the turn to spin backward and the wheels on the outside of the turn to spin forward, effectively reducing the turning radius of the robot to zero. The two-wheel drive and four-wheel drive, on the other hand, would have a harder time turning with the front and back motors on one side not being able to turn concurrently. For simplicity, the two-wheel drive was the most generic design, making it the easiest to implement. Lastly, in relation to power, four-wheel drive would have the most power because of the way that the motors interact with the wheels, with no gears reducing the speed. However, the team opted for the gear wheel system, as seen in figure 2, because of its accuracy, despite its complex design and it not being the most powerful option. A slightly more complex and reliable idea was chosen rather than a simpler and powerful one, due to the accuracy needed in these dire situations.

Figure 2 – Initial Design of Wheel System

Overall, the wheel system worked well at what it needed to accomplish. It accurately made tight turns and carried the cargo reliably from one spot to another. However, the GEARS had to go through multiple iterations with the motor placement and structure of chassis to accompany the wheel system chosen. For the motors, in figure 2 it can be seen that they were originally placed on the side of the Raspberry Pi case, but this caused issues with our wheels splaying, especially with the extra weight of the battery. However, after some testing, it was determined that moving the motors to be underneath the Pi case would greatly increase the structural integrity of the robot. This did help, but there were still issues with the weight, especially with the addition of the cargo mechanisms. Therefore, there was a second big change incorporated to the chassis. Multiple structural pieces were put in place to effectively reduce the degrees of motion of the motors and line of gears to zero. Originally, the motors were not connected to each other, and the lines of gears were only connected to their respective motors. This was the biggest cause of the GEARS issue with splaying. Therefore, after failing the second Presentation of Competency (PoC) because of the GEARS falling apart, as many structural beams as possible were added to make each component of the chassis as connected as possible. The two motors were connected as multiple points, the two lines of gears were connected around the perimeter, and both the lines of gears were also connected to the Pi case. Overall, after these additions were implemented, the structure of the GEARS was improved dramatically. There was a great improvement from the second PoC, where the chassis broke every time, the GEARS attempted a turn. However, there was still one incident during the final demonstration where the chassis broke and one time where one of the gears popped off, which is a negative of the GEARS chassis design and can be improved with more reinforcement of the structure and better materials to hold the gears on their axles.

The next major component of the GEARS system is the battery holder. This was also another major issue of the GEARS splaying issue. Because of that, there were many iterations of the battery holder design. Initially it was placed at the back of the robot with square brackets. However, it caused the robot to be back heavy impeding the proper maneuvering of our robot. In addition, it left no room for our cargo holder and dropping mechanism. Therefore, the battery holder was moved to the top of the Pi case allowing the main structure of the battery case to hold most of the weight of the robot. However, it was offset and hanging off the back to leave the ports of the Raspberry Pi open, but this still caused issues with the GEARS being back heavy. Based on these issues, the battery holder was relocated to its final position as seen in figure 3 to be centered on the top. It made the Raspberry Pi's ports harder to access but made the robot much more stable and have an easier time maneuvering around.

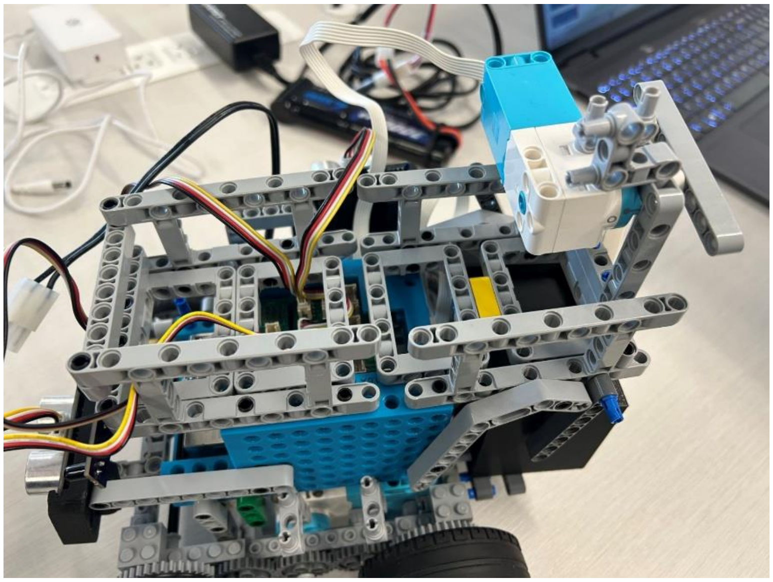

Figure 3 – Final Placement of the Battery Holder

The final component of the GEARS is the cargo holder and dropping system. The cargo holder was 3D printed using Fusion 360. The team decided to prioritize securing the cargo safely, ease in opening the container, and durability. Therefore, the design was a basic sliding door mechanism, with an arrow, shown in figure 4, etched on the box to indicate the direction to open the container. There is a slit on one side of the box, as seen in figure 4, which tightly holds a door that can be slid out. Because of the tight fit, the door cannot be slid out just using gravity, allowing the cargo to be secured in the box at any orientation. This was tested in the second PoC, and the container passed also during the final demonstration. In addition, because the container is in the shape of a box and made of light, durable plastic, the structure of the container is reliable and will not be an issue during transportation. Also, due to its shape, once it lands on the ground, it will stay put and not roll away if on an incline, making sure the cargo is precisely delivered to the correct area. However, one area of concern of the cargo container is the material used and its effect on the environment. Because it is made of plastic, the container, if not properly disposed of after drop-off, would then start breaking down, forming pollutants and microplastics. This is a big issue with the cargo container, especially if it is traversing regions with low population density. Therefore, an area of improvement for the cargo container would be to change the material used to a biodegradable plastic, or other material that can safely decompose into the soil.

