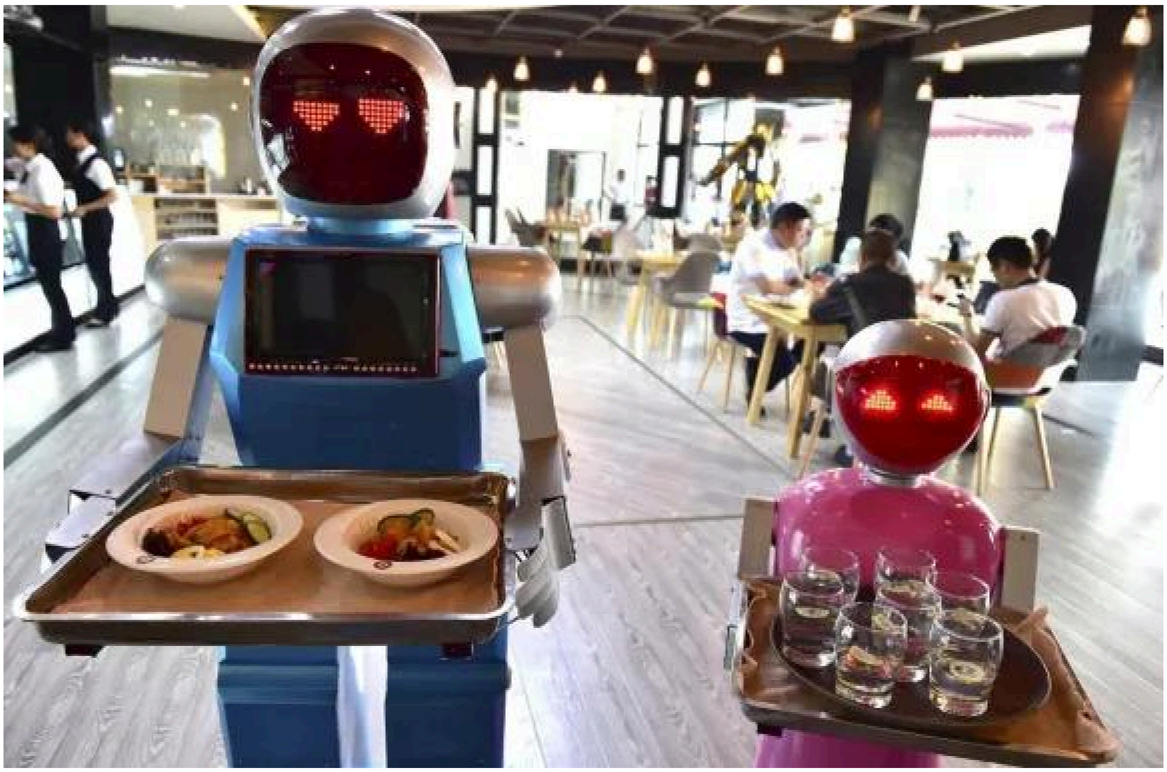
FoodieRover:
The Food Delivery Robot



Group 29 Members

Mauricio Ferrari - Computer Engineering

Alexis Gyselinck - Electrical Engineering

Antonia Soto - Electrical Engineering

Chidoziri Maghiro - Electrical Engineering

Committee Members

Aman Behal - Ph.D.

Chinwendu Enyioha - Ph.D.

Mike Borowczak - Ph.D.

Table of Contents

1. Executive Summary

Foodie Rover represents a compact autonomous ground vehicle meticulously engineered for the purpose of acting as a proficient food delivery robot. Configured with a diverse array of sensory peripherals, this vehicle is poised to accomplish autonomous navigation devoid of any external intervention. This project, being multifaceted, serves as an exemplary showcase of various aspects within the domains of electrical and computer engineering.

On the electrical front, Foodie Rover necessitates an extensive degree of circuit integration, considering the amalgamation of diverse components that encompass motors, sensors, a camera, a microcontroller (or microprocessor), and a GPS module. These intricate connections and interfaces are paramount to the system's operation, showcasing the profound importance of electrical engineering within this endeavor.

In parallel, the domain of computer engineering takes center stage as it orchestrates the orchestration of autonomous functionalities through the skillful programming of the aforementioned devices. The efficacy of Foodie Rover's autonomous operations is inherently linked to the programming and logical execution embedded within the system. Additionally, Foodie Rover will aim to feature user interactivity through a mobile application interface which is directly linked to the locking mechanism that is used to safeguard the food.

Noteworthy is the departure from conventional methodologies, as Foodie Rover adopts a unique approach by integrating Google Maps' API endpoints into its navigational system, thereby attaining advanced mapping and location-awareness capabilities. This incorporation of API endpoints demonstrates the integration of cutting-edge technology within the project's framework, marking a distinctive and innovative dimension of computer engineering.

The principal aim of our project is to present a compelling demonstration of Foodie Rover's autonomous driving capabilities, exemplified through the successful completion of a delivery task from one campus building to another. This mission mandates Foodie Rover to execute a series of intricate decisions pertaining to routing and navigation, bearing in mind the dynamically evolving sidewalk configurations inherent to the University of Central Florida's campus.

Additionally, our subsidiary objectives encompass the deployment of computer vision techniques to identify viable and navigable pathways while efficiently detecting and mitigating obstacles that may impede the rover's progress. Furthermore, Foodie Rover is charged with the task of maintaining a consistent walking pace throughout the journey and, upon reaching its destination, initiating communication with the customer to confirm the successful delivery of their food order. These sub-objectives collectively contribute to the overarching goal of showcasing Foodie Rover's prowess in autonomous delivery operations.

2. Project Description

This chapter aims to provide a clear and formal introduction to the background and motivation driving the development of FoodieRover. It will encompass detailed descriptions of both fundamental and advanced goals and objectives that underpin this project. Furthermore, we present specific engineering specifications that align with the specifications and goals introduced in their respective sections.

In the subsequent sections, we present hardware and software block diagrams in a comprehensive manner, offering an in-depth understanding of the technical aspects of the FoodieRover design. Additionally, a product planning matrix for quality function deployment will be introduced to ensure a structured and balanced approach to product development and quality assurance.

2.1 Project Motivation and Goals

The objective behind this project is to cut down on time wasted waiting to receive our orders from the busy restaurants on campus (e.g. Chick-fil-A). As full-time students, we experience a restrictive time window for lunch due to our tight schedules so, when we spend 20 out of the 30 minutes we have loafing around in await for our meal, it takes away from what is supposed to be a midday break where we can recharge and relax as we enjoy what we're consuming. Rather, it becomes a stressful contest of how fast can you finish an entire combo and still have time to walk to your next class. Additionally, it would offer students and faculty alike the chance to be more productive as they won't have to remove themselves from their workspace to acquire sustenance.

FoodieRover is a small self-navigating robot that doubles as a food convoy. Our grandest goal for this project is to design FoodieRover to be able to detect objects. Obstacle detection is no small feat, especially when dealing with active traffic (moving objects). Examples of active traffic on our campus could range from human feet, squirrels, bikes, scooters, skateboards, and even golf carts. The issue here is finding the right sensitivity for our obstacle detection program. We certainly don't want FoodieRover to come to a lasting halt each and every time a foot cuts in front of it. Ideally, FoodieRover should only stop if the object is close enough for impact and not moving out of the way anytime soon. Static objects could include large rocks, parked vehicles, or even conversing humans. Our plan is to attach an array of ultrasonic sensors onto the chassis of FoodieRover and when their feedback is high enough and consistent enough then we can determine that we should come to a stop.

Figuring out when a detected object actually presents a notable obstacle is only one piece of the puzzle though. We want FoodieRover to not only be able to pick up any obstructions in its path but also be capable enough to route around the said obstruction. This is mainly a software level challenge and we plan to address this by developing an algorithm in conjunction with API calls to Google's Maps services. Google Maps allows users to send coordinates of their starting and ending locations for a trip in order to return an acceptable route.

This service coupled with the GPS dedicated hardware (e.g. magnetometer) will enable FoodieRover to come up with new routes for itself on the fly. Which leads us into our next goal of this project, which is to showcase our ability to integrate varying electrical components together. For the processes already described above we would require a microcontroller with wifi capabilities (possibly a microprocessor instead if the code is too heavy). Other components we have to consider are the battery pack, motors for the tracks, locking mechanism, and keypad.

The locking mechanism is a feature we want to include within the food storage container so that the food won't be stolen. To couple this lock and key pad, we'd also like to create an interface for cellular devices that can communicate with FoodieRover (i.e. provide tracking information, notification when FoodieRover has arrived) for both the customer and restaurant to utilize. We're interested in implementing computer vision to enhance FoodieRover's obstacle detection capabilities.

Autonomous delivery robots are already present in the market, although they do not dominate the space. They are a slow, but growing, sphere of technology with one of the industry leaders being Kiwibot. Kiwibot also operates on college campuses and similarly provides interfaces in which users can interact with their product. Kiwibot features cameras, 3 frontal and one rear wide, along with LTE and GPS capabilities.

Kiwibots are significant as they exhibit level 4 autonomy, which is achieved when the vehicle has full self-driving capabilities. However this is within limited ODDs (operational design domains) so outside of these constraints it is acceptable for the vehicle to rely on human intervention. Another aspect of level 4 vehicles is that they do not need to be continuously monitored. They should be competent enough where the driver can have trust that it will arrive at its location safely without hiccups. Likewise, my group aspires to develop FoodieRover to have level 4 autonomy.

2.2 Technical Objectives

Our aim is for FoodieRover to be a self-navigating automated food delivery system that can receive location and destination information and use it to navigate a course to the destination without user interference, all while keeping the enclosed delivery safe from intruders and the elements.

As mentioned above, FoodieRover aspires to have obstacle-detection capabilities. In order to achieve this, proper sensory information about its environment is required. Therefore we are looking to provide FoodieRover with 8 ultrasonic sensors: 1 for each side (front/back/left/right) and then one for each shoulder. The idea here is that when the sensors are picking up strong feedback (indicating an obstacle), we can check the other ones to decide what the best route around the obstacle would be.

Other software specifications are the necessity of an application interface for users to interact with FoodieRover. The application will support two types of users: consumer and producer. The consumer interface needs to have options to put in orders, enter drop-off locations, and unlock the container. The producer side needs to be able to support commands to deliver food, return to base, and post food items.

After the application, the last key component of the software is its API capabilities. This would require FoodieRover to have a wifi-chip so that it can make the API calls. As of now, the two servers we're looking to contact is Google Maps and the eventual server of the application.

In terms of performance, FoodieRover will be expected to travel distances over 200 meters per trip at a speed of 1.5 to 2 miles per hour. It will carry around 3 pounds of food kept fresh by insulating material. While delivering food the power consumption will be kept under 400 Watts. When not moving, the bot ideally will reduce its power consumption to under 50 Watts. The battery capacity will correspond to 1 to 2 hours of runtime with a battery that can be replaced and recharged. FoodieRover will adapt to the different conditions it will encounter while delivering food. Its hardware design will accommodate for the non-ideal scenarios it may encounter.

Furthermore, its dimensions will exceed or be equal to a cubic foot to provide enough space for a meal with a drink. The interior design of the container will prevent spillage. The electrical components will be kept safe and in place with custom frames.

2.3 Marketing and Engineering Requirements:

2.3A. Introduction:

This section explores the creativity of presenting a cutting-edge food robot that not only draws the imagination but also carefully complies with an amazing set of specifications.

FoodieRover is designed to travel distances exceeding 200 meters at a graceful speed of 1.5 to 2 miles per hour. Imagine it delicately carrying an amazing delight weighing up to 3 pounds, all while preserving its freshness through the mastery of insulating materials. As it gracefully navigates the landscape, its power consumption remains a testament to efficiency – capped at 400 Watts during delivery and a mere 50 Watts when at rest, conserving energy without compromise.

FoodieRover is more than a mechanical marvel; it's a culinary chameleon, adapting seamlessly to the diverse conditions encountered while delivering food. Its hardware design anticipates and accommodates non-ideal scenarios, ensuring reliability in every culinary mission it undertakes.

The dimensions of FoodieRover extend or match a cubic foot, providing ample space for a delectable meal accompanied by a refreshing drink. Inside, spillage is a thing of the past, thanks to a carefully crafted interior design. The electrical components, the heartbeat of this gastronomic companion, are safeguarded and organized with precision, nestled within custom frames.

2.3B Tables:

Engineering requirements:

Robot Dimensions	>=1 cubic foot
Movement Speed	1.5 - 2 MPH
Travel Distance	200 meters per trip
Weight of food robot can carry	~ 3 pounds
Battery Life	1-2 Hours
Circular sensing area	360 degrees
Radius of sensing area	2 meters
Idle Power consumption when on Go	400 Watts
Idle Power consumption when resting	50 Watts

Figure 2.1

Marketing Requirements:

Marketing requirements:

1. **This Robot should move on its own**
2. **The robot should be able to move at fast enough speed**
3. **The robot should be able to travel good distances**
4. **The robot should be able to carry the weight**
5. **The robot should have something to keep food fresh**
6. **The robot should have good battery life**
7. **The robot can't be too small**
8. **The robot should be able handle stuff getting in its way**
9. **The robot should be able to keep the food safe**

Figure 2.2

Table explaining how the marketing requirement meets the engineering requirement:

Robot Dimensions	>= 1 cubic foot
Speed	>=1 mph
Travel Distance	200 meters per trip
Weight capacity of food	<= 3 lbs
Battery Life	1-2 Hours
Circular sensing area	360 degrees
Radius of sensing area	>= 1 meter
Computer Vision	1 camera
Power consumption when on Go	400 W
Idle power consumption when resting	50 W
API Connection Delay	<= 5 sec
Locking/Unlocking via Keypad	95% accuracy

Figure 2.3

2.4 Hardware and Software Diagrams

The diagrams comprise both the hardware and software designs separately. These diagrams are intended to offer a comprehensive and high-level overview of the system, encompassing both the hardware and software key components and the interactions of the many components within the system.

Hardware Block Diagram

Following a thorough evaluation of the essential hardware components required to meet the specified objectives and specifications, a comprehensive block diagram has been meticulously crafted. This diagram intricately interconnects the key components in a coherent manner, facilitating a clear understanding of their interactions within the system.

In the diagram, arrows directed toward a block signify inputs, while arrows extending from a block denote outputs. Detailed descriptions of each component, along with their current statuses as of the provided date, can be found in the accompanying legend table. Furthermore, for clear attribution of responsibility for each major component, distinct color-coding has been employed to designate the respective team members accountable for their development and maintenance.

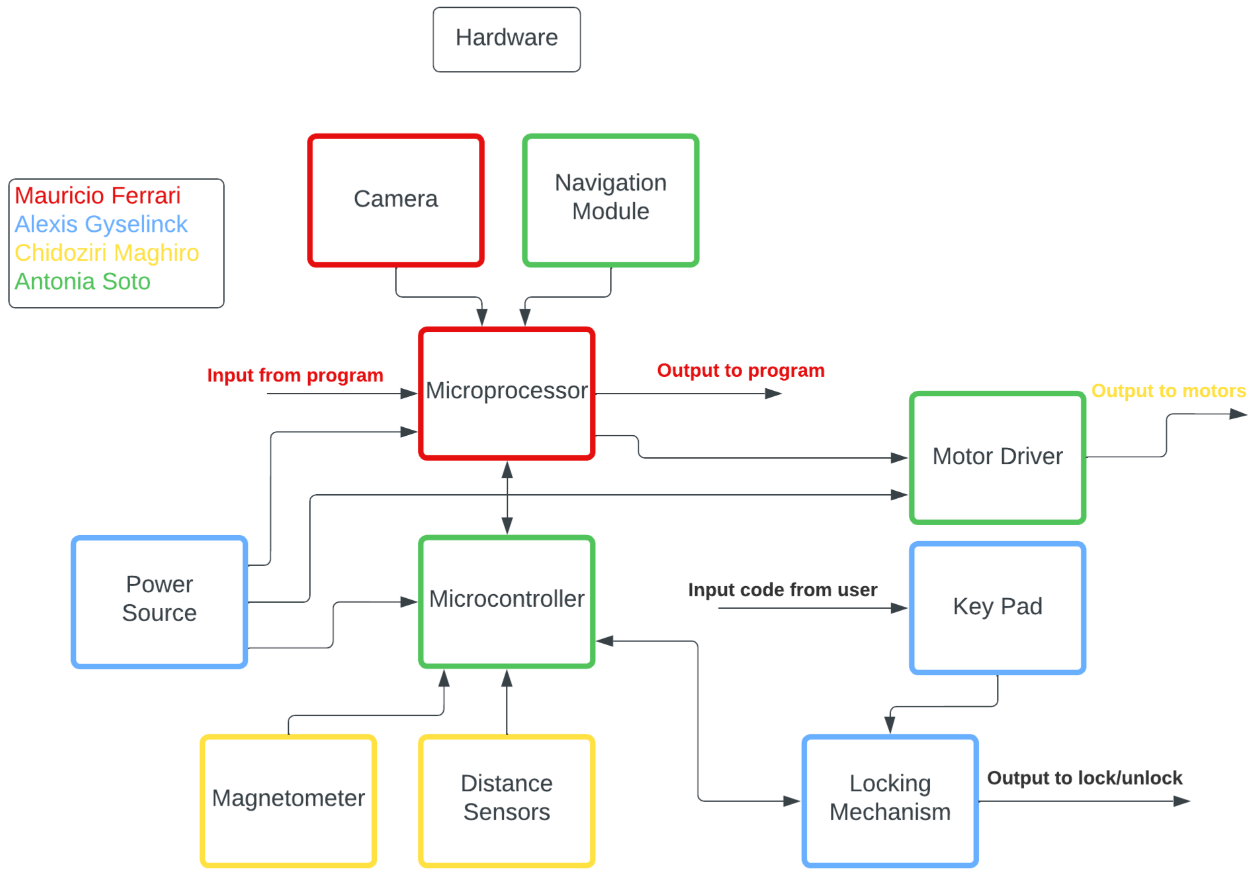

Hardware
Mauricio Ferrari
Alexis Gyselinck
Chidoziri Maghiro
Antonia Soto
Camera
Navigation Module
Input from program
Microprocessor
Output to program
Output to motors
Motor Driver
Power Source
Microcontroller
Input code from user
Key Pad
Magnetometer
Distance Sensors
Locking Mechanism
Output to lock/unlock

Figure 2.4

Diagram Legend		
BLOCK NAME	BLOCK DESCRIPTION	BLOCK STATUS 4/5/24
Microprocessor	Computer processor on a microchip. Functions as the communication module of the system.	Integrated
Microcontroller	Integrated circuit design for specific control tasks.	Integrated
Navigation Module	Satellite-based location receiver.	Integrated
Keypad	Small set of keys for user input.	Integrated
Locking Mechanism	Controls access authorization and provides security.	Integrated
Magnetometer	Instrument that measures magnetic forces; works as a compass.	Integrated
Motor Driver	Current amplifiers that bridge the controller to the motors.	Integrated
Power Source	Provides electrical energy to the circuit.	Integrated
Distance Sensors	Sensor that measures distance using ultrasonic sound waves.	Integrated

Figure 2.5

Software Flowchart

Software flowcharts play a pivotal role in the development and understanding of software systems, serving as visual representations that illustrate the logical flow of a program's execution. These diagrams provide a high-level, easy-to-grasp overview of the entire software structure, outlining the sequence of operations, decision points, and interactions between various components. The importance of software flowcharts lies in their ability to enhance communication and collaboration among developers, designers, and stakeholders involved in the software development process. By offering a visual roadmap, flowcharts facilitate clear comprehension of the program's logic, aiding in error identification, troubleshooting, and refining the software design. They serve as valuable documentation tools, assisting both in the initial development phase and subsequent maintenance or modifications, ensuring that the software remains comprehensible and maintainable over time. Ultimately, software flowcharts contribute to the creation of robust, efficient, and well-structured software systems, aligning the development team and stakeholders with a shared understanding of the software's functionality and design.

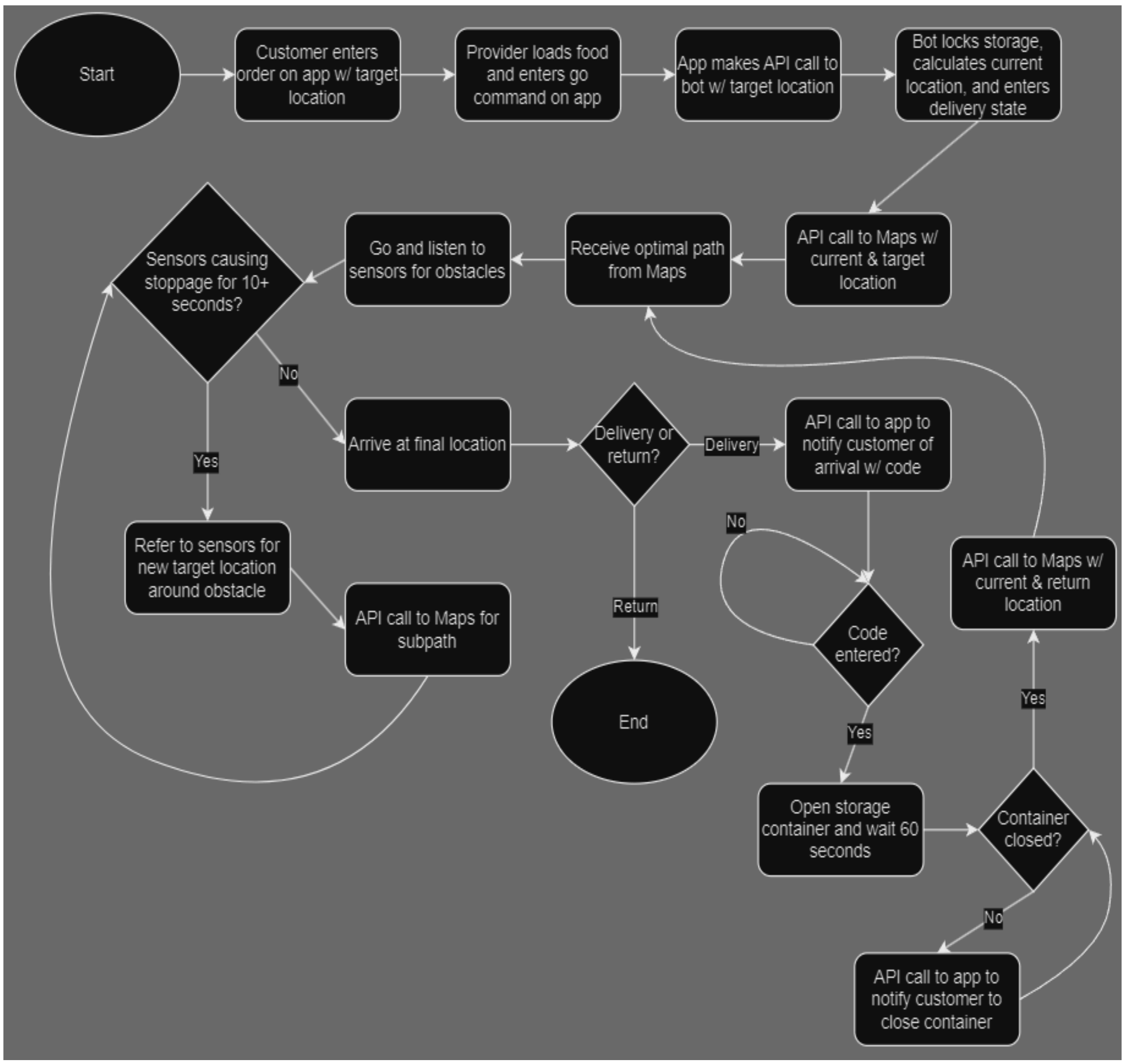


Figure 2.6

2.5 House of Quality

HOQ Introduction:

A crucial element of Quality Function Deployment in the creation of products and processes is the House of Quality (HoQ). It visually links engineering qualities with customer requirements and is shown as a matrix. The importance of the HoQ in bridging the gap between technical design elements and customer expectations is explained in this section. It highlights how it may be used as a cooperative tool to help teams communicate effectively. As a central center, the HoQ clarifies the connections between design characteristics and client needs, which improves the effectiveness of decision-making. It basically guarantees that the finished good or service closely satisfies client needs, which makes it a vital instrument for upholding high levels of quality.