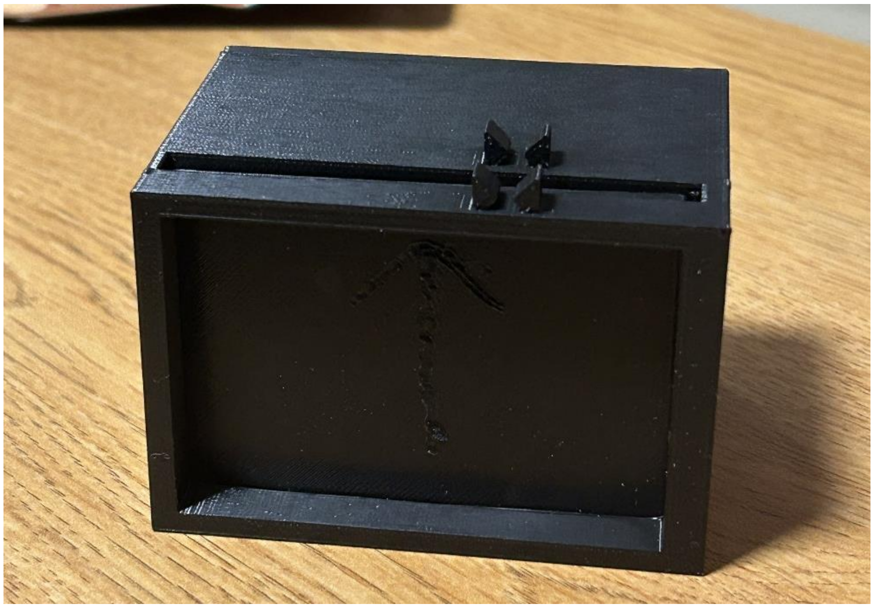

Figure 4 – 3D Printed Cargo Container

Moving on to the dropping mechanism, the first iteration was placing a rod in the back connected to a motor that the cargo container would sit on and would act like a trap door. After testing it though, it was scraped due to the stability issues of the cargo container on the rod and issues with the rod being an effective trap door. Therefore, the second and final design, pictured in figure 5, was created. The cargo container would sit on a ledge with a couple long beams, the white Lego pieces in figure 5, hanging down holding it in place, and the motor was moved to the top of the battery holder and connected to the beam that held the cargo container. When the GEARS was ready to drop, the motor would swing the arm, and the cargo container would fall off the ledge landing its target area. With this design, the ledge size had to be varied during testing to find the optimal length to both keep the container stable during maneuvering while also being short enough for the cargo container to fall off when the stabilizing arm swung away. Once the optimal ledge length was determined, the cargo dropping mechanism was very reliable, being able to drop consistently every single time.

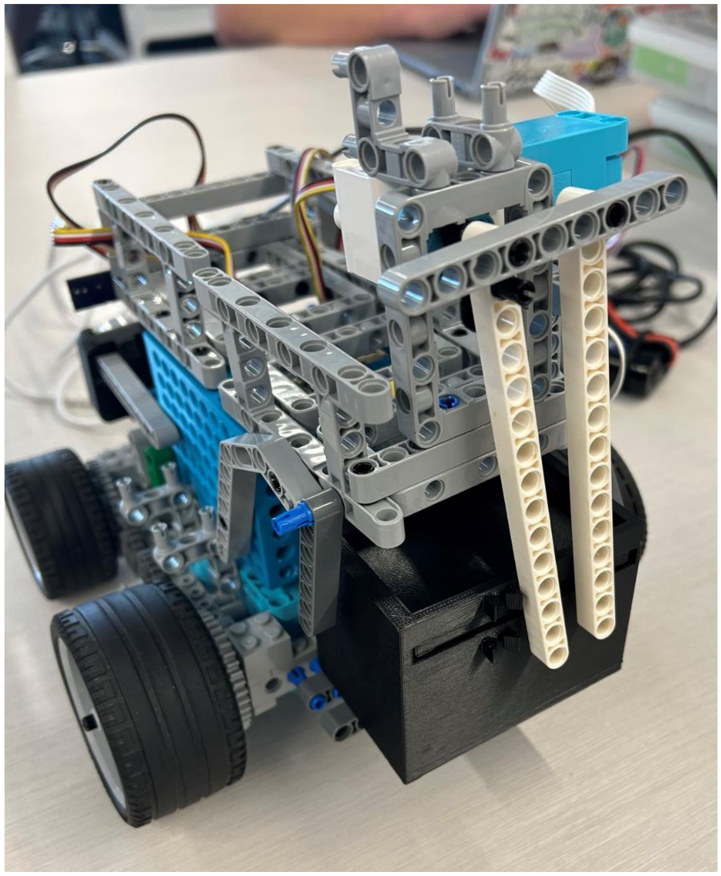

Figure 5 – Cargo Dropping Mechanism

Overall, the design of the hardware of the GEARS went through many iterations, leading to the most optimal final design that was able to carry cargo and traverse unfamiliar terrain, while being as nonthreatening and ethical as possible. The final design in figure 6 includes the three main components of the chassis, battery holder, and cargo mechanisms, as well as sensors placed around the robot enabling it to execute successful runs and complete the given objectives. Hokwever, future improvements to strengthen the structure of the chassis and change the material of the cargo container would further the GEARS' success.

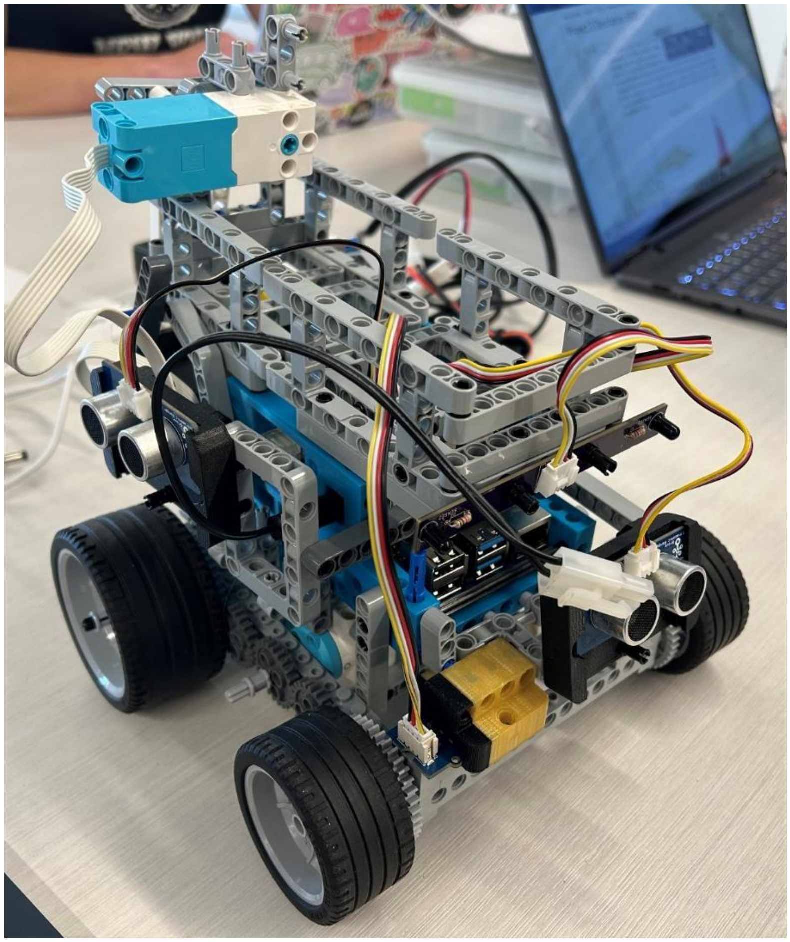

Figure 5 – Final Design of GEARS

The software was designed to be modular, both for ease of testing and implementation of the code into future iterations of the robot, where design may change its functionality. The main functionality of the software was built around simple motion and turning, with the flowchart pictured in figure 6. If GEARS sensed an obstacle farther than 10 centimeters away from the front sensor it would move forward at 50% power. If it sensed an obstacle less than 10 centimeters away from the front sensor, but greater than 10 centimeters away from the right sensor, it would turn towards the right. Otherwise, it would turn to the left. GEARS was only programmed to move forward, then make 90 degree turns in terms of motion, until it exited the maze. To incorporate hazard recognition, minimum danger values were calibrated. If the sensors read these minimum danger values, the value of the front sensor was set to less than 10 centimeters. Therefore, GEARS would think there was a wall in front of it and turn. If GEARS did not sense a wall up to 75 centimeters away from all sensors, it was assumed that it exited the maze. This assumption was initially based on sample mazes from the presentation of competency, but it was confirmed during the calibration process at the demonstration. This was a necessary assumption to make the exit a variable, so GEARS could perform adequately in any new environment, without needing to be hard-coded. Once GEARS exited the maze, a third motor was turned to deposit the cargo. Then, GEARS would spin three times to indicate that it had dropped the cargo, and that the cargo contained something safe. This spinning is meant to be interpreted as a dance, and make GEARS appear more cute and friendly to all cultures in any place affected by a natural disaster that GEARS may be supporting.