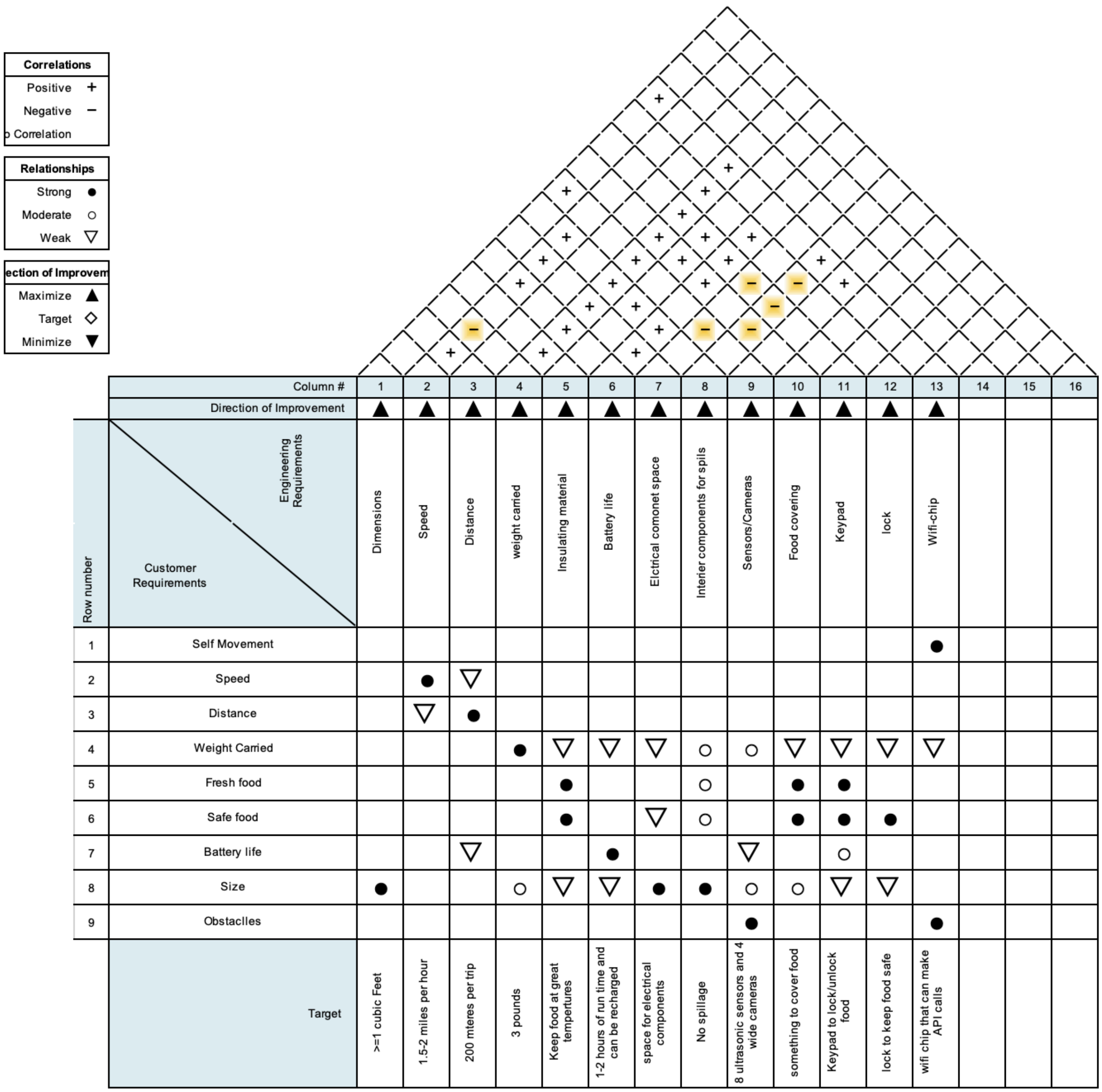


Figure 2.7

3. Research

This research chapter commences by conducting an examination of existing products or projects that bear similarities to FoodieRover. Whether it pertains to their functionality, movement mechanisms, utilized components, or any other relevant aspects, the respective section delves into how these entities align with the envisioned concept of FoodieRover. The subsequent section provides an in-depth analysis of pertinent technologies and strategic components. This analysis involves a comparative evaluation, enabling the identification of the most suitable technology solutions. In the final section, the chosen technology is applied to the selection of specific components for the construction of FoodieRover. This process culminates in the deliberate and informed choice of parts to be used in the development of the FoodieRover project.

3.1 Existing Similar Projects and Products

Similar Product: Kiwibot

Kiwibot represents a groundbreaking achievement in autonomous robotics, purposefully designed to cater to the dynamic food delivery needs of environments resembling college campuses. At its core, Kiwibot is a compact and lightweight Level 4 autonomy robot, driven by artificial intelligence to navigate the intricacies of sidewalks and pedestrian zones in pursuit of delivering delectable sustenance. It distinguishes itself by its dexterity and nimbleness, qualities that enable seamless operation in high-traffic locales.

Intricately equipped with an array of cameras, sensors, and a sophisticated communications module, Kiwibot effectively surmounts the challenges inherent to autonomous mobility. Customers play an integral role in this system through a user-friendly mobile application, facilitating the effortless placement of orders and the convenience of real-time order tracking and notifications. Recognizing the propensity of college campuses for time-constrained schedules and bustling traffic, Kiwibot has successfully established partnerships with various universities across the United States, collaborating with restaurants and businesses to provide an indispensable service.
One of the overarching objectives of the entity steering Kiwibot's development is the conscientious reduction of the carbon footprint associated with the food industry. The robot emerges as a compelling and ecologically sound alternative to conventional gas-powered delivery vehicles. To manifest this vision, the company aspires to extend its

presence to additional locations, becoming an emblem of efficiency and convenience in the realm of food delivery.

Kiwibot's navigation is underpinned by cutting-edge technology. LiDAR, an acronym for Light Detection and Ranging, is instrumental in both mapping and obstacle avoidance. This remote sensing method leverages light pulses to measure distances, enabling the generation of precise 3D models of the surrounding environment through the analysis of reflected pulses. The robot's navigation is further complemented by the integration of GPS technology, facilitating its orientation and coordination in diverse settings. Machine learning algorithms play a pivotal role in enabling the robot to exhibit nimble maneuvering capabilities, aiding in computer vision and ensuring rapid, adaptive responses when confronted with unexpected obstacles.

Drawing inspiration from Kiwibot's successful model, FoodieRover aspires to deliver a seamless user experience and exhibit a comparable Level 4 autonomy. Its mission converges with Kiwibot's in addressing the time-sensitive and bustling demands of high-traffic scenarios while adhering to a profound commitment to environmental sustainability. The FoodieRover design is poised to encapsulate simplicity, cost-efficiency, and effectiveness, striving to replicate the functionality and user-friendly interface that has made Kiwibot a standout innovation in the food delivery landscape.

Similar product: GPK-32 Tracked Inspection Robot

The GPK-32 represents a notable advancement in the domain of robotic technology, devised with a primary focus on inspection tasks within environments that present inherent challenges for human accessibility. Its ability to capture high-definition video and imagery renders it particularly well-suited for professionals in fields such as home inspection, insurance adjustment, and pest control. Notably, the GPK-32 excels in performing inspections within crawlspaces, navigating diverse terrains that may include debris and humidity with exceptional ease. Key elements of its design encompass camera versatility and robust illumination, facilitated by high-intensity LED spotlights that dynamically light up the camera's line of sight.

The effectiveness and reliability of the GPK-32 have been rigorously validated through extensive testing, substantiated by positive feedback from its clientele. It has solidified its position as one of the leading and most sought-after crawlspace inspection robots available today. The advantages it offers to its users encompass safety, efficiency, accessibility, visibility, quality, profitability, and cleanliness. The primary user base primarily consists of professionals who regularly engage in inspections, recognizing the manifold benefits it brings to their respective roles.

The robot itself boasts a chassis constructed from anodized black aluminum, a material lauded for its corrosion resistance and lightweight properties. It features a high-definition camera with 10x zoom capabilities, storing both videos and images on an SD card. The camera's tilt is customizable, ranging from a negative 20-degree angle to a positive 75-degree angle. The robot's battery sustains a continuous operation for up to 2 hours and can be swiftly replaced, ensuring minimal downtime. Its mobility is facilitated by tracks with a zero-turn ratio, enabling precise axis rotation without linear movement, alongside differential steering, affording it the capacity for a wide spectrum of turning maneuvers, be it sharp or extensive.

The GPK-32 package includes a remote control with a 7-inch display, offering live video feeds, a transportation case for convenience, and a removable roll cage along with a wheelie bar. The wheelie bar serves to prevent the robot from toppling over during operation. Its dimensions measure 12.5 inches in width, 9.5 inches in length, and 7.25 inches in height, with a weight of 9 pounds. The GPK-32 Tracked Inspection Robot is competitively priced at $3,600.00.

There are multiple facets of the GPK-32 robot that align with FoodieRover's overarching vision. Foremost is the unwavering commitment to delivering tangible benefits to its customers. FoodieRover seeks to provide advantages akin to those enumerated in the context of the GPK-32, enhancing convenience and efficiency while maintaining a polished design tailored to its intended use. The robot aspires to feature a battery life that aligns with user expectations, complete with rechargeable and replaceable batteries. Incorporating a camera is of paramount importance, not only for navigation but also to enhance security and situational awareness. The principles of zero-turn ratio and differential steering, akin to the GPK-32, are pivotal to FoodieRover's maneuverability strategy in traversing a college campus. Lastly, affordability is a pivotal goal for FoodieRover, aiming to provide its advanced functionality at a cost-effective price point, thereby enhancing accessibility to a broad range of users.

Similar project: Arduino Powered Autonomous “Follow Me” Cooler

The "Follow Me" Cooler project represents a remarkable innovation in the realm of portable, user-friendly cooling solutions. The core concept involves the utilization of a smartphone connected to the cooler via Bluetooth, thereby enabling the cooler to autonomously track and follow the user. What sets this cooler apart is its modular design, allowing it to serve not only as a means of carrying refreshments but also as a versatile cargo transporter. This ingenious system harnesses GPS coordinates, seamlessly streamed from the user's smartphone, to determine its precise location. By constantly comparing these coordinates with its own position, the cooler dynamically adjusts its direction to trail the user faithfully.

Crucially, the cooler interfaces with a sophisticated Internet of Things (IoT) application, bestowing upon users the dual capability of not only controlling when the cooler follows but also designating specific waypoints for it to navigate to independently. Moreover, the cooler's lid is under the discerning command of the user through the app, thanks to a meticulously engineered motor affixed inside the cooler. All the electronic components are housed within a robust wooden frame underneath the cooler, ensuring both durability and reliability.

The cooler frame is equipped with two front wheels, each measuring 6 inches in diameter, and a free-spinning back wheel akin to those found on shopping carts. Powering the front wheels are two 12-volt brushed geared DC motors. Custom-designed motor mounts and connecting components, crafted through 3D printing, securely fasten the motors to the wheels. The frame design accommodates a mid-sized cooler, approximately 17 by 12 inches in width and length. Within this frame, a suite of vital electronics finds its home, including an Arduino Uno, GPS unit, magnetometer, motor driver, and batteries. The Bluetooth module, essential for communication with the user's smartphone, is cleverly situated outside the frame within the chassis.
The intricacies of wiring involve meticulous soldering of fine wires from the motors to the motor driver, alongside the use of jumper cables to connect the motor driver and the GPS unit to the microcontroller. To power the microcontroller, a dedicated 5-volt battery source is employed, which in turn provides energy to the other smaller electronic components. A larger, separate battery is allocated to the motor driver, essential to accommodate the power demands of the motors.

The software driving this technological marvel, programmed within the Arduino Integrated Development Environment (IDE), exhibits a sophisticated algorithm. By leveraging data from the magnetometer to determine the angle between the cooler's orientation relative to the North Pole and the user's GPS coordinates, the system calculates the angular deviation from the North Pole to the user's location. This information, in conjunction with a dynamic analysis of distances, enables precise adjustments in the cooler's trajectory. The outcome is an impressively responsive and agile system that seamlessly follows the user's lead.

While the "Follow Me" Cooler project undoubtedly represents a significant leap forward in the realm of portable cooling and cargo transportation, the FoodieRover project aspires to elevate this technology to new heights. FoodieRover's vision encompasses enhanced autonomy and functionality, catering to a broader spectrum of applications. The project aims to empower FoodieRover with the capability to maneuver effectively, even in less than ideal conditions, cover extensive distances, and synchronize its software interface seamlessly with the food delivery aspect of its multifaceted functionality. In essence, FoodieRover envisions setting a new benchmark in autonomous mobile systems.

3.2 Relevant Technologies and Strategic Components

This section will discuss many of the existing technologies that we are hoping to implement into our project design. In-depth research of the individual components that we intend on purchasing will be summarized below, including their functionalities, characteristics, technical parameters, available varieties, and price ranges. This will allow us to compare components for purchase and understand the items we intend on using to ensure they are compatible with our needs and with each other.

Microcontroller vs Microprocessor

For the FoodieRover project one of the most, if not the most, essential components of the hardware design will be its “heart”. An IC unit in charge of automating all the processes happening within the system. Microcontrollers and microprocessors are two very well-developed types of electronics that can take the functionality of automation both efficiently and at an affordable price. Both have a wide range of options on the market with varying specifications as well as price points.

The main difference between a microcontroller and a microprocessor lies in their applications. A microcontroller contains a processing unit, memory, IO ports, timers, and others all used together to specialize in embedded systems applications. It excels in performing tasks that involve controlling other hardware components. A microprocessor, on the other hand, contains exclusively a powerful processing unit. It specializes in the processing of data at high speeds and amounts. Although they are different, they can both be used to perform tasks specialized for the other.

In order to select between the two, the complexity of the desired tasks must be assessed. In some applications, both a microcontroller and a microprocessor can be implemented with an interface between the two. The microcontroller, with a lesser processing load, would function as a subsystem. Such an application would, for example, entail artificial intelligence in a robot that incorporates sensors, motors, and others.

A safe choice for new projects would be based on microcontrollers. They offer a wide range of performance at an affordable price while including several specialized features that align with automation. In light of the aspiration to implement intricate solutions aimed at enhancing task performance, which demands increased processing capabilities, opting for a microprocessor over a microcontroller would be a more judicious selection. This choice would facilitate the exploration of more intricate applications, including but not limited to artificial intelligence. The table below summarizes the applications that differ in preference for microprocessors and microcontrollers.

Summary Table		
Applications	Microcontroller	Microprocessor
Computer Vision	HD 720p camera	HD 1080p camera
Advanced data processing	-	Preferred
3rd party network connectivity	-	Preferred
Embedded systems	Preferred	-
Artificial Intelligence & embedded systems	-	Preferred

Figure 3.1

Online Routing vs. Offline Routing

In the context of routing, this investigation sought to assess the feasibility of integrating the Google Maps Application Programming Interface (API). Prior to delving into the details pertaining to data exchange facilitated by the API, it is imperative to establish a foundational understanding of what an API encompasses. The acronym "API" signifies an Application Programming Interface, which serves as a set of guidelines and protocols that regulate intercommunication between distinct applications operating across networked environments. To elucidate, consider the scenario of user authentication when accessing a website. The web browser or user's device does not inherently possess knowledge of the requisite username and password, as these credentials are not stored locally. Instead, such information is maintained within a database linked to the web server hosting the website. Consequently, when the user initiates the login process, an API call is initiated, transmitting the provided username and password data to the database for validation. Subsequently, upon receiving this API call, the database conducts an internal verification process to determine a congruence between the supplied credentials and the stored data. Following this verification, another API call, this time emanating from the database, is conveyed back to the server, communicating either a successful or unsuccessful outcome.

Having established a foundational understanding of the function of an API, it is pertinent to delve into its operational mechanisms. Earlier, an analogy likened APIs to regulatory frameworks, and this analogy is indeed apt. Much like regulations, APIs are accompanied by comprehensive documentation, delineating the specific data parameters to include when initiating contact with an endpoint and the nature of the data to be expected in response. The endpoint, frequently presented in the form of a URL, serves as the designated reference point for interfacing with the API. Beyond defining the data exchange parameters, APIs also accommodate a variety of API calls that are contingent on the HTTP method being employed.

- GET
    - The GET call is used to request information from another application.
    - It’s read-only and therefore does not change any data.
- POST
    - POST does the opposite of GET in that it sends data rather than receiving.
    - It can be both write and read (sign-up vs. login).

- PATCH
    - PATCH is used to update information, making it a write-only call.
- DELETE
    - DELETE does exactly what you'd think, permanently removes information.

These four elements constitute the fundamental aspects of API calls, though it's important to note that there are additional components that may be involved. The last, yet integral, facet of an API call is its metadata. This metadata encompasses various parameters, including but not limited to the endpoint, an authentication key, data format, error handling procedures, parameters, and response format. While this may appear to be an extensive list, within the context of FoodieRover's functionality, the response format emerges as the paramount consideration. The underlying concept of FoodieRover's navigation system revolves around the utilization of a GPS sensor to continuously furnish our microcontroller with real-time coordinates denoting its precise location. Subsequently, these coordinates are transmitted to the Google Maps Directions API endpoint, whereupon a detailed JSON object is received, offering comprehensive information regarding the specific directions and maneuvers requisite for successful route navigation. To illustrate, consider the following example of such a JSON response:


```
"distance": { "text": "0.6 km", "value": 615 },
"duration": { "text": "2 mins", "value": 106 },
"end_location":
  { "lat": 39.8681019, "lng": -4.029378299999999 },
"html_instructions": "Head <b>northwest</b> on <b>Av. de la
                       Reconquista</b> toward <b>C. de la Diputación</b>",
"polyline":
  {
          "points": "quhrF`rqWCBQJUJm@PQFg@Ni@JeBh@}@XaD|@
                     {@Vk@Ns@RUFoA^u@R_AXwA`@WHMBG@
                     C?E?GAC?IC",
  },
"start_location": { "lat": 39.862808, "lng": -4.0273727 },
"travel_mode": "DRIVING",
```


The preceding description presents just one among numerous step objects contained within the JSON response. This particular step object, as evident, furnishes a wealth of detailed information, including terminal coordinates, anticipated transit time, and the distance to be covered. The Google Maps Directions API employs a segmentation approach, dissecting the overall route into manageable, task-specific steps, corresponding to distinct maneuvers (e.g., turning left, turning right, and more). This method enhances the tracking process by simplifying the navigation. My aspiration for FoodieRover encompasses the capability to autonomously compute the target coordinates relative to its own position. Subsequently, it would determine the angular disparity between these coordinates, effecting an orientation adjustment that aligns it with the target direction. The rover would then proceed to traverse the prescribed distance. Furthermore, I envision the implementation of predefined parameters, wherein specific maneuvers, like turning right and forking right, would result in distinct angular adjustments, such as 45 degrees and 60 degrees, respectively. This approach would require a very precise GPS sensor in order to accurately ping FoodieRover's location which could present financial strain on the budget.

Alternatively, another approach would be to have FoodieRover handle its own routing by allowing it to map out the environment on its own. In practice, FoodieRover's memory would host two distinct maps: a global map and a local map. The latter would encapsulate the immediate vicinity of FoodieRover, while the former would span the entire intended route. Leveraging its array of sensors, FoodieRover would compile data from the local map into the global map, thereby crafting a route that it deems traversable. It is worth noting that this method is notably reliant on the intricacies of its programming, potentially entailing a more time-intensive process, given the absence of direct control over FoodieRover's mapping decisions. Furthermore, the route derived from this approach, spanning from the initial to the final destinations, may not necessarily represent the most efficient path.

Both of these approaches introduce unique challenges, including the need to sustain a consistent Wi-Fi connection, the intricacies of managing code complexity, and the financial implications of the required instrumentation. In an ideal scenario, a hybrid approach combining elements of both methods could offer a balanced solution. This hybrid approach might involve permitting FoodieRover to autonomously map its immediate surroundings and employ computer vision techniques to discern grassy areas as "red zones" and sidewalks as "green zones," for instance. Once this initial local mapping is completed, FoodieRover could subsequently engage with the API for route generation and overlay the resulting path onto its locally stored map. This approach promises the benefits of an efficient route while maintaining well-defined boundaries, resulting in a dependable and adaptable navigation strategy.

Summary Table		
	Online	Offline
Memory	-	More Intense
Computing	-	More Intense
Network Dependency	Dependent	-

Figure 3.2

Communication Module

A pivotal element within the FoodieRover system is the communication module, a critical component that plays a central role in facilitating data exchange between various subsystems. This communication module can be comprehensively delineated at two distinct levels: the hardware layer and the software layer. While prior discussions have expanded upon the software aspect through reference to APIs, a more detailed examination is warranted regarding the underlying communication protocols employed to enable seamless intercommunication between disparate components.

In the realm of embedded applications, several communication protocols have garnered popularity for their effectiveness in fostering device connectivity. Notable among these are SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), and UART (Universal Asynchronous Receiver/Transmitter). All three of these protocols share a common characteristic—serial communication. This mode of communication entails the transmission of data one bit at a time, resulting in several advantages, particularly in scenarios where the simultaneous operation of multiple devices is not imperative. One of the principal benefits of serial communication lies in its capacity to minimize the complexity of cabling, thereby reducing the number of wires required for data transmission.