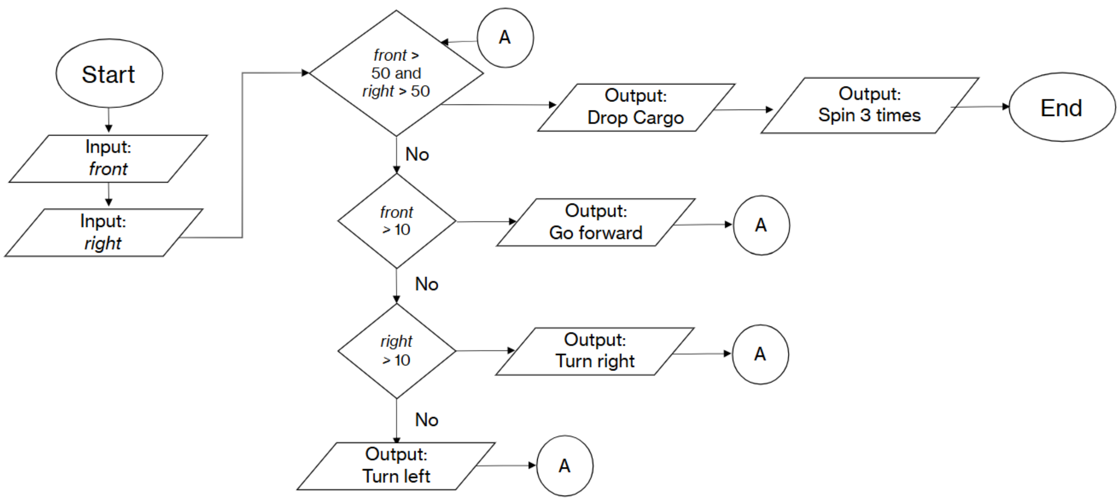

Figure 6 – Flowchart for the Motion of the GEARS

For the mapping code, as seen in figure 7, an 18 by 18 2-D list was created and initialized to be zeros, indicating places that GEARS had not traveled. It was assumed that this site would be enough to accurately detail the entire maze based off of the size of the room in the demonstration. This assumption was necessary because it was not known how large the maze would be. However, the list was built in such a way that the size could be changed if needed. This is very important for the future scaling of GEARS. The only inputs to GEARS were x and y values representing the start location of the maze. These values were used to traverse the list and insert 1's to represent the path taken by GEARS. Each index in the matrix represented a distance of 20 inches. Then, using the python time module, the map was set to be updated every two seconds. This is because GEARS took exactly two seconds to travel 20 inches. A list of possible orientations was created, which corresponded to negative or positive ones or zeros to increment x and y values by. The heading of GEARS was set to 0, representing North. Therefore, after two seconds had passed and it had not turned, the y value would increase by 1, and the x value would not change. Then, that value in the map would be changed to a 1. If the GEARS turned right—as recorded by the distance sensors--the heading would increment by 1. For example, if it starts facing north, then turns east, after two seconds the x value would increase by 1 and the y value would not change. If the robot made two turns to the right, the heading would increase by 2 remainder 4, ensuring that it did not go out of range of the headings list. Two turns to the right corresponds to a 180 degree turn, the heading would be set to exactly opposite what it was prior. In this way, the motion of the robot would be tracked.

A key assumption for the creation of this maze was that the time it took for GEARS to turn was the same it took to move forward. This assumption simplified the mapping code and was based off of the turning time for GEARS. While GEARS was turning, it would move forward in the direction of the turn and speed up slightly out of the turn. It was recorded to take just 0.1 more seconds to traverse a distance than forward motion, which was considered to be negligible in the mapping code. If a hazard was sensed, it would be recorded at the current x and y value. Whenever a value of 1 was added to the map, it was confirmed that it was then not replacing a hazard value. When the robot was recorded to have exited, a 5 was added to the map at that location, and the map was written to a text file to save the data. The Turtle module was used to draw the map, with symbols to represent possible hazards and the path GEARS took.