Within the FoodieRover system, the application of these communication protocols will be prevalent across a spectrum of devices. Specifically, ultrasonic sensors and the GPS module will be interfaced using UART, while the magnetometer will be harnessed through the implementation of I2C. This strategic allocation of communication protocols to the respective components aligns with the system's functional requirements and device compatibility, ensuring efficient and reliable data exchange across the platform.

However, the camera will not employ communication across these protocols due to their inherent limitations in supporting a sufficiently fast data rate transfer. Instead, the preferred method for connecting the camera entails the utilization of USB technology. This decision leads us to a pivotal aspect of the communication module's hardware implementation.

For the seamless operation of the APIs and the overall functionality of FoodieRover, a dependable and robust Wi-Fi connection is imperative. Consequently, the communication module must possess the capability to establish and maintain a stable Wi-Fi connection to ensure uninterrupted and efficient communication with the requisite network resources.

Summary Table				
	UART	I2C	USB	API
Ultrasonic Sensors	✓	-	-	-
GPS Module	-	✓	-	-
Camera	-	-	✓	-
Routing	-	-	-	✓

Figure 3.3

Camera

For the automated routing system of FoodieRover, the incorporation of a camera and the application of computer vision technology to aid in navigation are part of the strategic plan. The fundamental concept involves the utilization of computer vision principles, such as thresholding and segmentation, to delineate the sidewalk from other non-navigable terrains, such as grass or uneven landscapes. To ensure the optimal performance of this program, the camera to be employed must meet specific criteria. The foremost among these criteria is resolution, denoting the quantity of pixels delivered by the camera, directly influencing image clarity.

Furthermore, an essential metric to consider is the frame rate. Frame rate quantifies the frequency at which image captures are acquired per second. Ideally, a frame rate of 30 frames per second is deemed suitable for a vehicle traveling at a speed of 2 miles per hour, ensuring smooth and effective image processing for the navigation system. Lastly, it is necessary that the camera either has USB connectivity or can be connected by the CSI camera port provided on the Raspberry Pi as we intend to use that to transfer the data.

Summary Table			
Camera Specs	Resolution	Frame Rate	Interface
Minimum Requirements	1080p Full HD	30FPS	USB/CSI

Figure 3.4

<u>Navigation Module</u>

Global Position System, or simply GPS, is an electronic component that uses signals from satellites to determine geographically where the device is located in terms of latitude and longitude. Most GPS modules will also provide the current time. It works as a signal receiver from the earth's surface to multiple satellites within visibility. It is important for the satellites to be visible to the receiver for the trilateration technique to work, which is the main technique used in GPS modules. The technique measures the distances from the satellites to the receiver using the satellite's radio signals to calculate the position. Mathematically the intersection of spheres with radii of the measured distances will be the position of the receiver on Earth, such means that at least three satellites are needed to get a geographic location. In general, a minimum of four satellite signals are used in GPS modules for an accurate reading as the fourth serves to check for errors. GPS satellites broadcast a signal with a PRN (pseudorandom noise) code specific to the satellite and with the time it was transmitted. With the information gathered from the signals and possible errors fixed, the calculations for the latitude and longitude are performed and made available for the user or system.

FoodieRover will use a GPS module to keep track of its real-time position while making deliveries. Its positioning will be transmitted to the software program which will map its path using its current position as a reference. The module must be accurate and reliable for the system to work as a whole. A proper update rate for communications with the program is necessary, the higher the frequency the smoother the tracking. The number of satellite signals it can receive at a time will influence its accuracy, such is determined by the number of channels the module has. The capability to process signals that may be weakened must be good enough for when buildings or trees attenuate the signals. Such goes hand in hand with an antenna that can pick up and amplify weakened signals. Additionally, it is important for the module to be compatible with the chosen microcontroller, specifically for the communication interface.

Using a Global Navigation Satellite System instead of only GPS would provide enhanced accuracy and reliability. GNSS receivers, which are also GPS-compatible, expand the number of satellite signals that can be accessed.

Summary Table	
Parameters of interest	Navigation module
Tracking	Real-time
Accuracy	≥20 channels
Update rate	≥5Hz
Weak signals capability	Antenna amplification
Compatibility	Multiple
Navigation system	GNSS

Figure 3.5

Distance Sensor

It is essential to distinguish distance sensors from proximity sensors since the former focuses on measuring the distance between the sensor and an object, rather than merely detecting the presence of an object within a defined area. Various sensor technologies provide distance information between objects, including ultrasonic, infrared (IR) proximity, laser distance, and others. These technologies share a common approach: they transmit a signal and subsequently analyze its return. Parameters such as the time taken for the signal to return, the intensity of the returning signal, and phase changes are considered in this analysis. While ultrasonic sensors use high-frequency sound waves the others use light beams or pulses, both of which have their advantages and disadvantages.

Some applications that use the speed of light include laptops, security systems, machine vision, and others at varying levels of signal processing. The main advantages associated with such include a high range as well as high accuracy readings. Due to the fast speed of the signal wave, a high update rate can also be achieved which opens the possibility of reading more information about the object being detected. Laser distance sensors and LED Time-of-flight distance sensors can render a 3D images of the objects of interest. A disadvantage associated with both is high cost and possibly be harmful to human eyes. Ultrasonic sensors offer the advantages of low power consumption and multiple interface options at the expense of lower resolution, more limited detection range, and slower update rates. A key application of interest is their wide use for smart cars and as robotics sensors which aligns with the FoodieRover project.

When evaluating various distance sensors and their corresponding merits and drawbacks for the purpose of integrating them into the FoodieRover project, the ultrasonic sensor emerges as the most suitable choice. This selection is primarily driven by the specific requirements of the robot, which will primarily encounter objects in close proximity and operate at relatively low speeds which would not require a high update rate. Its multiple

interface options are also an advantage to the realization of the robot. The implementation of ultrasonic sensors aligns seamlessly with these requirements at a low price point.
To ensure comprehensive coverage of the robot's surroundings and prevent collisions, multiple ultrasonic sensors will be deployed to monitor blind spots effectively. For longer ranges, the navigation module will be utilized, and the possibility of incorporating a camera into the system is also under consideration.

Summary Table			
	Target for design	Sound-based sensor	Light-based sensor
Speed of signal	≥343m/s	343m/s	299 792 458 m/s
Update rate	<0.1 s	~40Hz	~>>40Hz
Distance	2 meters	~2 meters	≥3 meters
Application	High-driving automation	Smart cars	Machine vision
Cost	Low	Low	Low to High

Figure 3.6

Motor Driver

Motor drivers are used to control motors and to supply sufficient current for the motor to achieve the desired rotation per minute. They are placed between the microcontroller and the motor or motors. Speed and direction are two of the main parameters controlled by the motor driver. The instructions for the driver come from the microcontroller for the desired speeds and directions. A microcontroller alone would not be able to have high enough power in its signals to drive a motor to functionality. The circuitry inside the motor driver can change the direction of rotation by changing the polarity of the voltage to the motor terminals. Furthermore, Pulse Width Modulation is used to control the speed by adjusting the duty cycle of the signal. A motor driver also serves as protection to both the microcontroller and motors in case problems arise from operation.

FoodieRover will use a motor driver to control the speed and direction of its motors at any moment to perform the necessary movements to reach its destination. It is therefore a vital component to reach autonomous behavior. It will provide the necessary power to reach the desired speeds in an efficient and reliable way while ensuring the safety of the system. Some motor drivers offer more precise information regarding the performance of the motors by providing feedback to the microcontroller and including sensors.

Summary Table	
Parameters of interest	Motor driver
Motor speed	Current amplifier
Motor direction	Voltage polarity change
Signal duty cycle	PWM
Circuit protection	Current limiter

Figure 3.7

Magnetometer

What is a magnetometer:

- A magnetometer is a scientific instrument used to measure the strength and direction of a magnetic field. It is a device that detects and quantifies the presence of magnetic fields in its vicinity.

What can a Magnetometer be used for:

- Geophysics: Magnetometers are commonly used in geophysical surveys to map and study the Earth's magnetic field. They can help locate buried objects, study the Earth's crustal structure, and detect anomalies associated with geological features.
- Navigation: Magnetometers are essential components in compasses and navigation systems. They help determine the orientation of a vehicle or object relative to the Earth's magnetic field, allowing for accurate navigation.
- Space Exploration: Magnetometers are used on spacecraft to study the magnetic fields of planets, moons, and other celestial bodies. These measurements can provide valuable insights into the composition and interior structure of these bodies.
- Archaeology: Magnetometers are employed in archaeology to detect buried artifacts or structures that may have magnetic properties. They can be useful in non-invasive archaeological surveys.
- Defense and Security: Magnetometers can be used for security purposes to detect the presence of ferromagnetic materials, such as weapons or explosives, in various screening processes.
- Consumer Electronics: Magnetometers are integrated into smartphones and other consumer electronic devices to provide orientation and navigation features, such as compass apps and GPS functionality.

Magnetometer in our robot:

- Orientation and Heading: A magnetometer can provide our robot with accurate heading information. By measuring the Earth's magnetic field, it can help the robot determine its orientation with respect to magnetic north. This information is crucial for maintaining a consistent heading when navigating around the campus. It complements other sensors like gyroscopes and accelerometers, which may suffer from drift over time.
- Compass-Like Functionality: The Magnetometer is like our robots digital compass. It allows the robot to maintain a fixed heading, which is essential for following predefined paths, navigating along cardinal directions, or simply knowing which way it's facing. This can be helpful for tasks like following specific routes on a campus.
- Map Alignment: Our robot will have a second map on top of GPS so a magnetometer can help ensure the robot aligns correctly with the map's cardinal directions. This aids in accurate positioning and following planned routes.

- Enhanced GPS Accuracy: When combined with GPS (Global Positioning System), a magnetometer can help improve the accuracy of location data. It compensates for the fact that GPS alone may not provide precise orientation information.
- Obstacle Avoidance: While a magnetometer is primarily used for heading and orientation, it can indirectly assist in obstacle avoidance. By maintaining a consistent heading, the robot can navigate more effectively around obstacles while following its intended path.
- Calibration and Correction: Magnetometers may require calibration to account for local magnetic anomalies or interference from the robot's own electronics. Implementing calibration routines can ensure accurate heading measurements.
- Energy Efficiency: Properly using a magnetometer to maintain a consistent heading can help in energy-efficient navigation. The robot can avoid unnecessary turns or adjustments, conserving power.
- Having multiple sensors, including a magnetometer, provides redundancy in navigation. If one sensor fails or provides inaccurate data (e.g., due to interference), the robot can rely on other sensors for navigation.
- We have to keep in mind that the magnetometer has limitations. It can be affected by nearby magnetic objects and metallic structures.
    - All this means is that we need proper calibrations and filtering because if we don’t then we can have problems down the road.
- Magnetometers are best used when combined with other sensors like GPS, ultrasonic sensors, and cameras.
    - This will be essential to provide a comprehensive understanding of the robot's environment for safe and efficient navigation on a campus.

<u>DC Motor vs Servo Motor</u>

	DC Motor	Servo Motor
Control Complexity	• Meant for Basic Tasks • More complex for precise control ○ May need encoders and Algorithms	• Designed for precise control with built in feedback mechanisms • Has encoders
Precision	• Suitable for less critical precision tasks. • Achieving high precision may require additional components and programming.	• Designed for precise control, often with built-in encoders.
Ease of Use	• Generally easier to start with for simple applications. • Often less expensive.	• May require more setup and tuning. • Offers finer control over motion and position.
Cost	• Usually less expensive (price range $2-$300). • Budget recommendation: $10-$40.	• Can be more costly (price range $5-$400). • Budget recommendation: $15-$50.

Figure 3.8

What is a Servo motor:

- A servo motor is a type of electric motor that is designed to provide precise control of its output in terms of speed, position, and sometimes torque. It is commonly used in a wide range of applications where accurate and controlled movement is required, such as robotics, industrial automation, CNC machines, and more.

Key features and characteristics of servo motors include:

- Precision Control: Servo motors are known for their ability to provide precise and accurate control over their output. They can maintain a specific position or follow a predefined trajectory with high accuracy.
- Feedback Mechanism: Servo systems typically include feedback devices such as encoders or resolvers. These devices continuously monitor the motor's actual position and provide feedback to a controller, allowing for real-time adjustments to maintain the desired position or speed.
- Closed-Loop Control: Servo motors operate in a closed-loop control system, where the controller compares the desired output (setpoint) to the actual output (feedback) and adjusts the motor's operation accordingly to minimize errors.
- High Torque-to-Inertia Ratio: Servo motors are designed to have a high torque-to-inertia ratio, which means they can accelerate and decelerate quickly while carrying substantial loads.
- Variable Speed: They can operate at various speeds, making them suitable for applications that require both high-speed and low-speed movements.
- Compact Size: Servo motors are often compact and lightweight, making them suitable for use in space-constrained applications.
- Efficiency: They are generally highly efficient, as they only consume power when actively moving to maintain a set position or speed.
- Wide Range of Sizes and Types: Servo motors come in various sizes and types, ranging from small, low-power units to large, high-power motors, with different designs tailored to specific applications.
- Servo motors are commonly used in applications such as robotics for precise limb and joint movements, CNC machinery for accurate cutting and machining, and industrial automation for tasks like conveyor belt control and packaging. They play a crucial role in achieving automation and precision in various industries.

How a servo motor works:

- A servo motor operates based on a closed-loop control system to accurately control its position, speed, or torque. Here's a simplified explanation of how a servo motor works:
- Components of a Servo System:
  - Servo Motor: The motor itself is a key component of the system. It is responsible for generating the mechanical motion.
  - Feedback Device: Typically, a feedback device like an encoder or resolver is attached to the motor's shaft. This device continuously measures the motor's actual position.
  - Controller: A controller, often called a servo controller or drive, is responsible for processing information and generating control signals. It receives input commands and feedback from the feedback device.
  - Control Algorithm: The controller uses a control algorithm to compare the desired position or speed (setpoint) with the actual position (feedback). It calculates an error signal, which represents the difference between the setpoint and the actual position.

    - Power Amplifier: The controller's output is typically sent to a power amplifier, which amplifies the control signals to a level that can drive the servo motor effectively.
- Closed-Loop Control:
    - The servo system operates in a closed-loop control mechanism. This means that it continually compares the desired position or speed with the actual position and adjusts the motor's operation to minimize any error.
    - If there is an error between the desired and actual positions, the controller generates a control signal that directs the motor to move in the necessary direction to reduce the error.
- Feedback Loop:
    - The feedback device (encoder or resolver) provides information about the motor's actual position and feeds this data back to the controller.
    - The controller uses the feedback data to adjust the motor's operation, ensuring it moves precisely to the desired position or speed.
- Continuous Adjustment:
    - As the servo motor moves, the feedback device continuously provides updated position information to the controller.
    - The controller continually adjusts the motor's operation to keep the error as close to zero as possible, making small corrections as needed.
- Achieving Precision:
    - Through this closed-loop control process, the servo motor can achieve high levels of precision, accurately maintaining its position, speed, or torque, even in the presence of external disturbances.
    -

In summary, a servo motor works by using feedback from a position-measuring device to continually adjust its operation to maintain a desired position or speed. This closed-loop control system allows for precise and accurate control, making servo motors suitable for a wide range of applications where precision and control are essential.

What is a DC motor:

· A DC motor, or a Direct Current motor, is an electrical machine that converts electrical energy into mechanical energy through the interaction of magnetic fields. It operates on the principle of Ampère's law, which states that a current-carrying conductor generates a magnetic field around it.

Here's how a basic DC motor works:

· Construction: A DC motor typically consists of a stator (stationary part) and a rotor (rotating part). The stator contains permanent magnets or electromagnets, while the rotor has coils of wire wound around it.

· Application of Voltage: When a DC voltage is applied to the motor, it creates an electric current to flow through the coils on the rotor. This current generates a magnetic field around the rotor.

· Interaction of Magnetic Fields: The magnetic field produced by the rotor interacts with the magnetic field of the stator. This interaction creates a torque on the rotor, causing it to rotate.
· Rotation: As long as the voltage is applied, the rotor will continue to rotate. The direction of rotation can be controlled by reversing the polarity of the voltage applied to the motor.

What can a DC motor be used for:

· Automotive: They are used in electric windows, windshield wipers, and cooling fans.
· Industrial Machinery: DC motors are used in conveyors, lifts, and machine tools.
· Robotics: Many robotic systems use DC motors for precise control of movement.
· Home Appliances: Some household appliances like blenders and electric shavers use DC motors.
· Aerospace: DC motors are used in various systems on aircraft and spacecraft.

Pros and Cons for Dc vs Servo based on what we think we need for our project:

· DC motors spin continuously while servo motors rotate to a precise angle. DC motors are cheaper and easier to control, but servo motors offer more accuracy and power.
·

DC Motor:

Pros:

o DC motors are relatively simple and cost-effective, making them suitable for many applications.
o DC motors can be speed-controlled effectively, which is essential for a self-driving robot to maintain a consistent speed.
o DC motors can provide sufficient torque to move a 15-pound load at 2 miles per hour, especially if geared appropriately.

Cons:

o DC motors do not have inherent position control capabilities. To achieve precise positional accuracy, you may need additional sensors and control algorithms.
o Maintenance: Brushed DC motors require periodic maintenance due to brush wear and commutator issues.

Servo Motor:

Pros:

o Servo motors are designed for precise position control, making them well-suited for applications where accuracy is critical.
o Servo motors can offer high torque relative to their size, which can be beneficial for carrying heavy loads.
o Servo motors are often more efficient and generate less heat than brushed DC motors.
o Servo motors are typically brushless and require less maintenance.

Cons:

o Servo systems can be more complex to set up and control, potentially requiring more sophisticated control algorithms and feedback systems.
o Cost: Servo motors and their associated control systems tend to be more expensive than basic DC motors.

Considering our robot's requirements for speed and load capacity, a DC motor is the best choice for our project

Locking mechanisms

Given that one objective of this project is to keep the food secure as it is delivered, a locking mechanism must be implemented to prevent unwanted people from opening the container before the target destination has been reached.

The many available types of locking mechanisms on the market include: a mechanical lock and key padlock where a physical key must be inserted into the lock and turned; a combination lock where a (typically 4-6 digit) code must be entered by the user, such as those seen on school lockers or bike locks; deadbolt locks like those used in many home front doors where a metal piece fits into a cutout, preventing motion in the perpendicular direction; and electromagnetic locks which use an applied power source and forces of electromagnetism to draw together two metal components strongly enough to prevent someone from being able to separate them by force.

There are two main limiting aspects of this project in terms of choosing a lock. The first is the physical structure of the robot itself. Given that we are attempting to lock a small container to avoid a lid from being removed, we need to consider locks that can block parallel movement, like lifting a lid off of a cooler, instead of perpendicular movement, like swinging a door open, which most locks are designed for. The container which holds the food will likely be a cooler, given the temperature-regulating objective of the project, for which advanced locks are not common or easy to implement. A lock will have to be attached in some form to the lid of the container itself, providing a challenge in terms of the structural integrity of the container if we are to attach a heavy-duty locking mechanism to it. Most locking mechanisms are designed for use on flat surfaces as well, which are not common among many cooler containers. Thus, creative thinking will need to be applied to not only choose an appropriate locking mechanism but to determine how best to attach it to the physical structure itself. The second limitation is that we are not have the capability to supply the delivery recipient with any sort of physical component to open the lock (like a physical key). The key point of the delivery robot is that it is fully remotely controlled, so any sort of physical delivery of a key or other item used to unlock the container would defeat the purpose. A key cannot be provided via the robot itself either, because this would prove the locking mechanism pointless as another security feature would be needed to prevent passersby from accessing the key and unlocking the container. Even if a locked box was used to hold it, a physical key would still not be ideal because of the potential for it to get lost or stolen among the delivery recipients who fail to return the key to the robot, rendering the lock useless and causing annoyance for users who subsequently request the service before it is realized that the key is locked and there is no way to access their meal. Because of this, any option that relies on a physical key or other physical component to unlock the mechanism is not desirable. Another less vital but still important consideration is that this locking mechanism should ideally be reconfigurable between users so as to ensure security over time. If the lock code is always the same, past users of the delivery service or others who find out the code from word of mouth, social media, etc. will be able to open the lock any subsequent time they come across the robot. For guaranteed safe delivery each time, the code for the lock must be able to change often. Ideally, this process would be able to be done via remote configuration rather than a physical replacement of the locking mechanism each time the code is to be changed.

Locking control must be considered as well as the physical lock. Locking control encompasses the element which decides whether the mechanism is locked or unlocked, and can change that status when needed. An example of a locking control mechanism is a physical key, which, as discussed above, is not ideal for this application as we have no way of delivering a physical object to each customer using the delivery service. Other examples of locking control mechanisms include a combination for a combination dial lock in which the combination is set upon purchase and the lock only opens when the correct combination is entered. This type of locking control will not work either since we need to be able to change the code between users to prevent the code from becoming widely known and no longer a secret among the nearby population where the robot operates. Another type of locking control is via electronic control from a microcontroller/microprocessor, where this unit is able to recognize input from a keypad, determine if it is the correct code, and unlock the physical lock accordingly. An additional benefit to this type of locking control, which is important for consideration in this application, is that a microprocessor can be reprogrammed to change the code on a regular basis as needed, whether that is between each user, once daily, etc.

Physical locking mechanisms also need to be considered in detail to determine the one that will work best with the delivery robot. Some physical locking mechanisms include a deadbolt, as seen in most household front doors; a bike lock or similar item that runs a strong cord through some sort of opening and is secured, usually, by a physical combination locking control attached to the cord itself; a padlock, which is a small, usually metal lock with a round metal piece which can be slid through a small opening and back into the lock itself to secure and is also typically seen with a key or physical combination dial as its control mechanism. Another type of physical locking mechanism is an electromagnetic lock. These locks are more conceptually more complex than most widespread physical locking mechanisms because their ability to lock something relies on the application of electromagnetic force created by an applied charge. Electromagnetic locks are security locks that operate by creating an electromagnetic force between two components: an electromagnet and an armature plate. The electromagnet must be connected to power in order to be activated, creating a magnetic flux that attracts the armature plate and effectively “locks” the mechanism. When the power supply is removed from the electromagnet, the magnetic flux between the components is not present and the components can easily be pulled apart, thus effectively “unlocking” the mechanism. The armature plate is usually attached to the moving part of the system, such as the door (or in our case, the lid of the cooler), and the electromagnet is attached to the fixed part, such as the door frame (or the cooler body). The components are attached to the system in such a way that when the door (or cooler lid) is closed, the two components are in contact with each other so that they can properly attract each other when the mechanism is activated (locked). Electromagnetic locks are particularly interesting for this application because their design allows for blocking of parallel movement (pulling the pieces directly apart from each other), something most physical locking mechanisms are not designed for. Additionally, electromagnetic locks are designed to run based on a power input (or lack thereof) which determines the status of the lock. Therefore, an electromagnetic lock could be configured to work with our microprocessor which will supply power when the system should be locked, and turn that power off when unlocking is allowed. These types of locks are simple in structure, but they do rely on a power source for operation.