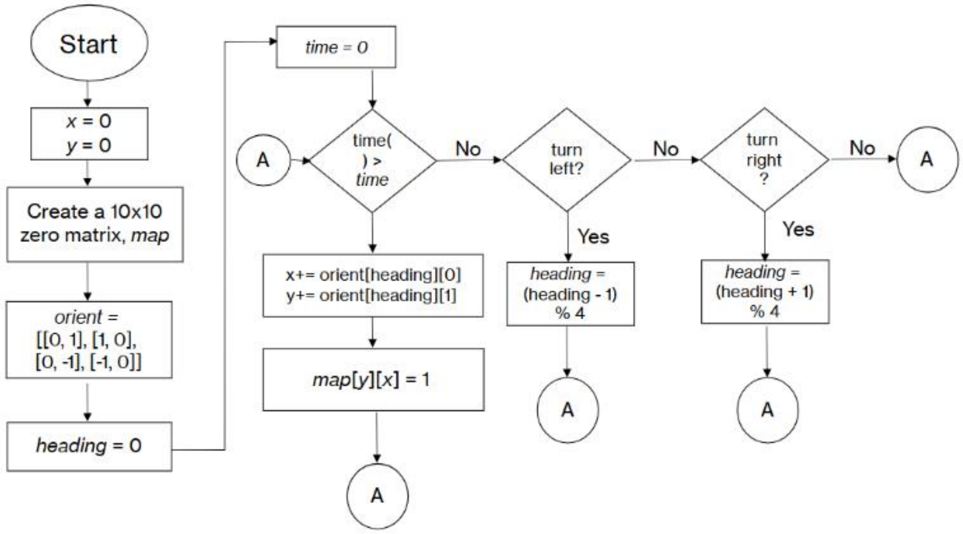

Figure 7 – Flowchart for the Mapping

Data Analysis

Gears used a total of 4 sensors with 3 separate roles in simultaneous location and mapping (SLAM). Two ultrasonic sensors were used for maze navigation. These sensors measure distance, so they are useful for determining how close to obstacles (maze walls) GEARS was, or if it had exited the maze by confirming all walls were too far away. These ultrasonic sensors were attached to the front and right side of GEARS. The second sensor utilized was the inertial measurement unit, or the IMU. The IMU contains a gyroscope, accelerometer, and magnetometer, but only magnetometer values were read for the code. The magnetometer can produce three-dimensional information about the location of magnets. However, since team 03 prioritized avoiding magnets in front of the GEARS system, only the y value was considered, based on the orientation of the IMU. The IMU is used to detect any possible magnetic activity in the surrounding area that could cause danger, such as from damaged power stations. Lastly, an infrared sensor was attached to the front of GEARS to detect the emission of infrared light, representing possible heat signals from fires in damaged areas.

Actual Distance (cm)	Ultrasonic Sensor Distance (cm)
2	1.87
4	3.26
6	5.98
8	7.49
10	9.74
12	12.08
14	13.92
16	15.56
18	17.53
20	19.96

Figure 8 – Front Ultrasonic Sensor Accuracy

Actual Distance (cm)	Ultrasonic Sensor Distance (cm)
2	2.12
4	4.09
6	5.67
8	7.23
10	10.21
12	11.32
14	14.23
16	16.12
18	18.23
20	19.78

Figure 9 – Right Ultrasonic Sensor Accuracy

The front and right ultrasonic sensors were reading distance from a wall in 2 cm increments and comparing it to the measured distance from a wall. It was found that both the ultrasonic sensors were accurate to half a centimeter, which the front sensor tending to underestimate while the right sensor would overestimate.

Readings from the ultrasonic sensor were taken every 0.1 seconds to every accuracy of the robot without overwhelming the sensors with data. The ultrasonic sensor was then programmed to make the motors move forward until it reached a distance of 10 centimeters from a wall. This number was determined based on the fact that the typical distance between two walls was 20.0 inches. Based on the length of GEARS (7 inches) and a turning radius of 1 inch, it was confirmed that stopping at 10 centimeters would provide ample space for the robot to turn without hitting any walls. Making GEARS turn at the halfway point to a wall creates a better allowance for possible drifting and over or under turning. If the front ultrasonic sensor detected a wall less than 10 centimeters away, control of the motors would switch to the right ultrasonic sensor. If this sensor did not detect a wall, it would power the motors to turn right 90 degrees; otherwise, it would power them to turn left. The robot was not programmed to go backwards, as during testing this would cause the robot to go off course. Instead, it would continue turning until the path was clear, which also ensured more accurate hazard detection. One flaw with this system, and an improvement for the future, is that it would be possible for GEARS to get stuck turning in an infinite square. To change this, proportional feedback control could be implemented so that a robot should follow a left wall and always try to maintain a certain distance from that wall. This would require the location of the right ultrasonic sensor to change.

Then, the magnetic sensor was calibrated. First, tests were performed to confirm the IMU's sensitivity to direction and distance. A magnet was placed 2 cm away from the IMU, then incrementally moved 2 cm away. The magnitude of the magnet was recorded by the IMU (Figure 9). Magnetometer readings significantly decreased more than 6 centimeters away from a magnetic source, dropping from 1000 to less than 300. The orientation of the IMU compared to the magnet at a distance of 20 centimeters was also tested and recorded, with angle increments of 30 degrees (Figure 10). To provide enough space for GEARS to turn away from the magnetic source and not become damaged by it, a value of 750 uT was set to confirm an area of high magnetic sensitivity. This value was not adjusted during the calibration process of the demo as it was determined that sensing a magnet through a wall was unrealistic compared to when the robot would actually enter the maze. The graph of the data is plotted in figure 10.