Below is a table summarizing and comparing the main types of locking mechanisms considered for this project:

	Padlock	Deadbolt	Bike lock	EM lock
Control options	Physical key; Combination; Fingerprint; Application	Physical key	Physical key; Combination; Application	Control via power source
Physical Compatibility with Delivery Container	Possible; requires drilling holes	Complicated; requires complex structure attachment	Possible; requires drilling holes	Possible; requires drilling into container
Reconfigurable	Some	No	No	Yes (via microprocessor control)

Figure 3.9

When considering the design of this project, the most practical locking mechanisms for this specific application include electromagnetic locks or advanced digital padlocks which can be reconfigured via connection to a microprocessor.

Power Source

There are many factors to consider when choosing an appropriate power source (or sources) for a project such as this one. We need to supply power to the motors to allow for movement as well as the microcontroller(s) to allow for communication to the motors and processing of sensor data. The power supplies will need to be able to provide enough power to the robot for it to operate for the designated runtime: 1-2 hours between charges. Enough power must be provided to the robot to keep the motors and the microcontrollers fully operational over this course of time.

Another important consideration for this project is the fact that our robot will be mobile. This provides an additional set of needs that should or must be met for optimal performance. The power source for a mobile robot must not be a tether, which provides a limited range of motion for the robot given that it must stay plugged into the source with some sort of cord of finite length. Instead, the robot will need a power source that is internal to the design and is able to dock and recharge when not in use.

The most prevalent of the possible electric power sources in robotics are batteries. Batteries come in many different types, sizes, and capabilities to meet the need of any robotics project, so it is important to ensure that the battery selected is suited for a specific application. Other sources of power in robotics include thermoelectric generators, photovoltaic cells, fuel cells, supercapacitors, compressed gases, and tethers. Given the scope of this project and the desire to have many options to select from when making our specific product choices, the most desirable choice for a power source for this project will be some variation of a battery.

The below table compares several common sources of power in robotics applications for features relevant to this project:

	Batteries	Generators	PV cells	Fuel cells	Tethers
Size/weight	Small (~2 lbs)	Large (~50 lbs)	Medium (~12 lbs)	Medium (~15 lbs)	Varies in length (~2-100ft)
Price	$40	$500	$100	$140	Usually included with chassis
Allows for mobility	Yes	No	Yes	Yes	No

Figure 3.10

When selecting a specific battery to supply power to a system, many specifications must be considered. These specifications include the voltage rating of the battery - the maximum voltage which a battery delivers to connected components. The current rating, also known as the capacity rating, is the current which can be delivered by the battery in one hour, usually measured in Amp-hours or milliAmp-hours. This rating can be used to find how long the battery will operate depending on the actual current being drawn from it. For example, if a battery has a capacity rating of 10 Amp-hours, it can supply 10 Amps for 1 hour, or 5 Amps for 2 hours, or 20 Amps for half an hour, etc. Thus, the current rating can be divided by the current drawn to find out how long, in hours, the battery will be able to operate with the given current. The C Rate of a battery is used to express the discharge current normalized in terms of the current rating. For example, a 1C discharge current is the current at which the battery will take 1 hour to discharge. For a battery with a capacity rating of 10 Amp-hours, as discussed above, 1C would mean a discharge current of 10 Amps. A C-rate of 2C would then be equivalent to 20 Amps, and a C/4 rate means 2.5 Amps. An E-rate follows the same concept as a C-rate, but expresses the discharge power normalized in terms of power capacity. An E-rate of 1E signifies the discharge power at which the battery will be fully depleted after 1 hour.

Even with all of these specifications considered, there are still many different types of batteries to choose from when selecting an appropriate power source. A common type of battery is the alkaline battery. This is the type that is commonly found in battery-operated household appliances and toys. Alkaline batteries can have 1.5 Volt ratings - such as AA batteries which have a current rating of about 1000 mAh, AAA batteries whose current ratings are smaller, C batteries with about 2500 mAh ratings, and D batteries with about 2000 mAh ratings. Alkaline batteries also come in a 9 Volt variety with current ratings ranging from 50 to 500 mAh. Alkaline batteries are non-rechargeable and, compared with other battery types, do not tend to last as long. This is an important consideration for this project given that we aspire for the robot to be rechargeable without the need to physically switch the batteries. On the other hand, they are usually cheaper than other options and more widely available. Other types of batteries include Nickel-Metal Hydride Batteries, also known as Ni-MH batteries. Each cell of Ni-MH batteries has a voltage rating of 1.2 Volts and their current ratings range from about 500 mAh to about 3000 mAh. One important desirable aspect of Ni-MH batteries is that they tend to last longer, and unlike Alkaline batteries, they are rechargeable. However, because of these traits, Ni-MH batteries can be more expensive. They are also less sensitive to temperature than Li-Ion batteries (discussed later), which gives them an edge in this application since the system must be able to operate in a wide range of temperatures given that it will be outdoors. A third battery category is the Lead-Acid/SLA Battery. These batteries are overall cheap and highly accessible, with relatively high capacities, and can have high levels of current output. They come in a range of voltage ratings, including 6 and 12 Volts, usually with cell voltages of 1.5 Volts each. SLA batteries can last much longer than other types and can also be charged. One drawback of SLA batteries is that they can be very heavy and are not recommended for smaller robots. SLA batteries are definitely an option to consider for our project for their high capacity and low price point, providing one is available at a weight that will not compromise the integrity of the robot structure. Lithium-Ion (LI-Ion) batteries are also widely accessible and come with high capacities. These batteries are also rechargeable, and one particularly desirable feature in this case especially is that they are lightweight and compact. However, they are sensitive to charging limits and temperature, so their circuitry must be carefully designed to keep the battery operating in the desired conditions. Lithium-ion battery cells have a voltage rating of 3.7 Volts but can come in a variety of voltages including 12, 24, and 48 Volts. Another battery option is a Lithium Polymer (Li-Po) battery. Voltage ratings for Li-Po batteries come in 3.7 Volt increments. These batteries are increasingly popular for robotics applications because they have desirable capacities and discharge rates, and are also considered lightweight.

When selecting a type of battery that will be used to power a motor, the motor specifications must also be taken into consideration. Given that DC motors are most commonly used in robotics, and we have previously decided that a DC motor is optimal for our project, we consider important specifications of DC motors here. For example, the voltage rating of the motor must be known to ensure that the voltage output of the battery does not exceed the maximum voltage that can be supported by the motor, which could result in breakdown. Free/no load current as well as stall current for the motor should also be known when selecting a power source. The free current is the - usually small - current at which a motor will run without the presence of a connected load. It follows that the current at which the motor will run with a load attached will be larger than this no-load current. The stalling current of a motor is the maximum current drawn by the motor when it is stalling, or not rotating. Motor stalling can occur in breakdown torque condition, when the torque of the load is larger than the torque of the motor shaft. Thus, the motor doesn't rotate but continues to draw a high level of current. Additionally, motors come with a power rating which quantifies the voltage and current ratings together through the formula $P = V \times I$. Thus, the power rating for two motors may be the same even though they have different voltage and current ratings, so it is important to not base the motor and battery selection combination on this alone and look at voltage and current ratings individually as well. The torque of a motor is the circular force it applies. It can also be considered as the weight which a motor is able to carry over a certain distance. A motor's specifications will include its torque and rotations per minute (RPM). These values, along with an efficiency constant, can be used to calculate the total energy that the battery needs to supply to the motor. This energy, in Joules, is equal to the power rating of the motor multiplied by the time frame for which it must supply this power ($E = P \times T$). Multiplying an efficiency factor greater than 1 (around 1.2) to this value will help account for power losses due to inefficiency. This factor should be larger if the battery will also be powering other electronics in the system (like a microcontroller), and it is still important to consider factors like the weight of the load as well. The final result of these calculations will represent the total energy which is required from the battery. Therefore, a battery with at least this capacity should be chosen to ensure that it will meet the power demands of the motors and the rest of the system.

Even once a battery has been determined to be compatible with the parts it needs to connect to, there are still further factors to consider. For example, the size and weight of the battery must be small enough that it does not produce an unbearable weight on the robot structure or significantly slow it down. The environment in which the battery will be used in is also important: our robot will operate in an outdoor environment, so it will be exposed to a range of temperatures as well as the potential for rain. Therefore, the battery must either be weather-proof itself, or care should be taken by the team to ensure that the placement of the battery within the robot structure protects it from any unwanted exposure to the elements. While the battery capacity itself - the duration of power it can provide to the required units - is important, so is the charging speed of the battery. At times, the robot may be required to make multiple trips with short or no breaks in between. The faster the charging speed of the battery, the more deliveries the robot will be able to complete in a shorter amount of time by minimizing the waiting period for the battery to recharge when necessary.

3.3 Part Selection

This section will provide a detailed account of the specific components to be acquired for the creation of the FoodieRover design. The ultimate choices, which are explicitly highlighted, have been arrived at through a systematic assessment of the information presented in the Relevant Technologies and Strategic Components section. A comprehensive evaluation of a range of alternatives was undertaken, with each option subjected to a thorough examination based on their key attributes.

Frame/Chassis

In the world of robotics, choosing the right chassis is essential to a project's success. Examining a variety of choices, we examine multiple state-of-the-art robot chassis models, each with unique features designed for different use cases. Through an analysis of the price, weight-bearing capacity, motor characteristics, size, wheel arrangements, and frame elements, we enable a well-informed investigation of these innovative robotic workhorses, enabling us to select the best option for our project.

Robot Frame/Chassis	SZDoit Smart Shock Absorption Robot Tank Car Chassis	SZDoit Big Load Large Size Robot Tank Car Chassis	DoHome Professional Large Size Big Load Metal Robot Tank Chassis	50kg Load Large C600 Full Metal Smart Robot Car Chassis	Hiwonder Tank Car Chassis
Cost	$70	$200	$169	$118	$70
Weight it an carry	3Kg	6Kg	6Kg	50Kg	5Kg
Motors it comes with	2pc DC encoder Motors	4Pcs DC encoder motors	4Pcs DC encoder motors	4 high-torque 37mm motors	2 DC ecoder motors
size	L: 275mm W: 190mm H:90mm	L:550mm W: 280mm H: 110mm	L:550mm W: 280mm H: 110mm	L: 300mm W: 200mm H:95mm	L: 306mm W: 246mm H:95mm
Wheels/tracks	2 Plastic Tracks	4 Plastic Tracks	4 Plastic Tracks	4 Metal Wheels	2 Plastic tracks
Parts	3 parts of Metal Frame	5 parts of Metal Frame	5 parts of Metal Frame	5 parts of Metal Frame	3 parts of metal frame

Figure 3.11

Magnetometer

The effective functioning of our project depends on the selection of the appropriate magnetometer, which permits accurate navigation and orientation in dynamic situations. An overview of many magnetometer options designed specifically for compact food robots is provided in this table, which guarantees optimal performance and precision in detecting and reacting to magnetic fields. This table attempts to assist in the process of choosing which magnetometer technology to incorporate into our project in order to improve its effectiveness and adaptability in world obstacles.

Magnetometer	Compass Module 3-Axis HMC5883L	SparkFun Micro Magnetometer - MMC5983MA (Qwiic)	Adafruit Triple-axis Magnetometer - LIS3MDL – Stemma QT/Qwiic	GY-511 LSM303DLHC Module e-Compass 3 Axis Accelerometer + 3 Axis Magnetometer Module Sensor
Communication protocol	I2C (400kHz)	I2C (400kHz) SPI	I2C (400kHz) SPI	I2C (400kHz)
Sensing ranges	±8 Gauss	±8 Gauss	±4 to ±16 Gauss	±1.3 to ±8.1 Gauss
Heading accuracy	1° to 2°	0.5°	1° to 2°	1° to 2°
Size	18mm x 17mm	3.0mm x 3.0mm x 1.0mm	25.7mm x 17.8mm x 4.6mm	3mm x 5mm x 1mm
Supply Voltage	2.7 to 6.5 V	2.8 to 3.6 V	3.3 to 5V	2.16V to 3.6V
price	$7	$16	$10	$9

Figure 3.13

Microcontroller/Microprocessor

We thoroughly assessed both microcontrollers and microprocessors for the FoodieRover project. This comprehensive evaluation was conducted due to the wide range of tasks it is expected to perform, some of which may necessitate a robust processing unit to handle high computational workloads. The selection of a microprocessor offers the advantage of scalability, enabling us to enhance the intricacy of the software design while maintaining seamless hardware control.

Our decision was significantly influenced by key criteria, including the presence of Wi-Fi and Bluetooth capabilities, memory capacity, processing speed, and others. These factors played a pivotal role in our choice of the appropriate processing platform for the FoodieRover.

Microprocessor	Raspberry Pi 4 B	NVIDIA Jetson Nano
Core	Quadcore64-bitARM-Cortex A72	Quad-core ARM Cortex-A57 MPCore processor
GPU	Broadcom VideoCore VI	128-core Maxwell GPU
Memory	2GB, 4GB, or 8GB LPDDR4-3200 SDRAM (depending on model)	4GB 64-bit LPDDR4
Storage	Micro-SD card	Micro-SD card, 16GB eMMC
USB Ports	2 USB 3.0 ports; 2 USB 2.0 ports	4 USB 3.0 Ports
Networking	Gigabit Ethernet, 2.4 GHz, and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE	Gigabit Ethernet, M.2 Key E
GPIO	40	40
Power Requirement	5V/3A	5V/4A
Cost	35-75	~$99

Figure 3.14

To guarantee the best decision-making process, we thoroughly evaluated the peripherals of each microcontroller. These peripherals are crucial as they provide essential information about the available communication interfaces for connecting with other components, the allocation of pins for connections, and the inclusion of additional peripheral features.

Microcontroller	ESP32-S3	Raspberry Pi Pico W	Arduino Uno
Core	Xtensa® dual-core 32-bit LX7	Dual-core ARM Cortex-M0+	Single-core ATmega328P
Wi-fi Protocols	802.11b/g/n, 2.4GHz	802.11b/g/n 2.4GHz	NA
Bluetooth	5.0	5.2	NA
Typical frequency	240MHz	133MHz	16MHz
SRAM/ROM	512KB/384KB	264KB	2KB SRAM / 32KB Flash
GPIO	45	26	20
ADC	2, 12-bit, 20 channel	3	1, 10-bit, 6 channel
SPI	4	2	1
UART	3	2	1
I2C	2	2	1
Cost	~20$	~15$	~15$

Figure 3.15

<u>Navigation system</u>

Choosing the proper navigation system is of imperative importance if we expect our project to reach the desired level of autonomy. It needs to be quantifiably accurate, reliable, and fast for FoodieRover to move in the right path. While the sensors will make sure nothing near the bot will lead to a crash, the navigation module will focus on the bigger scope and will lead the bot toward its final destination. In simple words, for the bot to know where to go, it first needs to know where it is and that is the task of the navigation module. The specifications of interest when looking at parts consist of how accurately it can assess position, the speed at which it can update position, how it can handle weak signals from satellites, and its compatibility.

Navigation Module	NEO-F9P-15B	NEO-M8P-2	ZED-F9P-04B
Position accuracy	0.04 m CEP	0.025 m CEP	0.04 m CEP
Tracking and navigation	-167 dBm	-160 dBm	-167 dBm
Update rate	15 Hz	5 Hz	5 Hz
Input voltage	3.6V	3.6V	3.6V
Interface	I2C, SPI, and UART	UART, SPI	I2C, SPI, and UART
Cost	$219.99	$179.95	$209.99

Figure 3.16

Motor Driver

When it comes to choosing the motor driver, the specifications of greatest interest consist of its input and output voltages, the constant current it can maintain while in operation, and how many motors it can control. The motors we choose to use will dictate how many amps of constant current we need to supply and the output voltage. The way we power the motor driver will be influenced by the choice of motor driver which is the reason why the input voltage is of interest. The parts chosen to be compared are diverse in both performances, with regards to the specifications of interest, and price. The selection of the part will expectedly favor the specifications of interest but will also take into account special features that can be of use or of significant positive impact.

Motor drivers for DC motor operation:

Motor driver	L298N	BTS7960B	TB6612FNG	VNH2SP30	Cytron 10A
Channels	2	1	2	1	2
Supply voltage	5V	5.5 to 27.5 V	2.7 to 5.5 V	5.5V	3 to 5.5V
PWM	Yes	Yes	Yes	Yes	Yes
Motor voltage	4.5 to 46 V	5.5 to 27.5V	2.5 to 13.5 V	12 to 16 V	5 to 30 V
Constant current	2A	40A	1.2A	14A	10A
Peak current	3A	60A	3.2A	30A	30A
Additional features of interest	Heatsink	Status flag	Standby mode	Current sense feedback	Activation buttons
Cost	$7	$3	$3	$16	$23.50

Figure 3.17

*MOSFET-based motor drivers are preferred for higher current output.

Distance Sensor

When considering the utility of the distance sensor, our objective is to address any potential blind spots in the vicinity of FoodieRover. It is imperative to acquire precise information pertaining to both the distance of objects and the respective angles at which these objects are situated. Achieving this objective may necessitate deploying multiple sensors or a highly sophisticated sensor configuration, which could substantially augment the overall project costs. Nevertheless, it is crucial to emphasize that a prioritization of simplicity over intricacy is paramount, given that additional features beyond distance sensing hold no practical utility. This principle has profoundly influenced the selection of sensors seen in the table provided below.

Sensor	URM07	HC-SR505	HC-SR04
Type	Ultrasonic	IR	Ultrasonic
Operating voltage	3.0-5.5V	4.5-20V	5V
Effective range	0.2-7.5m	3m	0.02-4m
Direction angle	60 degrees	<100 degrees	30 degrees
Communication	UART	Digital I/O	Digital I/O
Operating current	5mA (14µA standby)	<60µA	15mA
Other	Built-in temperature compensation	Minimum size (10*23mm)	Working frequency 40Hz
Cost	$16 for 1	$11 for 5	$12 for 8

Figure 3.18

Locking Mechanism

Given that most existing locking mechanisms are not designed for purposes similar to this project, only a few options exist that would work with our design. To meet the proposed level of security, the chosen locking mechanism needs to be able to be compatible with the physical structure of the robot, must not require a physical key or other material object to unlock, and should ideally be remotely reconfigurable so that the password can be changed between users of the delivery service. It was previously determined that the best options for locking mechanisms to meet these objectives include electronic padlocks with password-reconfiguration capabilities, and electromagnetic locks that can be connected to and controlled by our microprocessor.

Available varieties: There are many types of electronic padlocks available for purchase online. However, not all of these are compatible with our project's needs. Electronic padlocks come in a variety of sizes and many have different means of locking/unlocking control, such as via an application, fingerprint, or digital code. Electromagnetic locks are more standardized and have less variation between the available options, however, there are still many accessible listings for purchase. The main variations between the available options online include the physical size of the components (electromagnet and armature plate), the holding force it can withstand, and fail-safe versus fail-secure options.

Control: Upon an initial search, most electronic padlocks either use a fingerprint for verification, unlocking via an associated application created by the lock developer, or a digital code which can be changed or overridden via an associated app. These options are not ideal since the fingerprint of the recipient would change for each order, and we do not want to complicate the design by requiring installation of a specific app just for the locking capabilities. Electromagnetic locks are controlled by applying or removing a power source from the components to create or remove the electromagnetic force between the components. Turning this power supply on or off can be done via programming of the microcontroller to which the EM lock obtains its power from, which makes this type of lock very functionally compatible with our system.

Power source: The vast majority of digital padlocks are battery-powered, meaning the lock's battery would eventually have to be manually changed or recharged. Many of the varieties require common AA or AAA alkaline batteries that would need to be changed once their power is depleted, while some have built in battery packs that need to be plugged in for recharging. Neither of these are desirable since we are aiming to have a centralized power supply without the need to recharge individual components. Electromagnetic locks offer control via an attached microcontroller and also are able to draw their operating power from the same unit. This is a huge advantage for our project since it would limit the need for an additional power source for just the locking mechanism, however it will mean that we have to consider the power drawn by the lock when choosing appropriate power sources for the project and their capacities.

Physical compatibility with system: Neither electronic padlocks nor electromagnetic locks will be seamlessly compatible with our delivery unit. The structure of a padlock will require the team to drill holes in the lid and body of the delivery container/cooler to be used so that the round portion of the padlock can hold these two components in place when locked. The electromagnetic lock would also require some modification to the cooler as well, and there is not a clear place to put the lock components on the structure in a seamless manner. The pieces could be on the inner part of the lid and the body so that they are flush with each other when the lid is closed, but this may prevent the lid from fully closing and therefore take away from the container's insulating properties. Other ways of making sure the pieces of the lock are in contact when the lid is closed include attaching a solid support structure, like a small piece of wood, to the inside of the cooler where one piece of the lock can lie on it while the other is attached to the underside of the lid, for example.

Strength/reliability: Padlocks are typically very strong, can withstand extremely strong forces, and offer a very reliable standard for keeping something closed. Higher-end options available online claim to withstand 15 kN cutting force, 5 kN pulling force, and 100 Nm twisting force. The simple mechanical configuration of the padlock mechanism means that the lock will not come unlocked prematurely in the event of a dead battery or loss of connection to the locking control mechanism (i.e. application). Electromagnetic locks come in varying levels of strength with the price getting higher as the lock can withstand more force. Initial search results show EM locks with security ratings ranging from 130 to 1200 pounds of force, impressive but less than that of a quality padlock. Additionally, EM locks may be less reliable as they lose their capabilities in the case of a power failure. Some EM locks are fail-safe, meaning they are designed to unlock in the event of a power loss. Others are fail-secure, meaning if their power supply is cut off they will stay locked. This option is beneficial because the different failure mechanisms may be suitable for different applications.

Price: Sophisticated electronic padlocks that would work for our needs are available in a price range of $60 to $150, depending on how advanced the technology is. Electromagnetic locks are available on Amazon for anywhere between $15 and $65, depending mostly on the included components and the amount of force they are capable of withstanding.

	Electronic Padlock	Electromagnetic Lock
Available Varieties	Many varieties in size, features, etc.	Less variety; most options similar in design
Control	Control via associated app by lock developers	Control via turning power source on/off (can be done via microprocessor programming)
Power Source	Battery-powered; some rechargeable	Can be powered via connection to microprocessor
Physical Compatibility with Container	Requires drilling holes for metal padlock loop	Requires gluing/drilling lock parts; may require internal support structure
Strength	Up to 15 kN cutting force, 5 kN pulling force, 100 Nm twisting force	Varying strength options (130-1200 lbs); physical design doesn't allow cutting or twisting of lock
Reliability	Will not come unlocked prematurely due to simple mechanical design	2 options: Fail-safe: unlocks in the event of power failure Fail-secure: remains locked in the event of power failure
Price	\$60 - \$170	\$15 - \$65

Figure 3.19

Due to their compatibility with other components of our project as well as their lower price point, an electromagnetic lock will likely be the optimal choice for a locking mechanism. A more advanced implementation of this concept in the future could consider this area as one to improve by investing in a more intricate and advanced locking technology that has the ability to integrate with the consumer application used for the service.