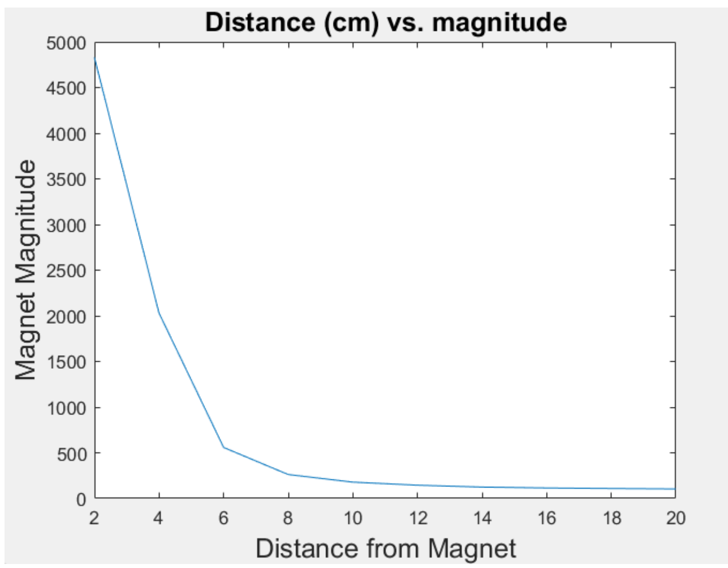

Figure 10 – Graph of Distance (cm) vs. Magnitude of Magnetic Field

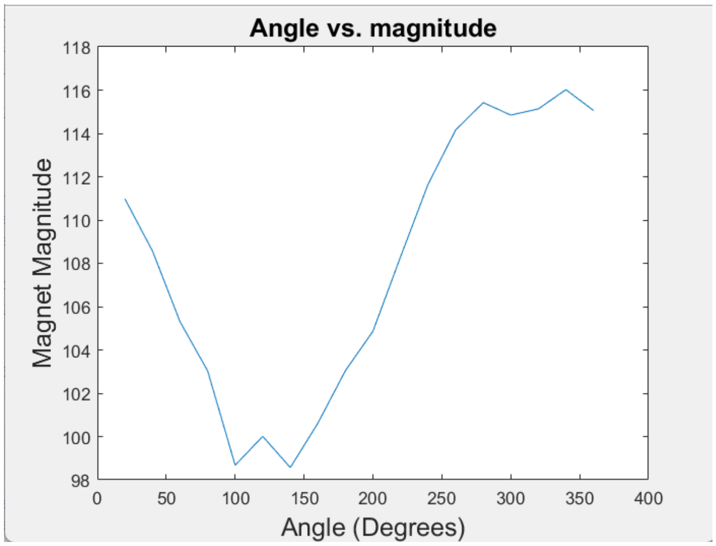

Figure 11 - Graph of Angle (Degrees) vs. Magnitude of Magnetic Field

The IR sensor was initially calibrated to the IR light emitted by humans, as no infrared source was available at the time. For this reason, its lower limit on the dangerous amount of infrared light emitted was initially 1 watt. This created problems for the robot's movement, as the sensor was sensitive enough to read 1 watt even with no IR source. When the IR source to be used for the demo was obtained, it was found that the sensor read just 250 watts from a distance of 6 centimeters away to be kept consistent with the magnetometer (Figure 11). Therefore, it was incorporated into the software that if GEARS sensed an IR source of more than 250 watts, it would mark it on the map and turn away. This value was not adjusted during the calibration process of the demo as it was determined that sensing an IR source through a wall was unrealistic compared to when the robot would actually enter the maze.

Distance (cm)	IR Reading (Watts)
2	387
4	335
6	252
8	187
10	172
12	168
14	159
16	156
18	147
20	143

Figure 12 – Right Ultrasonic Sensor Accuracy

Since the robot's movements are based off of ultrasonic sensors, if a magnetic or infrared source was detected in front of the robot (see lower limit values above), the front ultrasonic sensor value was set to less than 10 centimeters. This ensured that it would make a 90-degree turn away from the hazard. The direction of the turn was then dependent on if the right sensor had a clear path, the same as if the front ultrasonic sensor was registering a wall, with the calibration information plotted in figure 11. If a magnetic hazard was sensed, a 2 was inserted onto that map at the current read x and y values of the robot, and a 3 was used to represent an IR, or extreme heat source. Information about the source was written in a text file to provide accurate data to any first responders about potential danger in the area. GEARS was programmed in such a way so that if a location had both an IR hazard and a magnetic hazard in the same location, it could identify both and navigate around them. Based on saved x and y coordinates, it was also ensured that if GEARS encountered the same obstacle, it would not write it to the file twice.

Cultural and Ethical Considerations

When designing the cargo labels for GEARS, multiple factors had to be kept in mind. Since GEARS was to be applied to many cultures across the world, the cargo labels needed to effectively show the cargo inside using a universal, easily identifiable label while also being non-threatening. Eventually, four labels were decided on for food, medical supplies, shelter, and fuel. These were based off external research as well as a survey that was sent to over 100 people where they were asked to correctly identify the symbol out of the choices of food, medical supplies, shelter, and fuel. Participants were then asked if they had no contact with the modern world, would they have been able to identify said symbol? One issue when creating these symbols was avoiding cultural bias. For example, a thumbs up seems innocuous in Western countries when it has a positive connotation, but in the Middle East, it is seen as an offensive gesture (Anderson, 2019).

For food, a symbol of an apple and a water bottle was chosen (Figure 13). 100% of participants correctly identified the symbol, and 94% stated that if they had no contact with the outside world, they would be able to still identify it. Apples were chosen because they are the third most cultivated fruit worldwide and are present in all continents except Antarctica (Apple Production by Country, 2025). There is a very high chance that whatever civilization GEARS was sent to would be able to identify this as food, and even if they had never seen an apple before, it has many similar relatives present in multiple environments as well. The water bottle in the symbol was chosen due to water being a universal need and the bottle being clear, making the liquid inside clearly identifiable as water. Concerns were raised about civilizations not in contact with the modern world identifying the bottle as water; however, 94% of survey respondents said they would still be able to identify it. Additionally, the contextualization of the bottle being compiled with the apple likely would allow them to identify it as water.