Electromagnetic locks, however, still come with some variances that must be carefully considered. Upon an online search, most EM locks operate with a 12 Volt power source; this is important to keep in mind when designing how the system will provide power to the lock. The available locks have some variance in size, so one which fits well physically with our chosen carrier container will be chosen. The main difference among the available EM locks is their holding force, or the amount of force they can withstand before being pulled apart. This value ranges from 60 kg (130 lbs) to 500 kg (1200 lbs), with prices increasing as the holding force increases all the way up to $80 for the strongest locks. For the scope of this project, a lock with a relatively low holding force will likely be chosen due to their reasonable prices and the low-pressure nature of this application. The following table specifies some of the options available.

Magnetic lock	AGPtek® 60kg 130LBs Holding Force Electric Magnetic Lock	Electric Magnetic Lock 280KG 600LBS Holding Force	COUNS Single Door 12V 600lbs Electromagnetic Lock
Holding Force	130 lbs	600 lbs	600 lbs
Price	$21	$40	$36

Figure 3.20

The AGPtek Electric Magnetic Lock will suit our needs for this project. It requires a 12 Volt power source to operate, so a relay component must be considered to interconnect the microprocessor with the lock, given that connection to 12 Volts will likely burn out the microprocessor circuitry. The lock is small in size and should be able to fit on the lip of the container, or otherwise attached via an internal support structure which will need to be built inside the container. The current draw is 0.11 to 0.15 Amps which will also need to be considered with the compatibility of the microprocessor to supply this current. The lock claims to withstand up to 60 kg or 130 lbs of force, which may not hold back the most determined intruder but should serve its purpose for the demonstration of our design.

Power Source:

It was previously determined that some form of battery is the most desirable element to provide power to our system. However, batteries for robotics come in a wide array of varieties with many different factors and features that will be considered here. The main battery types listed in the previous section are summarized below.

	Alkaline Battery	Ni-MH Battery	SLA Battery	Li-Ion Battery	Li-Po Battery
Varieties	1.5 V; 9 V	1.2 V per cell	1.5 V per cell	3.7 V per cell	3.7 V per cell
Rechargeable	No	Yes	Yes	Yes	Yes
Cost	$5-12 for 12V options	$20-40 for 12V options	$20-50 for 12V options	$20-90 for 12V options	$30-50 for 12V options
Size/weight	Smallest (~ 1 oz)	Medium (~ 9 oz)	Large (~4.5 lb)	Medium (~13 oz)	Large (~2.5 lb)
Other		Less temperature sensitive than Li-Ion	High current output	Can be sensitive to temperature	

Figure 3.21

*Final selection of battery type will be made in the following discussion.

Among the battery choices, alkaline batteries can confidently be eliminated given that they are not rechargeable, and this is a key design aspect of this project. Lithium-ion batteries can be dangerous because they have been known to react poorly to overcharging and operating outside their temperature range, however, this type of battery has many desirable traits as well that may prove to outweigh the risks included. Given that there appears to be high availability of each of these types of batteries in low price ranges, price differences will not significantly impact the decision either. Thus, the type of battery should be chosen mainly for its lifetime and compatibility with the chosen motors, motor drivers, and other circuitry. Careful consideration of the operating parameters of each power-drawing unit resulted in the calculations summarized in the table below.

	Quantity	Supply Voltage	Peak Current	Power (W)
Motor Drivers	2	5 V	4 A	40
US Sensors	8	5 V	15 mA	0.6
GPS Mod	1	5 V	29 mA	0.145
Microprocessor	1	5 V	3 A	15
Magnetometer	1	3.3 V	100 uA	0.00033
Maglock	1	12 V	150 mA	1.8
Motors	4	12 V	1.2 A	57.6
Camera	1	3.3 V	300 mA	0.99
Lock Relay	1	5 V	5 mA	0.025
Total Power				116.16033
			Capacity Needed:	232.32 Wh
			At 12 V:	19.36 Ah

Figure 3.22

This compilation of data leaves adequate room for error by accounting for the maximum values incurred by each of the components, assuming that all components will be active at all times, and using the upper end of the design constraint which states that the battery life of the system must be 1-2 hours between charges. Thus, the final calculation of 20 Ah required capacity should be sufficient to meet the needs of this system. Several battery options meeting the 12V required voltage specification are detailed in the table below.

Battery	UPLUS LP12-20 12 Volt 20Ah Rechargeable AGM Battery	Tenergy NiMH Battery Pack 12V 2000mAh	Feuruetc 12V 12Ah Deep Cycle LiFePO4 Battery	Ovonic 100C 11.1V 8000mAh 3S LiPo Battery
Type	SLA	NiMH	LiFePO4	LiPo
Capacity	20 Ah	2Ah	12 Ah	8Ah
Quantity Needed	1	10	2	1
Cost per Unit	$46	$22	$40	$46
Total Cost	$46	$220	$80	$46
Weight per Unit	12.5 lb	9 oz	3.5 lb	1.1 lb
Total Weight	12.5 lb	6 lb	7 lb	1.1 lb

Figure 3.23

Due to the optimal tradeoff between cost, capacity and weight, the Ovonic 100C 11.1V 8000mAh 3S LiPo Battery will be utilized for this project.

4. Related Standards and Realistic Design Constraints

When designing a product, there is more to consider than just the technical and financial objectives needed to make the product viable. Also important are design standards: established sets of rules and criteria that provide guidelines for safe and effective operations. Standards generally are not required to follow by law, but prove to be beneficial in terms of interoperability between similar and adjacent systems, and can also be cited by a product manufacturer to show that the product is in compliance with certain safety, quality, or other standards, which increases marketability and acceptance among the public and industry. Organizations like the American National Standard Institute and the IEEE Standards Association help develop and pass standards for a variety of products and systems. Design constraints are less official considerations that must be taken into consideration when designing a product in order to maximize acceptance and marketability. These include economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability factors that must be balanced in order to create a product that is overall realistic, safe, desirable, and beneficial to consumers.

4.1 Standards

The following standards have been found to be applicable to the FoodieRover project. The current standards that are applicable to the project may change in the future as development continues. These standards ensure food safety, maintaining quality, follow regulations, and build good trust in the application of automations and food delivery robots.

<u>SAE Levels of Driving Automation</u>

According to the SAE J 3016-2021 standard, terms such as “driverless cars” and “autonomous vehicles” are inaccurate when defining driving automation systems which should be referred to as such. Automation refers to the use of electric or mechanical devices to replace human labor which is appropriate to use and can avoid misinformation. In order to understand and apply SAE levels of driving automation, dynamic driving task (DDT), operational design domain (ODD), and automated driving system (ADS) must be understood. They are defined respectively as “all the real-time operational and tactile functions required to operate a vehicle in on-road traffic”, “the specific conditions under which a given driving automation system or feature thereof is designed to function”, and the hardware and software responsible of DDT.

As the levels increase so does the degree of automation. Levels 0 to 2 expect the driver to be in charge of DDT whilst in levels 3 to 5 the ADS performs all of the DDT. The level of interest regarding a food delivery robot pertains to level 4 which is called High Driving Automation. It is sustained and ODD-specific with ADS in charge of DDT as mentioned. It is imperative that the user has no expectation of intervening with the system. The next level has a similar definition with the exception of having no condition to function which means it is free from geofencing.

Standard for Autonomous Robotics (AuR) Ontology

Standard IEEE P1872.2 which is an extension of IEEE 1872-2015. It expands on CORA ontology to include principles and concepts commonly used in the field of Autonomous Robotics. The knowledge domain required for building an autonomous system is defined in this standard for robots that are capable of operating in various unstructured environments. Additionally, it offers a standardized way to represent the architecture of autonomous robotics systems in different fields such as aerial, ground, surface, underwater, and space robotics. It enables a clear identification of fundamental hardware and software components needed for robots to be autonomous. It allows robots to perform tasks in unstructured environments without the constant interference of direct human supervision.

Standard Test Method for Navigation: Defined Area

The standard test ASTM F3244-17 for unmanned ground vehicles considers the sites of operation that may or not be defined and may or not be structured. The testing results should statistically describe how the vehicle is able to traverse the commanded path. It considers if the vehicle is able or not to deviate from the path and if it takes input from its local environment to its navigation method. In other words, the capabilities of the unmanned ground vehicle to adapt its navigation given an environment will objectively be determined by the result of the test.

Standard Terminology for Driverless Automatic Guided Industrial Vehicles

Based on ASTM F3200-20a, it covers terminology related to unmanned, ground, and industrial vehicles. Its purpose is to facilitate communication for research, design, deployment, and use of unmanned ground vehicles.

Robots and Robotic Devices

ISO 10218-1:2011 standards are a set of requirements and guidelines for the inherent safe design, protective measures, and information for the use of industrial robots. Basic hazards associated with robots are described as well as requirements to minimize or completely eliminate the risks associated with the hazards. ISO 10218 does not apply to non-industrial robots but the safety principles can be utilized for other robots.

Safety of Machinery

ISO 13849-1:2015 provides further safety requirements and guidance on principles based on the design with the integration of safety-related parts of control systems which includes the design of software. It specifies characteristics that include the performance level required for carrying out safety actions. It applies to high demand and continuous mode regardless of the type of technology used.

AGV Communication Interface

VDA 5050 V 1.1 document describes the communication interface for order and status data exchange from a central master control and AGV for intralogistics processes.

Automated Mobile Platforms (AMPs)

Set of requirements in accordance with ANSI/CAN/UL 3100 related to battery-operated mobile platforms with and without payload. It covers both indoor and outdoor devices in commercial or industrial environments. The batteries are either lead acid or lithium-based and may be rechargeable with a system on or off-board the device.

Standard Practice for Recording Environmental Effects for Utilization with A-UGV Test Methods

ASTM F3218-19 standard provides a means to record environmental conditions that may influence the performance of unmanned ground vehicles. Those conditions include lighting, external sensor emission, temperature, ground surface, air quality, humidity, and electrical interference.

Food Safety Standard: FDA Food Code

The Food Code serves as a blueprint for maintaining public health and ensuring that food remains untainted and accurately portrayed by the time it is given to the consumer. It embodies a consistent set of rules that address the safety and safeguarding of food available in both retail and food service settings.

4.2 Design Impacts of Related Standards

This section takes into account the standards introduced and discussed in the previous section, explaining their influence on the design of FoodieRover. While certain standards have a greater impact than others, all are carefully considered in shaping the final product that FoodieRover will become.

Driving Automation

In order to attain the targeted Level 4 automation for the FoodieRover system, it is imperative to ensure strict adherence to the precise definition of automation. Specifically, this involves the complete replacement of human labor in the context of food delivery tasks. The system must proficiently execute a series of core functions, encompassing order reception, secure ground transportation, and timely, precise delivery. Each of these functions must be performed autonomously, devoid of human intervention.

Moreover, to achieve a state of High Driving Automation, or Level 4 automation, it is imperative that the automated driving system undergoes a comprehensive design overhaul. This entails the seamless execution of all dynamic driving tasks within the designated operational design domain. The automated driving system comprises both software and hardware components, meticulously orchestrated to collaboratively execute the full spectrum of dynamic driving tasks, in addition to other functions as elaborated subsequently.

A critical aspect of this endeavor is the sustainable design of the system. The failure to meet any of the aforementioned design prerequisites will result in a degradation of the autonomy level, a situation incongruent with the stipulated requirements and objectives of the FoodieRover project.

Autonomous Robotics Ontology

The standard lays out fundamental principles and common concepts used in the realm of autonomous robotics, while also providing a structured way to represent the architecture of such systems. In terms of design implications, it offers a clear understanding of the hardware and software components involved in creating autonomous robots. The standard includes clear definitions for various aspects and organizes them into hierarchical design frameworks that govern how autonomous robots, like the FoodieRover, behave and operate.

Test Method for Navigation: Defined Area

The necessity to objectively assess FoodieRover's capacity to adapt its navigation capabilities within a specific environment is a paramount consideration. The statistical outcomes derived from the conducted evaluation will serve as a metric to gauge the efficacy of the design and, should any deficiencies be identified, provide guidance for targeted enhancements. It is imperative to subject FoodieRover to comprehensive performance testing, ensuring alignment with the established benchmarks set forth by the evaluation.

Terminology for Driverless Automatic Guided Industrial Vehicles

Employing standardized terminology serves to enhance effective communication in the process of designing FoodieRover, as well as in the contexts of research and deployment. Furthermore, it is important to emphasize that when evaluating the design, the terminology established by the standard should be consistently applied.

Robots and Robotic Devices

Incorporating safety considerations and preemptive protective measures is an indispensable facet of the design process for any robotic device. In the case of FoodieRover, a comprehensive assessment of potential hazards is imperative, encompassing mechanical, electrical, thermal, material, and environmental risks.

In the domain of mechanical hazards, FoodieRover must navigate its surroundings cautiously, prioritizing the safety of both pedestrians and itself. The design will be engineered to ensure this safety. Internally, specialized compartments will be integrated into the design to safeguard the system's components from harm. Stringent measures will be implemented to eliminate any risk of fire hazards, including continuous monitoring and temperature testing to ensure that component temperatures remain below the ignition threshold. The selection of nonflammable materials will further bolster fire prevention.

Additionally, adherence to established standards is paramount to mitigate environmental hazards. It is a core objective of FoodieRover to serve as an environmentally friendly alternative for food delivery. Consequently, the design will rigorously adhere to relevant standards to ensure minimal environmental impact and contribute to the achievement of this goal.

Safety of Machinery

The formal articulation of safety aspects within the design is expounded upon in the provisions of this standard. This involves the integration of safety components as an integral facet of the control system. This approach duly acknowledges the elevated demand rate anticipated for the FoodieRover system, a critical consideration essential to realizing its intended design functionality objectives.

AGV Communication Interface

The design of the software pertaining to the communication interface will be directly impacted by the stipulations and guidelines delineated in the relevant standard.

Automated Mobile Platforms

The power source holds a pivotal role in the design of FoodieRover. This crucial component of the design process will be subject to the guidance and specifications outlined in the relevant standard, thus ensuring its optimal integration and performance within the system.

Recording Environmental Effects

FoodieRover is expected to operate under diverse environmental conditions, including slippery surfaces, varying temperatures, and other challenges. The design process will be subject to the influence of established standards that record the impact of the environment on the robot's performance, contingent upon the outcomes of testing procedures. This influence will help shape the design to ensure optimal performance across these various conditions.

Food Safety Standard

In light of FoodieRover's involvement in food-related operations, the paramount consideration of food safety is intrinsic to its design. Preserving the integrity and safety of the food is of utmost importance, ensuring strict adherence to the standards and expectations set forth by the respective restaurants or businesses regarding food presentation to customers. The design process will be significantly influenced by these imperatives.

4.3 Economic and Time Constraints

Economic and time constraints serve as foundational elements in the management of any project. In the specific context of FoodieRover, where external sponsorship is not available, and the project timeline is limited to two academic semesters, each with distinct milestones—one culminating in the finalization of a comprehensive 120-page report and the other marked by the realization of conceptual ideas—our project team recognizes the paramount importance of these constraints.

Economic constraints can be further categorized into five key aspects: budget restrictions, resource allocation, cost estimation, return on investment (ROI) analysis, and opportunity cost assessment. The first among these, budget restrictions, pertains directly to the finite pool of financial resources at our disposal. FoodieRover comprises an array of distinct components, each necessitating sourcing and procurement. For certain components, such as the frame, our intent is to acquire assembly packages, while other components will require individual sourcing efforts.

Regarding resource allocation, it is imperative to recognize that resources encompass more than just the physical components and materials required. They also extend to human resources, such as labor, and access to specialized equipment. In the context of this project, it is noteworthy that none of the team members have an expectation of financial compensation, thus alleviating any financial impact on the budget.

In terms of equipment, one such advantage is that the University of Central Florida extends its facilities to students for the purpose of conducting testing and experimentation. As a result, should the need arise for the evaluation of circuit designs or the utilization of specialized equipment, the university provides accessible resources at no additional cost. This includes a wealth of tools and instruments such as oscilloscopes, multimeters, and function generators available for students' use within designated laboratories. This strategic resource allocation minimizes the financial burden associated with equipment procurement and underscores the advantage of being able to leverage the university's available resources.

The facet of cost estimation assumes a pivotal role within our project's strategic framework. In light of the team's primary commitment during the initial semester, which entails the composition of a comprehensive research report, an integral component of this endeavor involves the meticulous pricing evaluation of various project components. This comprehensive assessment involves the thorough examination of multiple models of each component, differentiated by varying price points. Each model's attributes, encompassing both advantages and drawbacks, are diligently scrutinized to discern the optimal selection for integration into our design. The precision and thoroughness of this cost estimation process are integral to the prudent allocation of project resources and fiscal responsibility.

Regarding the matter of Return on Investment (ROI), it should be made clear that the core motivation behind this project does not center around profit generation. However, in the event that the University of Central Florida expresses substantial interest in the practical applications and functionalities of FoodieRover, the consideration of patent filing and potential commercialization may arise.

Drawing from precedent set by analogous projects, notably exemplified by KiwiBots, there is a clear indication that the applications and services offered by FoodieRover possess inherent commercial viability. This observation underscores the potential for the project to yield substantial returns on investment, should the circumstances and market dynamics align favorably in the future.

The concept of opportunity cost is applicable in the context of the Senior Design process, particularly concerning the projects that were not selected for pursuit. Among the array of project ideas considered, our team exhibited a discerning approach in the selection of FoodieRover. This choice emanated from our collective confidence in its potential, as it offered a challenge commensurate with our expertise and dedication.

FoodieRover, as the chosen project, provides a unique opportunity to integrate our diverse disciplines encompassing computer and electrical engineering. Moreover, it strategically aligns with the burgeoning industry of autonomous vehicles, thereby positioning us at the vanguard of technological innovation. This strategic selection, driven by the pursuit of excellence, underscores the deliberate consideration of opportunity cost within the framework of the Senior Design process.

Conversely, time constraints serve as instrumental factors in delineating the organizational structure and sequential progression of our project. The salient elements encompassed within the framework of time constraints encompass the project schedule, delineation of project phases, critical path analysis, consideration of resource availability, adherence to project deadlines, and vigilant scope control.

As previously articulated, the FoodieRover project is bound by a finite timeline extending over two academic semesters, equivalent to 32 weeks. The initial semester is distinctly allocated to the research phase, characterized by the imposition of multiple intermediary milestones. For instance, within the initial fortnight, our project idea was required to be unequivocally specified, while within the inaugural month, a concise written exposition of our concept needed to be substantiated, enabling a thorough evaluation of its feasibility.

Notably, the incorporation of midway deadlines serves as a pivotal mechanism to foster and gauge steady progress within our team, assiduously steering us toward the overarching objective of producing a comprehensive 120-page report by the culmination of the semester. These time constraints, carefully structured and adhered to, shape the cadence and momentum of our project, directing our efforts towards the attainment of well-defined milestones and the ultimate realization of project goals.

4.4 Environmental, Social, and Political Constraints

Environmental, social, and political constraints must also be considered when designing our delivery system. These constraints deal mainly with the interaction between the system and the environment, the acceptability of the system and its features by general society, and legal constraints that must be carefully followed to ensure the design does not conflict with any laws or other regulations.

I. Environmental Constraints:

Environmental constraints are those related to the system's potential impact on its surrounding environment and the related design factors which must be considered to limit negative effects. Our system should be designed so as to not produce any external waste that may end up in the surrounding environment as pollution. The delivery system should be fully enclosed so that the food being delivered does not shed lightweight objects, such as napkins or straws, when faced with wind. Additionally, the system should operate on rechargeable battery power. This means it will not produce toxic air pollution that comes with combustion engines or similar powering systems, and it will not create excessive waste over time with the disposal of depleted battery packs. Low energy consumption is ideal for longevity purposes, but an added benefit is that less electric power is used to recharge the batteries over time, which also has a positive environmental impact since less power must be produced from potentially polluting fossil fuel power plants to service the system. Given this, the energy efficiency of the system should be kept as high as possible by choosing efficient power sources and motors and writing concise software to reduce processing power. Another important environmental consideration is the existence of certain protected or sensitive areas - such as where an endangered species of plant or animal is present. If applied in a region where these areas exist, the serviceable area should be carefully designed as to prevent the robot from moving through these sensitive areas and disturbing them.

II. Social Constraints:

Social factors also impact the design process, and care must be taken to ensure that the product does not have any inherent bias towards any specific group of people, for example. It is important to consider not just technical factors, but also others which may affect the acceptability or desirability of society in general if we want to design a product which can be used by the general public. For example, the system should be very simple and intuitive to use from the consumer's perspective. In a large-scale application of the system, delivery services should be available to a large range of areas, and serviceable areas should not be determined by any potentially discriminatory factors such as race, age, or economic demographics, but rather by a distance radius (and other reasonable, neutral factors such as terrain and presence of buildings in the area). The application in which users will interact with the system should have the capability to operate in English and Spanish, and potentially any other commonly-spoken languages in the service area.

Another social consideration is that some people tend to be wary of robots in general, especially in terms of giving personal information. Because of this, it should be clear that the system does not store or inappropriately use any personal information required, such as name and delivery address. Some form of data encryption should also be used with any stored data in the application to ensure security of users' personal information in the event of a database hack or information leak.

III. Political Constraints:

Political design constraints largely involve compliance with any existing laws, regulations, or standards that may be in place regarding the system. Examples of legal constraints relating to our design include privacy laws, food handling laws, intellectual property laws, and accessibility laws. Privacy and security laws require that consent must be gained from a user in order to store any personal information, and users have the right to to request any records which are stored in the application's database. This can be addressed by either not requiring personal information from the users (for example, the system can require only a general delivery area instead of a specific address, and no name, phone number, etc. can be requested), or any data which is required will be obtained after consent to a privacy policy and will be encrypted for security. In addition to the FDA food code, a general constraint regarding food delivery services is that the food should be kept at a reasonable temperature, in a sanitary container, and not handled by more people than necessary while delivering. The design of the container will keep the food insulated, the food will be packaged as any to-go order from the vendor before setting it in the container to keep a layer of protection between the food and the container, and the locking mechanism of the system will prevent any unwanted people from tampering with the food during delivery. Additionally, related patents, copyrights, and trademarks may exist which must be researched and understood in depth to prevent any potential infringement on an existing design or other piece of intellectual property. The delivery system must not include features which may be considered too similar to any existing design protected by these forms of legal ownership to prevent lawsuits and reputational damage. Several other required or strongly suggested legal constraints have been discussed in the Standards section. In addition to legal constraints, another factor which also falls in the political category is discrimination, which has been addressed in the social constraints section and care will be taken to ensure the system complies with accessibility and equality guidelines. The highly non-interactive design of the system ensures that accessibility will not be an issue, provided that the customer is in the serviceable area of the system and the order is placed to a location which they are able to physically access.

4.5 Ethical, Health, and Safety Constraints

FoodieRover is an innovative autonomous food delivery robot designed to navigate through various environments to deliver food items to consumers. The integration of advanced technology such as ultrasonic sensors, cameras, and computer vision enhances its obstacle detection capabilities, ensuring efficient and safe navigation. Additionally, the implementation of an application interface and API capabilities facilitates seamless user interaction and connectivity. This portion emphasizes the importance of prioritizing user safety, data privacy, and ethical considerations throughout the entire process.

I. Ethical Considerations:

In the context of ethical considerations, one crucial aspect is to ensure data privacy by securely storing and protecting all user data, including personal information and delivery details, from unauthorized access. Transparency is equally essential, and users should be provided with clear information regarding the data collected, its purpose, and how it will be used to ensure transparency and build trust. Fair access to the FoodieRover service for all consumers and producers, regardless of their location or socioeconomic status, must be guaranteed. Additionally, it is essential to develop algorithms and decision-making processes that prioritize ethical choices, such as avoiding discriminatory practices or biased decision-making during food delivery.

II. Health Constraints:

To uphold health constraints, rigorous food safety protocols must be implemented to ensure that the delivered food remains fresh and uncontaminated during transportation. High standards of cleanliness and hygiene should be maintained within the container to prevent any potential health risks or contamination of food items. Furthermore, measures to prevent cross-contamination and handle different types of food items, taking into account various allergens and dietary restrictions, should be integrated to ensure the safety of consumers with specific dietary needs.

III. Safety Constraints:

In terms of safety constraints, the accurate and reliable functioning of the ultrasonic sensors and cameras is imperative for effective obstacle detection and safe navigation in various environments. It is essential to develop emergency protocols and fail-safe mechanisms to address potential malfunctions, accidents, or unexpected situations during operation. Efficient power management systems should be implemented to maintain safe power consumption levels, prevent overheating, and extend the lifespan of the battery and electrical components. Moreover, the design of the container and the overall structure of FoodieRover should prioritize the structural integrity, enabling it to withstand external impacts, prevent spillage, and secure the internal electrical components to prevent any safety hazards or damage to the delivered food.

FoodieRover's development and operation must adhere to stringent ethical, health, and safety constraints to ensure the well-being of users, the safe delivery of food items, and the seamless operation of the autonomous system. By prioritizing these constraints, FoodieRover can establish itself as a reliable and ethical solution for efficient food delivery services while maintaining the highest standards of user safety and satisfaction.

4.6 Manufacturability and Sustainability Constraints

Manufacturability and Sustainability Constraints for FoodieRover

FoodieRover, an advanced autonomous food delivery robot, not only aims to revolutionize the food delivery industry but also seeks to set a new standard for manufacturing efficiency and environmental sustainability. This paper delves into the critical aspects of manufacturability and sustainability that are essential for the successful development and operation of FoodieRover. Emphasizing streamlined production processes, cost-effective manufacturing, and eco-friendly practices, the paper highlights the importance of ensuring that FoodieRover aligns with both efficient production requirements and sustainable principles to minimize its environmental impact and promote responsible manufacturing practices.

I. Manufacturability Constraints:

Efficient and cost-effective production processes are imperative for the successful deployment of FoodieRover. To address manufacturability constraints, the design and development phase should emphasize the selection of materials and components that are readily available and easily manufacturable. Implementing a robust supply chain management system is crucial to ensure the timely and consistent availability of necessary materials without disruptions or delays. Furthermore, the design of FoodieRover should prioritize modularity, allowing for the easy assembly, disassembly, maintenance, and potential upgrades of various components. This approach will not only streamline the production process but also minimize downtime and overall operational costs. Additionally, the standardization of manufacturing processes and components will enable efficient scalability, facilitating increased production to meet growing market demands without compromising quality or efficiency.

II. Sustainability Constraints:

In a rapidly evolving world focused on environmental conservation, the sustainable development and operation of FoodieRover are of paramount importance. Incorporating sustainable practices into the manufacturing process is essential to minimize the environmental impact of the robot's production. This can be achieved by emphasizing the use of sustainable and recyclable materials in the construction of FoodieRover, thereby reducing its carbon footprint and ensuring a more environmentally responsible approach to production. Moreover, integrating energy-efficient components and implementing intelligent power management systems will significantly reduce the overall energy consumption of FoodieRover during operation. Adhering to proper recycling protocols and promoting the reuse of materials in the disposal process will further minimize waste generation and contribute to the promotion of a circular economy model. By embracing sustainable manufacturing practices, FoodieRover can serve as a pioneering example of how technological innovation and environmental responsibility can coexist, setting a precedent for the integration of sustainability into future robotic designs.

FoodieRover's successful integration of manufacturability and sustainability constraints not only ensures the efficient production and operation of the autonomous food delivery robot but also sets a benchmark for responsible manufacturing practices within the robotics industry. By prioritizing cost-effective production processes, streamlined manufacturing, and the use of eco-friendly materials, FoodieRover can be manufactured in a manner that minimizes its environmental impact and promotes sustainable practices. As technological advancements continue to shape the future of robotics, the adherence to these constraints will be crucial in ensuring that future innovations align with both manufacturing efficiency and environmental sustainability, contributing to a more responsible and sustainable future for the robotics industry.

5. Comparison of Chat GPT or Similar Platform

Introduction

The fundamental idea behind machine learning and artificial intelligence (AI) is that an entity that is provided with enough information collected from an influx of questions will be able to generate an original response based off of this research. There exists the comparison that this is a very similar model to how humans collect information and are therefore capable of creating original responses to new questions. However, as with any innovation developed within the exponential growth of the technological industry, there exists pitfalls that are aligned with the use of artificial intelligence. The characteristics of platforms such as ChatGPT and its peers will be explored based on their assets, advantages, and disadvantages.

What is AI?

De Spiegeleire et al (2017) describes artificial intelligence as the ability of a machine being able to mimic the process of human intelligence, reasoning and interpretation of natural language (p. 28). The concept of artificial intelligence was initially brought to academia with Alan Turing's work "Computer Machinery and Intelligence," where the concept of machine intelligence was introduced. Within the decade of the 1950s, the concept was explored in the form of simple and algorithmic games being implemented on computers. Just two years after the birth of machine intelligence, Arthur Samuel would develop the first game of checkers, where the machine was able to project its own strategies. Naturally, over the next few decades there would be interest in the subject of artificial intelligence, however the boom for interest would occur in the 1980s. While public use was not available, massive amounts of funding was funneled into the research of artificial intelligence. This boom in the interest of artificial intelligence would parallel another technological advancement - the internet ("What is the history of artificial intelligence (AI)?", Tableau, 2023.)

Despite the growth of artificial intelligence, one may describe the prior presence as the "backend" of operations. Despite the clear patterns in advertising and the exponential production of simple tasks and algorithms, most of the world was not focused on the topic of artificial intelligence. It was not until the late 2010s that artificial intelligence seemed to transition from science-fiction to reality. Art from generative AI seemed to suddenly appear in artist galleries and students seemed to suddenly be able to write a paper in less than fifteen minutes. Applications such as Grammarly were able to rewrite an essay within fractions of a second. The presence of artificial intelligence was simply a background noise until the november of two-thousand twenty-two, when openAI abruptly shined a light on the concept and opened up one of the most complex discussions on ethics and privacy.

Comparisons Between ChatGPT and Its Peers

While the culmination of artificial intelligence had shown significant growth in the past, ChatGPT seemed to be the catalyst for much discussion. ChatGPT is a conversational artificial intelligence model that does not just respond to a question, but manages to carry the concept on. Scharre (2018) noted that one of the pitfalls of artificial intelligence was the inability to jump from one task to a similar task, describing it as "narrow" (p. 8.) ChatGPT manages to create transitions between different topics, which shows quite a bit of growth in just four years.

While simplified services similar to ChatGPT have existed in the past, the format has been developed by many companies within the years of two-thousand twenty-three. The three services that were compared were ChatGPT by openAI, Bard by Google, and ChatSonic by Y Combinator.

A test was done to create a direct comparison between the three. A script of eleven questions was presented to the services. The questions varied between logistics, ethics, and fiscal attributes of the project. Prior to submitting the questions, a description of the device was cultivated to create a comparison to the depth in which the AI platforms would describe the planned design. The description was as follows,

> "*Food delivery robot will move on ground, will look like a tank, have ultrasonic sensors, GPS, locking mechanism, a communication module*"

For the first request, "*Design the most efficient self-automated food delivery machine that could find and deliver to addresses within a designated region,*" was presented to all three platforms. Each of the platforms responded with an itemized list that included attributes such as hardware, software and delivery. ChatGPT was careful to note that the software would benefit from being powered by AI. Between the three, ChatGPT went the most in depth with its responses, breaking down into subheadings within the answer to specify components down to the type of battery the delivery device would benefit from. Bard had a concise description but no detail was given aside from the efficiency of the device. Bard mentions optimizations in the form of algorithms in the software and automated unloading in the design. Bard made no mention of safety, however ChatGPT does mention that the device should meet regulation. More interestingly, ChatSonic actually addresses the security of the device as it travels, going as far as to suggest that the device must have cameras, QR scans to ensure that the correct party has access to the food and an emergency stop system. This divergence suggests growth in AI that shows an ability to reason and consider real threats to the device. Such a form of security is generally cultivated in experience between humans, for the AI to consider this shows a good amount of understanding from the data it has collected so far.

For the next question, "*How much would the materials cost for this device?*." Another diversion is relatively interesting, as ChatGPT initially designed the device with a robot arm, which Bard and ChatSonic did not account for. Bard mentioned an unloading system, however ChatGPT was the only platform to add an appendage to the device. As a matter of fact, this causes a slight schism between the results. Therefore, when the question of budget was offered, ChatGPT gave a slightly different answer from the rest. Once again, all three gave an itemized list, and described the cost by part.. While Bard and ChatGPT were relatively specific in their responses, ranging from thousands to tens of thousands of dollars, ChatSonic was vague, and mostly mentioned that those parts would be more expensive.

When asked how long it would take to develop the device, Bard gave it around nine to ten months, while ChatGPT estimated several years. ChatSonic did not give a definite time frame. The ethics and safety of the device were addressed in the next question. While Bard and ChatGPT gave an itemized list, ChatSonic simply gave a general response. ChatGPT and Bard reference the concern of privacy and security as well as the danger of contamination of the food. ChatGPT did, however, mention the safety of pedestrians. The request was to show similar products to the food delivery device. Naturally, all three gave responses such as Amazon Scout and Kibo Robot, however ChatGPT was the only one that did not provide images, as its interface was not integrated in the same way as Bard.

ChatGPT is primarily for conversational use. In a sense, the user submits a question and the conversation cascades in such a way that ChatGPT responds with reference to the previous statement. ChatGPT requires an account log-in prior to using the service. Along with logging in, ChatGPT has tiers based on subscriptions. While the benefits of the subscription are not significantly different from the basic access, it does guarantee connectivity even in the moments of heavy site traffic. One of the limitations is that ChatGPT refrains from using images.

Bard is powered by Google, and naturally, has the largest pool of information to pull from. Due to the vast amount of information, Bard does take slightly longer to respond than ChatGPT and ChatSonic. One major advantage is that Google has integrated their AI into all of their products, such as the search engine. While the entirety of the features are still within a focus group, the progress and additional information that is being fed to Bard is insurmountable. However, a major drawback is actually that the development is still relatively early for the service. While the service works very well and only slightly lacks the transitional finesse of ChatGPT, it still has many phases to go through before it is considered a complete product.

Between the three platforms named for comparison, ChatSonic has the most streamlined use of conversational AI as the initial was for an automated creative writer. Despite not having the greatest well of information in comparison to Bard, ChatSonic has a surprisingly advanced interface. When a question is thrown at ChatSonic, it actually denotes references to each of its responses, it even allows the users to cite the information through just a click of a button. However, the most surprising feature is the fact that the generative AI is embedded in the chat AI, as both Bard and ChatGPT are incapable of this feature at the time. A drawback is that ChatSonic draws from a smaller pool of information compared to Bard and ChatGPT as most of the information it is collecting does not have the same entropy a service like Bard does.

The three share a limited inability to form truly original thoughts. While certain aspects, such as the robot arm, are really interesting, it is of note that most of this information is likely coming from the bots referenced in the final question. No matter how the prompt is augmented, the reference will still have been of an object that existed, the simple difference would be that certain specifications would be added post-discussion. Another issue is the depth of these questions, answers and the realism of the responses. In a sense, in this instance, the questions used would need to be present with reference to the actual materials available, the data used, and the anticipated region of use to give a truly useful answer. While ChatGPT can reference a question, an image or even a text, it cannot truly understand the concept of a small scaled project that is a testing stage. It also cannot give information or ideas that did not already exist. It is also of note, that the questions that would, potentially, yield an original answer would have to be posed in such a specific way that even the platform would likely be unable to ask.

The most concerning consequence of using the three platforms resides in ethics. The concern over the source of the data is important for even the smallest bit of advice taken from these machines.

The Good

The descriptions developed by the AI platforms helped to give a better understanding of the scope of the project. While asking about safety concerns, some aspects were not considered prior to the responses. Concepts such as the safety of pedestrians and the security of the device were realized based on the responses. One might ration that a company often develops concerns based on consumers and prior products. While prior products were reviewed and analyzed, there were still minor innovations that were beneficial in the process of development. For example, one of the concepts that was brought up was inventory management. While it would be possible to acquire a device such as Amazon Scout, it would not be practical to purchase and does not deliver in the current region. A manual could give a good understanding of the inventory system, it is still important to consider the processes, however the occurrence of thought would not necessarily present itself without direct interaction with the device.

While working with the compared platforms, AdobeFirefly was employed for an extra aspect. It could be noted that the design of a project often relies on simulations of a product. Without a designer, it could be difficult to visualize what the product will be without a true reference point. While similar products exist, it can be helpful to have a personal design. For this reason, a similar prompt was run through the generative AI to see what ideas could be cultivated. After several changes to the prompt, the following simulations were generated.



AdobeFirefly (2023)

Naturally, an exact product of such specification is not entirely within range for the project, however it is an interesting concept to have in mind when developing. The prompt for this section was significantly more difficult than that of ChatGPT. This is likely due to the fact that Chat GPT is building information based on a prior response while the AdobeFirefly is only going based off of a single shot.

The platforms such as ChatGPT also create a good idea of what is on the market and how useful something can be to society. It is of use to understand which companies are investing in something and how they are doing it. While a simply search could yield similar answers, a direct comparison to an idea and the similar products that already exist might be of great use to the designer.

<u>The Bad</u>

The scope was a relatively double-edged sword. While the new concepts and ideas were presented, some were not feasible and others were simply not possible within the scope of the course. For example, the robot arm proposed by ChatGPT would likely require significantly more development than is possible in the timeframe offered in the Senior Design course. Especially considering the safety of anyone who is using the device. The mechanics of appendages on a robot are quite delicate and require a good amount of fine-tuning. To consider such a task with both the sensors of the main cabin of the device as well as a robot arm would simply be approaching the line of scope-creep.

Naturally, the consequence of a well of knowledge is also the validity of the responses handed by the platforms. While it is necessary for human correction for the responses before their use, it is important to note that assumptions are made on the part of the platform that might not be accounted for in the development of the project. For example, Google estimated that the design for a food delivery robot would cost around one-hundred eighty thousand dollars and two-hundred fifty thousand dollars. This price is quite steep for a group of students. The simple fact is that the materials that will be used are not going to align with what Google anticipates the user will be using. In a sense, the difference between a group of students and a company is not embedded in the response. While this could be seen as an issue in the question, it could also be seen as a flaw in the platform not being adapted to give follow-up responses for clarification.

One might also note that the information, again, is coming from somewhere. As stated previously, the AI is not creating a brand new thought. A natural concern is not only the ethics of the data collected but the risk of creating a product that is far too close to an existing item on the market. While the questions are relatively general, it is possible for a proposal of design to be an aspect from a previous bot that was either patented or a niche of the company. For example a certain tread of tired or a certain alignment of parts might be exclusive to a company that has a design that was collected by the platforms.

6. Hardware Design

This section will discuss the hardware design of the FoodieRover system. In a system with many sophisticated hardware components such as this one, extreme care must be taken in order to integrate all components together in a sensible and functional manner so that each item is able to perform to the best of its ability and complete the necessary tasks without hindering other components of the system. For that reason, it is imperative to create a detailed hardware design that considers each component's features, functionalities, and interaction capabilities with other components to develop a robust and fully integrated system design. Given that our product is being developed to carry out many tasks simultaneously, with each task relying on different hardware components, the overall hardware design has been broken down into several subsystems based on function. These subsystems will be described in detail in this chapter, including the components involved in each one, the purpose of each component as part of the subsystem, the connections and compatibilities between components, the control mechanism of the subsystem, and any future considerations or possible enhancements that can be made as the project progresses further or for similar projects in the future.

6.1 Subsystem Block Diagrams

Navigation Subsystem

The Navigation Subsystem serves the critical purpose of allowing the robot to determine its current location in time. This function is imperative if we are to provide the robot with self-navigation capabilities. Though the system may have control over the movement of the robot via the Motor Subsystem, there must be a frame of reference of its current location if it is to determine the correct direction in which to move. This subsystem includes the magnetometer and the GPS module, each connected independently to the centralized microprocessor in order to pass information to and from the unit for analysis and refinement of movement.

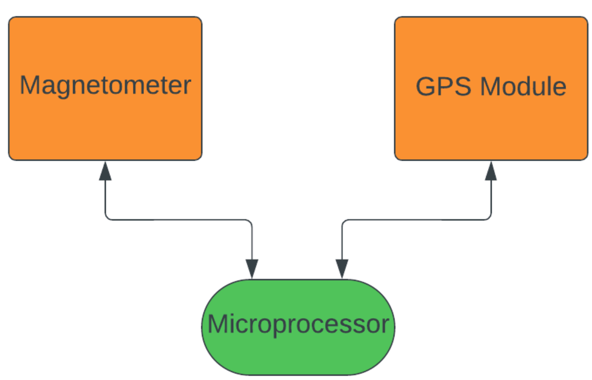


Figure 6.1

Components Overview:

HMC5883L Magnetometer/Digital Compass (x1):

- This 3-axis digital compass unit utilizing an internal magnetometer is designed to sense and measure magnetic fields, contributing to the system the ability to determine its spatial orientation relative to the global frame of the North/South/East/West cardinal directions.
- Accurate realization of the robot's orientation is necessary to accomplish the functions of directional navigation and is vital information, in addition to the GPS information, for the processor to be able to determine the correct direction in which to move at any given time.

L76K GPS Module (x1):

- The Global Positioning System module uses satellite communication to geographically triangulate its spatial position in the global frame of latitude and longitude coordinates.
- This positioning information is imperative to the Navigation Subsystem as the current location must be compared with the final location information in order to determine the spatial error between the two, which allows the processing unit to determine the direction in which to drive movement.

Raspberry Pi 4 Model B Microprocessor (x1):

- As the "brain" of the Navigation Subsystem (and the entire system itself), the microprocessor unit is the crucial hardware component which allows the detailed information received from the Compass and GPS Module to be analyzed and applied correctly for the desired result.
- This centralized unit triggers communication with the Compass and GPS Module to obtain location and orientation details which will then be applied to carefully designed software that examines this input, compares current location with the known desired end location, and uses the error vector to orient and mobilize the robot in the correct direction via the Motor Subsystem components.

Integration and Functionality:

Interconnection:

- The HMC5883L Compass Module requires a power supply of 2.16-3.6V and communicates information via I2C protocol. Therefore, the SDA and SCL pins of the compass module will be connected to the SDA and SCL pins of the Raspberry Pi Microprocessor for proper configuration of I2C communication, and a 3.3 Volt input voltage will be supplied to the unit's VDD pin.
- The L76K GPS Module utilizes the UART communication protocol, however, so it will instead be wired to the TXD0 and RXD0 pins of the Microprocessor to allow for compatible information relay between the two components. The GPS

VDD pin will be connected to a 5 Volt input voltage node, satisfying the input voltage requirement.

Control and Precision:

- Both the Compass Module and the GPS Module are wired separately to the central processing unit which provides the control mechanism for each of these peripherals. The software run by the microprocessor will trigger communication with the Compass Module for orientation data and with the GPS Module for location data frequently as it consistently checks and reacts to this information to allow for accurate navigation.
- The Compass Module provides a resolution of 1-4 milli-gauss, providing sufficient accuracy to the proposed application. The GPS provides a positional accuracy of 2.0 meters Circular Error Probable (CEP); this will be considered when creating the navigation software as leave room for potential error in measurement.

Performance and Future Enhancements:

- Given that the Magnetometer determines orientation based on readings of the magnetic field, it is sensitive to interference from nearby current-generating devices that cause fluctuations in their surrounding magnetic fields. Consequently, this device should be placed as far away as possible from heavily conducting areas on the robot structure.
- Potential enhancements that may be made to the Navigation Subsystem include obtaining more sophisticated components with higher accuracy ratings, and optimizing the frequency of communication triggering of the modules for minimum overall power draw.

Motor Subsystem

The Motor Subsystem of the FoodieRover stands as a cornerstone in enabling precise movement and navigation essential for efficient food delivery operations. This subsystem consists of four DC motors integrated and controlled by two L298N Dual H-Bridge Motor Drivers, which are in turn connected to a powerful microprocessor, offering unparalleled control and maneuverability to the robot.

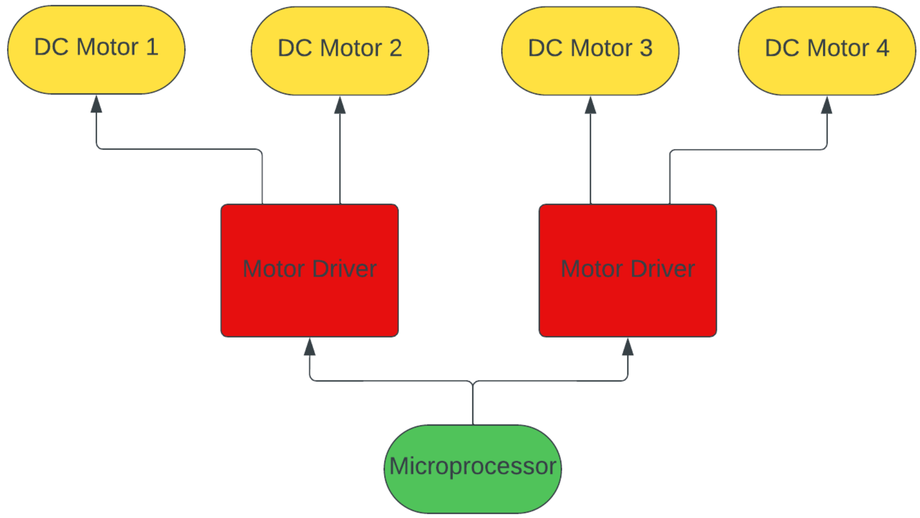


Figure 6.2

Components Overview:

DC Motors (x4):

- The backbone of the locomotion system, these high-torque DC motors are strategically positioned on the FoodieRover, ensuring optimal weight distribution and balanced movement.
- Each motor is carefully calibrated to deliver synchronized motion, allowing the robot to traverse varying terrains effortlessly while carrying food orders securely.