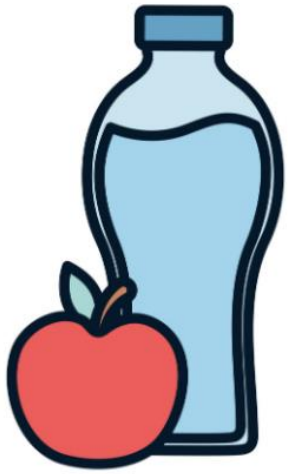

Figure 13- Food and Water Label

The symbol for shelter chosen was a picture of a tent in a rural setting (Figure 14). This was chosen because although it is not applicable to modern societies, all of them would be able to identify it, and the civilizations not in contact with the modern world likely still use similarly structured shelters. To further support this idea, 100% of survey respondents correctly identified the symbol and stated that they would be able to identify it with no modern contact.

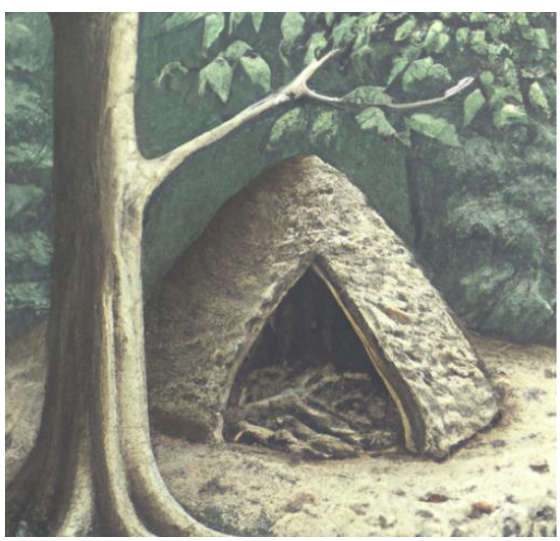

Figure 14- Shelter Label

For medical supplies, the symbol decided on included a pill, pill bottle, syringe, and medical test tube (Figure 15). 100% of survey participants correctly identified the symbol, but only 25% said that they would still be able to do so with no contact with the outside world. However, it is likely a civilization with no modern contact would not even know how to correctly use these supplies, so this was not seen as a major issue. Additionally, there was a high chance that these civilizations are not as likely as others to receive medical supplies due to the fact that they have no immunity, and things like vaccines and antibiotics could in fact negatively impact the community (Nuwer, 2022).

Figure 15- Medical Supplies Label

For the final symbol of fuel, a green cannister with a lightning bolt and a leaf along with a red cannister with fire were chosen (Figure 16). The green canister represented renewable or clean energy, while the red represented more traditional fuel types such as fossil fuels. Together, these conveyed the message of fuel that could be used for almost anything. 100% of respondents correctly identified this symbol, but only 19% said they would be able to identify it with no contact with the modern world. However, if a community with no outside contact was to receive fuel, there is a high chance they would not know what to do with it due to the fact that they only use natural sources such as wood and plant materials.

Figure 16- Fuel Label

GEARS was also designed with the goal of having a non-threatening appearance in mind as well. Natural colors such as soft shades of purple, blue, and green are often seen to be the friendliest and calming. Additionally, the structural design of GEARS aimed to use sleek and smooth lines and designs. Robots that try to mimic human design but do not quite reach that accuracy suffer from the “uncanny valley” effect, which evokes fear and aggressiveness in other humans (Perez et all, 2020). To combat this, GEARS was designed to mimic a more pet-like design.

Different cultures react differently to aid as well. In 2004, North Sentinel Island, which is inhabited by the Sentinelese, a tribe in voluntary isolation, was struck by the Indian Ocean earthquake and suffered negative aftereffects (Tsunami leaves Tribal Island High in the water, 2005). However, when an Indian government helicopter went to observe how the islanders were adapting, they shot arrows and threw spears at the helicopter. When interacting with often-hostile civilizations in isolation, being culturally aware is important because there are so many factors that could go wrong when attempting to communicate.

Another example of this was again seen in the 1971 Iraq poison grain disaster. Mercury coated wheat grain was sent to Iraq meant for farmers to plant, but many of them instead used it for individual consumption. (Chepkemoi, 2017). These bags of grain were labeled with “DANGEROUS” and a skull and crossbones design, but many of these farmers did not speak English and the design meant nothing to them. As a result, mercury was ingested by many communities with 6,530 patients admitted to hospitals and 459 deaths reported.

Results and Discussion

The GEARS prototype was required to navigate a maze with infrared and magnetic hazards, deliver cargo at the end, and create a map of the maze. The team expected the prototype to navigate the maze using ultrasonic sensors and pinpoint turning to deliver the cargo successfully at the end of the maze. A map was to be generated that was to represent which areas of the maze were navigable versus those that were not.

Going into the demonstrations, GEARS was expected to be able to navigate the maze, deliver the cargo, and generate a map of the maze. There was a lower level of confidence as to whether GEARS would be able to successfully navigate the infrared and magnetic sensors. This was due to a lack of testing of these sensors and the software created to navigate such hazards.