L298N Dual H-Bridge Motor Drivers (x2):

- Acting as the intermediary between the microprocessor and the DC motors, the L298N Motor Drivers provide a robust interface for precise motor control.
- These drivers are adept at managing high-current requirements, ensuring efficient power delivery to the motors while maintaining stability and minimizing heat dissipation.

Microprocessor (x1):

- At the heart of the Motor Subsystem lies a sophisticated microprocessor, programmed to execute intricate navigation algorithms and respond to real-time environmental cues.
- The microprocessor serves as the brain of the FoodieRover, processing sensor data, calculating optimal routes, and sending precise commands to the motor drivers for seamless movement and accurate positioning.

Integration and Functionality:

Interconnection:

- The four DC motors are paired and connected in parallel to the two L298N Dual H-Bridge Motor Drivers, distributing the workload evenly and ensuring redundancy for fault tolerance.
- The motor drivers are intelligently interfaced with the microprocessor, establishing a communication link that enables swift data exchange and control signals transmission.

Control and Precision:

- Through intricate software algorithms, the microprocessor orchestrates the movement patterns and speed adjustments for the DC motors, facilitating smooth acceleration, deceleration, and precise turns.
- The L298N Motor Drivers implement H-bridge configurations, enabling bidirectional control of each motor independently, allowing the FoodieRover to navigate tight spaces and execute precise maneuvers effortlessly.

Safety and Reliability:

- Built-in safety protocols within the microprocessor constantly monitor motor performance, ensuring operational limits are adhered to, preventing overheating or excessive power consumption.
- Redundancy measures are in place within the Motor Subsystem to ensure continuous operation, mitigating the impact of potential failures and optimizing the reliability of the FoodieRover during its delivery tasks.

Performance and Future Enhancements:

- The Motor Subsystem of FoodieRover is a benchmark for reliability, precision, and efficiency in food delivery robotics. However, continuous research and development initiatives are underway to further enhance its capabilities. Future upgrades may involve integrating advanced sensor technologies for enhanced obstacle detection, implementing machine learning algorithms for adaptive navigation, and optimizing power efficiency to extend operational range.

- In conclusion, the Motor Subsystem of the FoodieRover represents a sophisticated blend of engineering prowess and cutting-edge technology. Its seamless integration and precise control mechanisms ensure that the robot navigates with finesse, delivering delectable meals promptly and reliably to its destination.

<u>Sensor Subsystem</u>

The Sensor Subsystem includes the hardware components that work together to allow the robot to perceive its surroundings to the degree that it is able to re-route its direction manually when an obstacle is detected in its path. As a key design specification of this system, this capability is essential so that the robot does not collide with obstacles in its path, including objects as well as moving people. This system serves not only to avoid collision but also to allow the robot to determine the best path of re-routing when an obstacle is encountered so that it does not remain stuck indefinitely until the obstacle moves out of the way (which, in cases of street signs or other fixed objects, may never happen), allowing for a fully self-navigating system. Therefore, a delivery will not be completely halted by the presence of an unexpected obstruction. This subsystem consists of a series of 8 Ultrasonic Sensors that connect individually to a GPIO Expander, which in turn connects with our central microprocessor unit to allow for reading and analysis of the valuable data that these sensors provide.

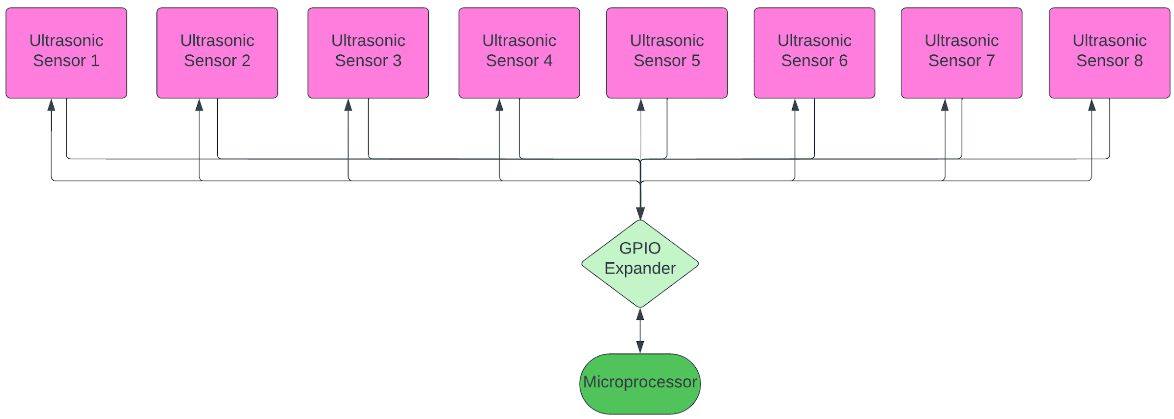


Figure 6.3

Components Overview:

HC-SR04 Ultrasonic Sensors (x8):

- These sensors are designed to measure the distance to objects within their effective sensing range (0.02-4m), providing the ability for obstacle detection and renavigation.
- The sensors communicate their detected distance measurements using digital I/O information which can then be sent to and processed by the central processor in order to achieve full self-navigation capabilities.

MCP23017T-E/SO GPIO Expander (x1):

- The GPIO Expander serves the purpose of providing extra connection terminals to the microprocessor to allow for higher volumes of data transfer and connections to peripheral devices than the module allows for by itself.
- This component provides an additional 16 GPIO pins to serve as an intermediary between the Ultrasonic Sensors and the microprocessor.

Raspberry Pi 4 Model B Microprocessor (x1):

- The microprocessor at the heart of this subsystem serves to analyze the data that it receives from the Ultrasonic Sensors (via the GPIO Expander) through sophisticated software algorithms in order to guide control of the Motor Subsystem on when to stop, turn, and proceed after an obstruction is encountered.
- This component is essential as it provides the logic for the entire system and allows for effective interaction between each of the peripheral components and the different subsystems for fully comprehensive and integrated operation.

Integration and Functionality:

Interconnection:

- Each of the eight Ultrasonic Sensors are wired in parallel to GPIO pins on the Expander unit, with each sensor using two pins: one for triggering an ultrasonic wave and the other for receiving the reflected sound (echo). The latency between these signals allows for determining the distance to nearby objects. The proper operating voltage of 5V will be supplied to each of these pins.
- The GPIO Expander is connected to the Raspberry Pi via the SDA and SCL pins, allowing for I2C communication between these two components for the exchange of data. An input voltage of 5V will also be supplied to the Expander, which is the operating voltage required by the Ultrasonic Sensors.

Control and Precision:

- This subsystem's control ultimately comes from the Microprocessor, which communicates via the I2C communication protocol to the GPIO Expander, allowing it to activate the GPIO trigger pins of the sensors when a distance measurement is to be conducted.
- The precision of this subsystem is dependent on that of the ultrasonic sensors given that these components are taking the measurements which are being used for processing at the central unit. The HC-SR04 Ultrasonic Sensors are specified to have an effective sensing range of 0.02 to 4 meters with a precision of several millimeters, which is more than sufficient for the needs of this project.

Performance and Future Enhancements:

- While the range and precision of the Ultrasonic Sensors used in this project are considerably high quality, more advanced sensors could be used to detect objects at a higher range or with greater accuracy and precision.
- Tradeoffs between number of sensors needed (depending on the angular range of each one and the needs of the system), current consumption of each sensor, and frequency of measurement triggering could be compared in order to maximize the power efficiency of this obstacle-detecting Sensor Subsystem.

Camera Subsystem

The Camera Subsystem is the portion of the FoodieRover hardware which exists to relay visual information with two main objectives: the first being to serve as an added security feature which records the space in front of the robot as it delivers, providing a record of and insight about any potential damages that may have occurred or incorrect behavior exhibited by the system, and the second being to pass images to the processing unit for software analysis in order to help identify obstacles and/or navigate around them effectively using computer vision. This subsystem is comprised of a Camera unit which connects directly to the Microprocessor for two-way communication between these hardware components.

Figure 6.4

Components Overview:

5MP OV5647 Camera Module (x1):

- The camera module uses incoming light patterns to capture and store images that come across its path. These images serve two purposes in this subsystem. In real-time, they are analyzed by the processor unit for enhanced obstacle detection and rerouting capabilities. In terms of this function, the Camera Subsystem is separate but intricately related to the Sensor and Navigation Subsystems, as they will all tie together to realize the goal of advanced obstacle detection.
- The camera unit also serves to implement the purpose of storing this incoming visual information for later viewing if something goes wrong. This will increase the security of the system en route and should assist the team in understanding the source behind any incorrect or unpredicted behavior in terms of navigation.

Raspberry Pi 4 Model B Microprocessor (x1):

- Once again, the microprocessor unit is the heart of this subsystem as it allows for intelligent evaluation and application of incoming information from the peripheral components - in this case, the Camera Module. This information will be utilized by the processor to redirect the Motor Subsystem accordingly.
- In the context of the Camera Subsystem, the microprocessor specifically will be used to implement computer vision software that realizes obstacle detection and understanding of surroundings. This complex operation will further enhance the self-navigation ability of the product for reliable and efficient delivery.

Integration and Functionality:

Interconnection:

- The Camera Module will be connected to the Raspberry Pi board via a 15-pin interface that utilizes Camera Serial Interface (CSI) communication protocol serial communication to relay captured visual information with the processing unit. SDA and SCL pins are also attached between these units for I2C parallel transmission, increasing the overall data transfer capacity.
- A 3.3V input voltage to the 15-pin connector interface between the Camera Module and Raspberry Pi will provide the required working voltage for optimal Camera Module operation conditions.

Control and Precision:

- The Microprocessor component will provide the control mechanism for this subsystem. The processing unit will realize software that initiates data transfer via the 15-pin digital interface as well as the parallel I2C interface connections as needed.
- The OV5647 Camera Module provides 1080 pixel resolution at 30 frames per second. This matches exactly with the camera quality requirements determined in the component selection section of the report for practical operation.

Performance and Future Enhancements:

- Given that computer vision is a field of research which is advancing substantially in recent years, there is always the potential to utilize a higher resolution camera in combination with this emerging software for further heightened detection capabilities that recognize obstacles with greater accuracy, faster, and at further distances.
- Another way to enhance this subsystem would be to implement multiple cameras that allow for a 360 degree field of vision around the entire system as it is traveling, further improving the level of analysis which could be done and avoiding collisions from all directions.

<u>Locking Subsystem</u>

The Locking Subsystem of the FoodieRover is the primary security system which serves to realize the design objective of keeping the enclosed order materials safe during delivery and avoiding theft. This subsystem operates through the activation or deactivation of an electromagnetic lock which will serve to hold the container lid closed with a strong magnetic force when active. Deactivation or “unlocking” will be achieved through input of the correct code to the key pad by the consumer. This subsystem realizes this goal through careful interconnection between the key pad and the microprocessor, and the microprocessor and the electromagnetic lock through the lock relay.

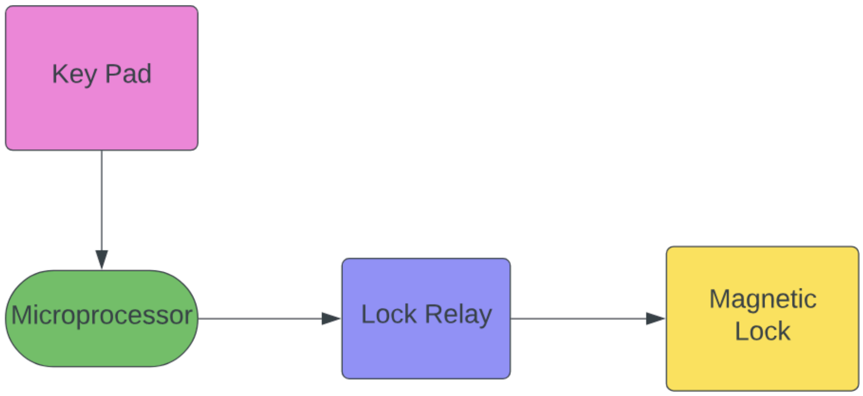


Figure 6.5

Components Overview:

419 4x4 Matrix Keypad (x1):

- The Keypad component is a 4x4 button membrane which is used to deliver user input as signals to the processing unit. Transferring the pattern of keys pressed by the user to the processor allows for software to check the validity of this input and react accordingly to correct or incorrect code information.
- This Keypad unit will operate as the access control mechanism for the system user interface. Upon delivery of an order, FoodieRover's container component will remain locked until the consumer enters the correct code into this keypad, to be verified by software which compares the input with internally programmed passcodes known to be correct.

5V 1-Channel Relay Module (x1):

- This component serves as an electrical switch that can be used to control the status of connected devices which use higher levels of voltage than the microprocessor is capable of delivering. Given that the EM lock operates at 12 Volts, an intermediary such as this relay module must be used to avoid burning the microprocessor with this high voltage.
- The interconnection of this device between the microprocessor and the EM lock allows for the control of the lock's status by controlling the status of the switch: in the "normally open" mode of the relay which we intend to use, the presence of a voltage at the input pin closes the switch and thereby activates the passing of voltage to the device, activating the EM lock's magnetic capabilities.

AGPtek Electric Magnetic Lock (x1):

- The Electromagnetic (EM) lock uses applied electrical power to generate a strong magnetic field in the electromagnet component of the lock, attracting the armature plate component with approximately 60 kg of holding force. Upon proper placement, this lock will be used to hold together the lid and body of the delivery container to serve as the primary method of securing the delivery materials.
- This lock will be interfaced with the Relay Module to allow for the control of the locking/unlocking status of the system by programming whether the switch in the Relay Module is “open” or “closed”, thereby controlling whether a voltage is or is not being applied to the electromagnet.

Raspberry Pi 4 Model B Microprocessor (x1):

- The Microprocessor unit serves to pass information between each of these peripheral components and inspect the information received via applied software design to determine adjustments that need to be made and send data signals to the proper components to adjust these relevant parameters.
- In this Subsystem, the Raspberry Pi will receive input data from the Keypad component as users enter the passcode given to them, inspect this data to determine the accuracy of the entered code, and, upon verification of an accurate passcode, deliver the proper signal to the Lock Relay component to deactivate the lock, allowing the consumer to receive their delivery.

Integration and Functionality:

Interconnection:

- The Keypad unit is designed for simple digital communication with the microprocessor and is interfaced using 8 GPIO pins, one for each row and column, with the combination of each pin’s status allowing for realization of the status of each button when reading user input.
- The Relay Module is connected on the input side to the microprocessor, and on the output side to the EM lock. On the input side, the Relay connects to a 5V VCC node, a GND node, and a GPIO pin to pass the control signal. On the output side, the Relay connects one node to the 12V power source, and the other to the EM lock’s active wire for permitting or preventing the passing of voltage.
- The Electromagnetic lock is interfaced to the Relay Module for control by connecting its active wire to the Common node of the Relay, and connecting its ground wire to the ground terminal of the 12V power source. The positive terminal of the 12V source is thus essentially wired through the Relay, allowing for this module to control the status of its connection to the lock.

Control and Precision:

- Besides the fact that information about the Relay Module status will be passed to the Microprocessor for status checks and appropriate changes, this Subsystem is primarily a one-way communication system. Therefore, the reaction of each element is controlled by the action of the element preceding it. The user input data to the Keypad will dictate what software running in the Microprocessor decides. This decision is passed to the Relay Module, controlling the status of the switch. Finally, the status of the switch controls the status of the EM lock.

- The precision of this system, assuming all components operate in accordance with their design specifications and not considering software factors, depends on the sensitivity of the Keypad unit. The level of pressure one must apply to the buttons of the Keypad in order for the unit to transfer the proper data to the processor dictates the degree to which this system will function correctly.

Performance and Future Enhancements:

- Given that each hardware component of this Subsystem is relatively simple in operation, potential enhancements primarily include upgrading the components themselves to more advanced and complex units that achieve the same function.
- For example, the keypad could be replaced with one which offers better response sensitivity, or the EM lock could be replaced with one which offers a higher degree of holding force. As always, optimization of software applications related to this Subsystem will help to limit unnecessary data readings and activation of components at incorrect times.

Power Subsystem

The Power Subsystem contains the major power supply component and each of the peripheral components to which it connects and supplies with their required operating voltages and currents. This Subsystem must be very carefully integrated, not only to ensure that enough power is distributed to the correct units to meet the 1-2 hour battery life specification, but also to apply the correct balance of voltages to each component and their subsequent peripheral units to meet the operating voltage requirements while not exceeding them, which may cause improper function and breakdown of the components, and can potentially compromise the safety of the system by burning the circuitry.

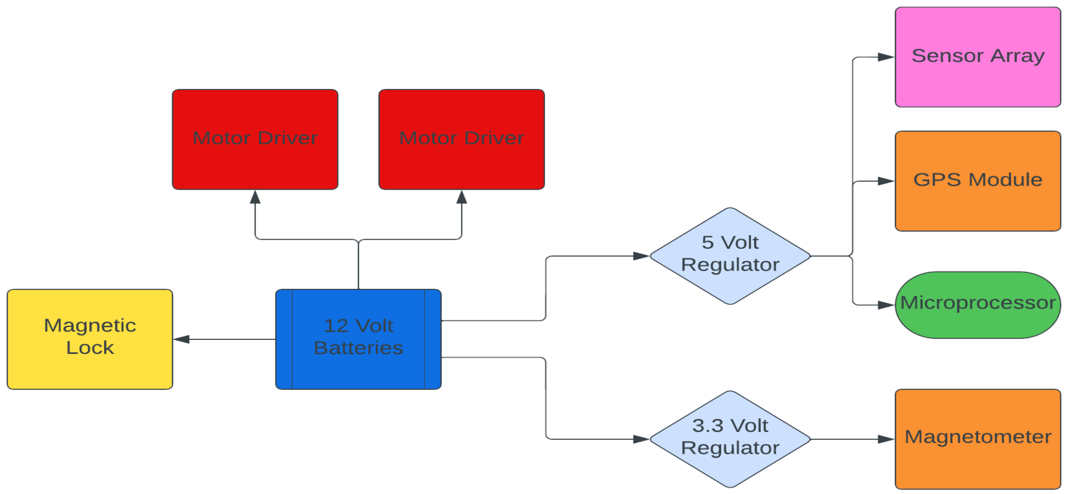


Figure 6.6

Components Overview:

Feuruetc LFP1212 12V Battery (x2):

- The battery components of this subsystem operate as the ultimate source of power for the rest of the system components. Without these batteries, neither the microprocessor nor the many peripherals integrated into the system would have the energy needed to operate in any way.
- The battery capacities have been deliberately selected through calculation of power requirements from each operating unit within the system. The selected batteries will provide sufficient voltage and power to meet the system lifetime before recharging specifications set forth earlier in this report. Additionally, these batteries are able to be recharged, verifying another important specification which was laid out in the planning stage of the project.

L298N Motor Drivers (x2):

- These hardware components are the units used to control the motors for movement of the robot along the determined delivery path. The Motor Drivers supply the correct voltages to each motor and allow for control of their movement, allowing for the path of FoodieRover to be fully automated.
- Each of the Motor Drivers contains two channels and thus controls two motors, supplying the required power to each of them through their interconnections with the 12V Batteries. Interfacing with the Microprocessor allows for the Drivers to control the degree of movement of each peripheral motor at any time.

TPS563200DDCR 5 Volt Regulator (x1):

- This step-down Voltage Regulator serves the vital role of adjusting the voltage level of the batteries at 12V to the 5V input which is required by several of the hardware components in the system, including the Sensor Array, GPS Module, and Microprocessor.

- Utilizing this 5V Regulator allows for operation of connected components at a voltage level which meets their optimal performance specifications, ensuring that each component receives its required input without overloading or burning out their internal circuitry, as would happen if they were directly interfaced with the 12V Battery.

TPS7A4201DGNR 3 Volt Regulator (x1):

- Like the 5V Regulator, this 3V Regulator also serves to step-down the high voltage output of the Batteries for proper interfacing with connected modules.
- This Regulator receives the 12V input from the Batteries and outputs a 3.3V voltage, supplying components such as the Magnetometer with an input voltage that meets safe and proper operation requirements.

Sensor Array (Containing 8 HC-SR04 Sensors):

- The array of 8 HC-SR04 Ultrasonic Sensors used to detect obstacles during travel requires an input voltage of 5V, hence it is connected to the output port of the 5 Volt step-down Voltage Regulator.

L76K GPS Module (x1):

- The GPS Module used to receive spatial positioning information for proper path routing and navigation requires an input voltage of 5V, thus it is interfaced with the 5V output port of the 5 Volt Voltage Regulator Component.

Raspberry Pi 4 Model B Microprocessor (x1):

- The Raspberry Pi Model B Microprocessor component which serves to process incoming information from peripheral units and adjust the operation of these units system-wide requires a 5 Volt input voltage, which it will receive from the output terminal of the 5 Volt Regulator.

HMC5883L Magnetometer/Digital Compass (x1):

- The Digital Compass Module used for obtaining and relaying system orientation data for navigational purposes will operate at an input voltage of 3.3V, which it will obtain through interconnection with the 3 Volt Voltage Regulator.

AGPtek Electric Magnetic Lock (x1):

- The EM Lock system specifies an input voltage of 12V, so this component will not obtain power through one of the Voltage Regulators, but instead from the Battery itself, via routing of the 12V supply through the Lock Relay as discussed.

Integration and Functionality:

Interconnection:

- The basic interconnections involved in the Power Subsystem are achieved through simple connections via physical wiring of input and output terminals of the various components with the desired voltage levels for each case.
- For the 5 Volt Regulator, 3 Volt Regulator, Motor Drivers, and EM Lock, the desired input voltage is achieved through direct connections with the 12V Batteries. In each case, the positive (12V) terminal of the Battery will be tied to the positive voltage source node of the component, and the negative terminal will be tied to ground nodes.
- For the Sensor Array, GPS Module, and Microprocessor, which receive their specified input voltages from the output of the 5V Voltage Regulator, each unit's VDD/supply voltage active node will be routed to the 5V positive output node of the Regulator, with the negative node being routed to the GND pins.
- For the Magnetometer, which receives its 3.3V input voltage from the 3V Regulator, the same pattern of positive to positive and negative to negative routing will be implemented to transfer the proper voltage to the component.