At the beginning of the maze, the robot was successfully following walls and turning, following the path of the maze. However, the GEARS prototype kept running into the walls by drifting into them when going straight, or the prototype would be turning too sharply or not enough and drive into a wall. Due to this, the team decided to move the robot to the final checkpoint and demonstrate GEARS' ability to navigate the maze and drop off the cargo. The team decided to move to this checkpoint as there were fewer turns that the prototype must complete. While the cargo deployed successfully when the robot exited the maze, the same problems with drifting and imprecise turning persisted. The team was aware of such issues before the demonstration, but the compounding effect of these imprecisions was not considered throughout the whole maze. The hardware design for GEARS is believed to be sufficient for navigation of debris. The current design of an ultrasonic sensor on the front and right of the robot is enough to navigate walls and debris in a maze setting. The reason the prototype failed was due to its lack of a feedback control mechanism to regulate the distance that the robot is from the wall. Such feedback control would allow the robot to stay a reasonable distance from any debris by moving slightly towards the wall when it is further away and away from the wall when it is close. This would eliminate the chance of GEARS crashing into debris due to a slight drift in its movement, as it would regulate the distance it is from the wall using the ultrasonic sensor on its right side.

The infrared and magnetic sensing tasks were not attempted at the demonstration. Since the robot was having trouble navigating many walls and turns, the team believed that it would be best to spend the most time on the first and last tasks, as they required the least number of turns. This decision also considered that the red infrared and magnetic hazard code was not as refined as other aspects. It is expected that should the tasks have been attempted, GEARS would have been able to identify the hazards but not able to navigate the maze due to its imprecise movement and lack of feedback control.

The robot successfully constructed a map, pictured in figure 13, to resemble what areas of the maze were open and therefore safe and where the walls and dangers were. This worked as expected. The mapping code utilized a two-dimensional array of ones and zeros to resemble the maze. Based on the sensor reading and robot orientation, the array would update, with zeros representing clear, safe areas and ones representing dangerous areas. High infrared and magnetic locations would also appear on the map, as indicated by the rainbow circle on figure 10.

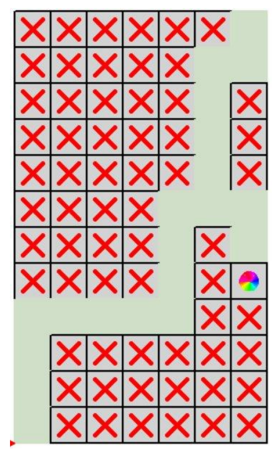

Figure 17 - Sample Map Output

Finally, the symbols used to represent the four types of supplies GEARS could be carrying passed the inspection of the instructional team. This was as expected due to a survey where approximately 88% of respondents were able to identify what each symbol represented.

Conclusions and Recommendations

In conclusion, the team 03 global emergency autonomous response system was incapable of completing all necessary tasks due to compounding errors during translational or rotational movement of the prototype. The robot had sufficient hardware that included four-wheel pinpoint turning and an infrared, inertial mass unit, and ultrasonic sensor on the front and right of the robot. These sensor placements would allow the robot to navigate walls and avoid hazards. Due to various testing beforehand, sensor values were determined as to what represents a safe and dangerous distance from magnetic and infrared sensors. Software was written to move the robot forward if it could and turn to the right if it could not move forward and turn to the left otherwise. If the robot detected a high source of infrared radiation or a strong magnetic source, it would avoid such locations.

The overarching problem with the team's design was that there was not a feedback control mechanism to keep the robot a certain distance away from the wall. Due to this lack of a feedback mechanism, the robot would drift to the left or the right while traversing the maze. This, compounded with the robots' imprecise turning, caused the robot to run into the wall even though it may have started off straight. A control algorithm to have the robot turn away from the wall when it is close and turn towards the wall when it is far away would have solved this problem.

The robot effectively was able to carry and drop off cargo due to its unique cargo storage and dropping mechanism of a half ledge and a sweeping door that would allow the cargo to fall out when the ultrasonic sensors detected that the robot was out of the maze. The cargo container itself was a spacious rectangle that allowed for plenty of volume to store cargo. One of the faces of a rectangle slid into the rectangular prism and remained due to friction. A symbol was used to represent the contents of the cargo. There were four symbols in total, each for the possible cargo. According to a survey of almost twenty people, the symbols were sufficient to communicate the contents of the cargo to people of any community or culture.

References

Apple Production by Country 2025. Apple production by country 2025. (n.d.). https://worldpopulationreview.com/country-rankings/apple-production-by-country

Chepkemoi, J. (2017, April 25). *What happened during the 1971 Iraq Poison Grain Disaster*. WorldAtlas. https://www.worldatlas.com/articles/what-happened-during-the-1971-iraq-poison-grain-disaster.html

David Anderson, M. S. (2019). *5 everyday hand gestures that can get you in serious trouble outside the US*. Business Insider. https://www.businessinsider.com/hand-gestures-offensive-different-countries-2018-6

Nuwer, R. (2022, February 24). *Anthropology: The sad truth about uncontacted tribes*. BBC News. https://www.bbc.com/future/article/20140804-sad-truth-of-uncontacted-tribes

Perez, J., Goo, H., Ramos, A., Contreras, V., & Strait, M. (2020). *he Uncanny Valley Manifests Even with Exposure to Robots*. ACM Digital Library. https://dl.acm.org/doi/10.1145/3371382.3378312

Tsunami leaves Tribal Island High in the water. ESA. (2005). https://www.esa.int/Applications/Observing_the_Earth/Proba-1/Tsunami_leaves_tribal_island_high_in_the_water