Control and Precision:

- Control of the Power Subsystem is limited in terms of interjecting or modifying any information remotely. Given that this system is highly physically based without software implementation or control (overlooking software's role in activating or deactivating power to certain GPIO or data relay pins to peripheral units), the control of the power distribution in this system is reliant on physically wiring the correct nodes to their respective pairings and expecting the step-down Voltage Regulators to operate properly. Changes to this wiring structure must be done physically if needed.
- Additionally, the Feuruetc Batteries come equipped with a built-in Battery Management System (BMS) which functions to mitigate potential damages from improper use, such as overcharging or overdischarging the batteries. This additional control feature minimizes involvement needed from the team in terms of monitoring and adjusting these practices of battery charging and discharging.
- The precision and accuracy of this Subsystem, again, mostly relies on the integrity of the components and the physical connections themselves. Desired performance may be hindered if certain wired connections are not stable or properly secured, or if the batteries are being used outside of their optimal range, such as in stages of severe discharge.

Performance and Future Enhancements:

- Given that extreme care and caution are observed when wiring these electrical connections between components and that the 12V Batteries are not severely overcharged or discharged during application, this Subsystem should not experience unsatisfactory performance. Variations in performance from moment to moment or trial to trial may potentially arise due to changes in power consumption in hardware components that spur from environmental conditions such as temperature, or from changes in the level of voltage which can be supplied by the batteries given their current state of charge.

- The cyclic lifetime of this Power Subsystem, given that the Batteries are rechargeable, depends on how efficiently the power from these ultimate sources are distributed and consumed. Thus, the most room for enhancement relies on using highly power-efficient devices and routing power from sources to peripheral modules in the most economical manner to minimize wasted power.

6.2 Schematic Diagrams

This section encompasses schematic captures for the system at a global level, as well as schematics utilized in the development of the PCB design. It is intended to provide a more detailed technical insight into the hardware design of FoodieRover. Altium Designer, a professional electronic design automation software, is uniformly utilized for all schematics within this section.

Global Schematic

In this section, we delve into the intricacies of the FoodieRover's electronics, as encapsulated in the comprehensive global schematic of its printed circuit board (PCB). At the core of our design is the ATmega328P-AU microcontroller, the brain behind the Rover's operations, meticulously interfacing with an array of peripheral devices. This schematic serves as a blueprint, detailing the meticulous interconnections that empower the Rover's functionalities—from processing sensor data to driving actuators and managing power.

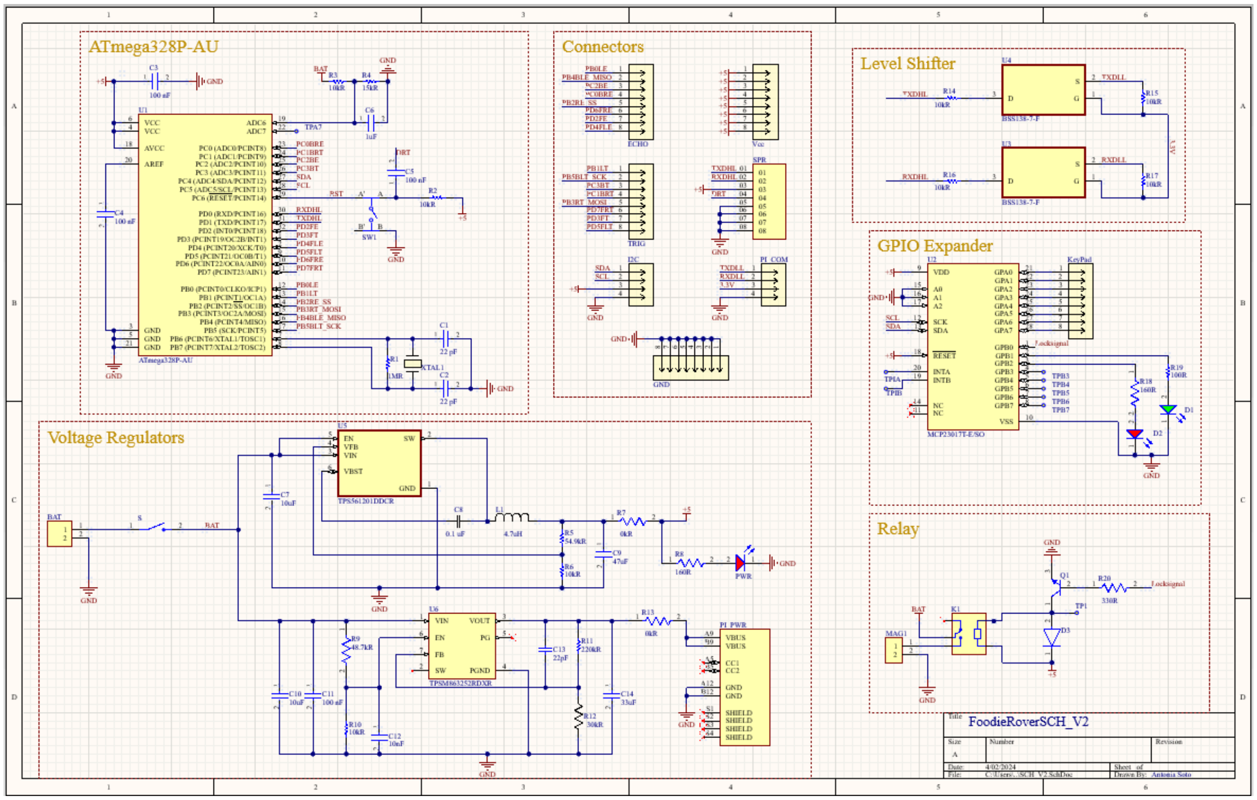


Figure 6.7

The voltage regulators are central to ensuring a stable power supply to various components, a fundamental aspect that was refined following Professor Weeks' invaluable insights. Adjacent to this is the level shifter circuitry, a critical element that facilitates seamless communication between devices operating at differing voltage levels.

Further, we integrated a GPIO expander, broadening our microcontroller's I/O capabilities—crucial for the multitude of sensors and actuators. The relay circuit, represented with clarity, underscores our commitment to safety and control, allowing for the secure actuation of the magnetic lock mechanism.

Each segment within the schematic—be it connectors, level shifters, or the power supply—is demarcated and labeled for clarity. This diagram is not just a representation of the circuitry; it narrates the story of our iterative development process, where each revision brought us closer to the robust and reliable system that FoodieRover is today.

Atmega328p-AU

The ATmega328P microcontroller, renowned for its use in the popular Arduino Uno platform, is the quintessential choice for numerous hobbyist and educational projects due to its versatility and ease of use. In the context of FoodieRover's PCB design, it sits at the nexus of several critical subsystems, interfacing with components such as ultrasonic sensors, the GPIO expander, and potentially a magnetometer, all orchestrated through its array of GPIO pins.

With the ultrasonic sensors connected directly to its GPIOs, the ATmega328P is responsible for initiating distance measurements and processing the resulting pulse signals to determine proximity and aid in navigation. These tasks are well within its operational purview, as the microcontroller is equipped with digital I/O pins that are capable of handling pulse-width modulation and timing-sensitive tasks, which are essential for ultrasonic measurements.

Communication with the GPIO expander via I2C protocol is an efficient way to extend the I/O capabilities of the ATmega328P without the need for additional microcontrollers. This is particularly useful for complex systems requiring numerous sensors and actuaries, as it allows for a streamlined design without sacrificing functionality or scalability.

While the magnetometer's inclusion is optional, its connection via the I2C or serial peripheral interface (SPI) could enhance the FoodieRover’s navigational capabilities by providing heading data. This addition would allow the ATmega328P to calculate orientation relative to the Earth’s magnetic field, a feature that is especially useful in robotics for direction finding and course correction.

Lastly, the ATmega328P's role in facilitating communication with the Raspberry Pi through UART (Universal Asynchronous Receiver/Transmitter) illustrates its function as a bridge between the low-level sensor/actuator interactions and higher-level processing and decision-making carried out by the Raspberry Pi. This UART communication enables a seamless exchange of data between the microcontroller and the microprocessor, ensuring that the FoodieRover’s systems work in concert to perform complex tasks.

Connectors

The connectors, serving as the critical interface points on the PCB, are intuitively designed with a mix of female and male headers to ensure a user-friendly experience. The female headers offer flexibility for frequent connections, while the dedicated male pins for serial programming underscore the precision required during firmware updates. Additional ground pins are incorporated for enhanced circuit stability, a reflection of the careful attention paid to detail. Each connector's label is a nod to simplicity and functionality, facilitating straightforward assembly and maintenance. This design element speaks to the thoughtful planning that went into making the FoodieRover's hardware both accessible and reliable.

Level Shifter

The level shifter on the PCB plays a pivotal role in facilitating communication between the Arduino and Raspberry Pi board, which operate at different logic levels—5V for the Arduino and 3.3V for the Pi. Utilizing N-channel MOSFETs, this level shifter safely converts the UART signals between the two devices, ensuring that the data is transmitted without the risk of voltage-induced damage to the lower-voltage Raspberry Pi.

N-MOSFETS are an excellent choice for this purpose due to their low on-resistance and high-speed operation, which are crucial for maintaining the integrity of the UART signals that are essential for serial communication. This bidirectional level shifting is not only efficient but also a testament to the intelligent engineering behind the FoodieRover, reflecting a design that values both compatibility and component longevity.

Voltage Regulators

The voltage regulators on the PCB are a critical component in ensuring that all peripherals, as well as the Raspberry Pi board, receive the appropriate voltage from the FoodieRover's power source. In our design, we have incorporated two distinct step-down (buck) regulators to manage the power distribution efficiently.

The first regulator, designed to handle up to 1A of current, is responsible for powering the array of peripheral devices. This includes sensors and actuators which do not individually draw significant amounts of current but collectively require a stable and reliable power supply.

For the Raspberry Pi board, which serves as the central processing unit and is inherently more power-hungry, especially when additional components are powered through its GPIO pins or USB ports, a more robust solution is required. Hence, a separate 3A regulator is dedicated exclusively to the Pi. This ensures that the Pi operates within its optimal power specifications, preventing any brown-out or power insufficiency issues that could lead to system instability or unexpected behavior.

By splitting the power supply tasks between two regulators, we not only balance the load effectively but also add a layer of redundancy. Should one regulator fail, it won't affect both the peripherals and the Raspberry Pi, thus enhancing the overall reliability of the FoodieRover. This thoughtful power management strategy exemplifies the project's design philosophy where system efficiency and reliability are paramount.

GPIO Expander

The GPIO expander on the FoodieRover's PCB is an astute addition that multiplies the microcontroller's I/O capabilities. It extends the limited number of GPIO pins available on the ATmega328P, providing additional ports needed for the myriad of sensors and controls required in the rover. By using an I2C interface, this expander minimizes pin usage on the main microcontroller, preserving valuable resources while maintaining a breadth of functionality. This allows for a more complex system that can handle multiple tasks simultaneously, without sacrificing design simplicity or efficiency. The choice of an expander is a hallmark of modular design, enabling the FoodieRover to evolve with future expansions or enhancements with ease.

Relay

The relay circuit on the PCB is a quintessential component, designed to control the magnetic lock, which is a key part of the FoodieRover's security mechanism. In the schematic, you can identify a transistor, which acts as a switch, driven by a signal from the GPIO expander. When the expander outputs a signal, the transistor activates, allowing current to flow and energize the relay coil.

This activation of the relay establishes a connection between the contacts that control the power to the magnetic lock. Also included in the circuit is a flyback diode, which is crucial for protecting the transistor from voltage spikes that occur when the relay coil is de-energized.

The selection of these components reflects a design that prioritizes both functionality and safety. The relay allows for the use of higher current and voltage to operate the magnetic lock, which the low-power signal from the GPIO expander wouldn't be able to drive directly. This setup illustrates a well-engineered approach to power management and component control within the FoodieRover's electrical system.

6.3 Structural Illustration

The FoodieRover is an innovative endeavor aimed at crafting a highly adaptable robot capable of transporting food in various environments. Its structural layout embodies a fusion of robust engineering and technological sophistication. This section delves into the intricate design elements, showcasing the core components that constitute the FoodieRover and explaining their functionalities. Onshape, a computer-aided design software system, was used to create the model of the FoodieRover design in this section.

Robot Chassis and Mobility:

At the heart of the FoodieRover lies its structurally engineered chassis, a resilient foundation equipped with four tracks carefully designed to ensure stability and flexibility in movement. These tracks afford the robot exceptional traction and maneuverability across a spectrum of terrains, ranging from rugged landscapes to wet surfaces to indoor settings. The chassis' robust construction not only bolsters the robot's durability but also guarantees its ability to navigate challenging environments with ease, ensuring the safe and efficient transportation of food items.

Under compartment:

Strapped under the chassis is a specialized compartment dedicated to safeguarding the intricate network of circuits, wires, and batteries vital for the robot's operation. This compartment is designed to shield the internal electronics from moisture and other external elements that could potentially impede the robot's functionality. Its waterproof design empowers the FoodieRover to traverse wet surfaces without compromising its operational efficiency, thereby enabling seamless performance regardless of weather conditions. This compartmentalization ensures the safety and integrity of the robot's electronic infrastructure, contributing significantly to its reliability and adaptability.

Cooler Holder :

Sitting atop the chassis is a purpose-built holder meticulously designed to secure the cooler in place. This holder's engineering prowess is evident in its stability-enhancing features, ensuring the cooler remains firmly fixed during the robot's movements.

Cooler:

The cooler, an integral component of the FoodieRover, serves as a temperature-regulated chamber capable of housing perishable items. Its insulation properties maintain the desired temperature, preserving the freshness of food items throughout the robot's journey.

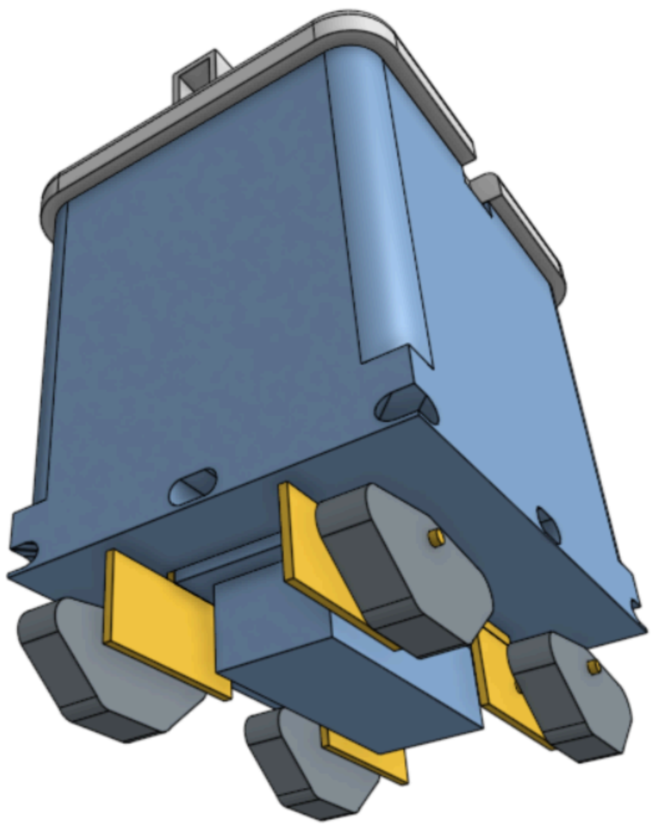


Figure 6.13

Sensing:

Above the cooler resides the sophisticated visual sensing mechanism, constituting cameras, sensors, and potentially lidar technology. This cutting-edge assembly endows the robot with the ability to perceive and interpret its surroundings accurately.

Keypad and Lock:

An integrated keypad which provides users with an intuitive interface to input commands and instructions. The Keypad and Lock is an important aspect of FoodieRover because it ensures the safety of the product inside the cooler.

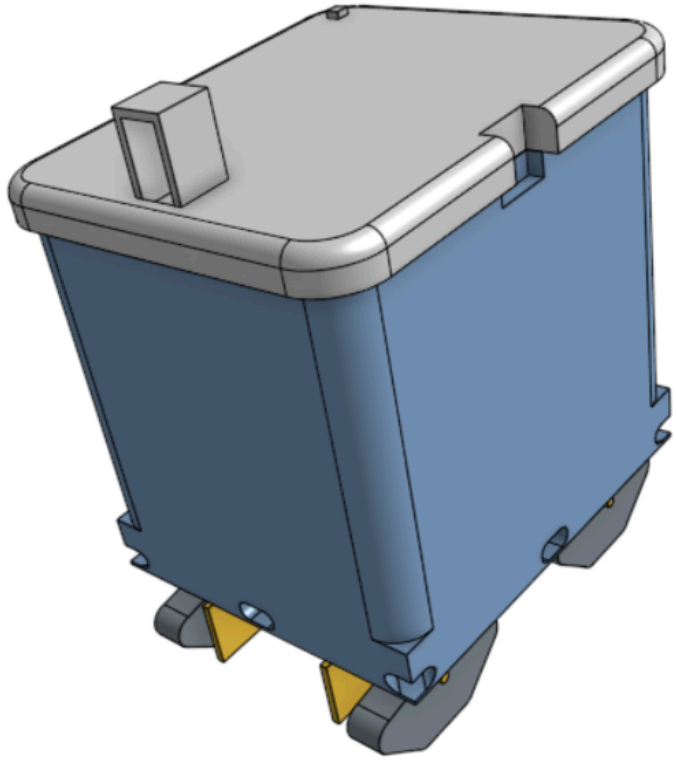


Figure 6.14

Conclusion

The FoodieRover's structural illustration reveals a well-balanced blend of technical innovation and engineering understanding. The design exemplifies adaptability and durability, from its sturdy chassis with four tracks that provide mobility across a variety of terrains to the watertight compartment that fortifies essential electronics and the designated cooler holder for safe food delivery. The combination of sophisticated visual sensing techniques and intuitive user interfaces highlights "FoodieRover's" potential as an adaptable food delivery system in a variety of settings.

7. Software Design

This segment is specifically designated for the examination of the software employed in our project. The scrutiny will encompass diverse applications of the aforementioned software, encompassing areas such as hardware implementation, computer vision, and robotic frameworks. Additionally, a comprehensive review of the software's design will be undertaken, elucidating its objectives and aspirations, thereby providing a more profound understanding of the principles that informed specific decisions.

<u>Use Case Diagram</u>

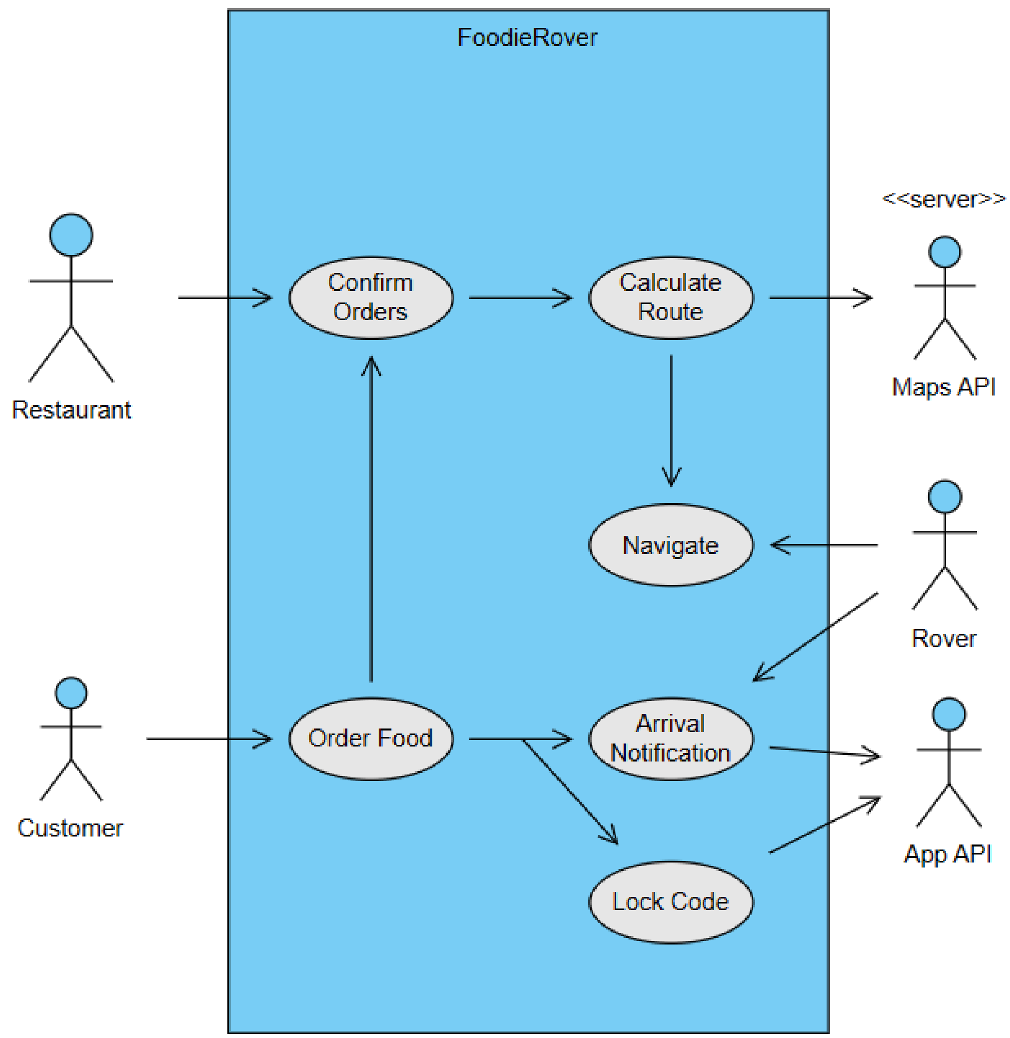


Figure 7.1

A use case diagram is a visual representation of the interactions between various actors (users or external systems) and a system, showcasing the different ways the system can be utilized. It is a type of Unified Modeling Language (UML) diagram commonly used in software engineering to depict the functional requirements of a system from a user's perspective. In the use case diagram for FoodieRover, three primary actors and two servers are depicted: the customer, the restaurant worker, the rover, Google Maps' API, and our proprietary API. These entities engage in collaborative interactions to facilitate a streamlined food delivery process. The sequence commences with the customer initiating an order, prompting communication with the restaurant worker for confirmation. Subsequently, the customer's location is transmitted to Google Maps' API to establish an optimal route for the rover. Upon the rover's successful navigation, a notification is relayed back to the customer via FoodieRover's proprietary API, accompanied by a code enabling the unlocking of the delivery container.

State Diagram

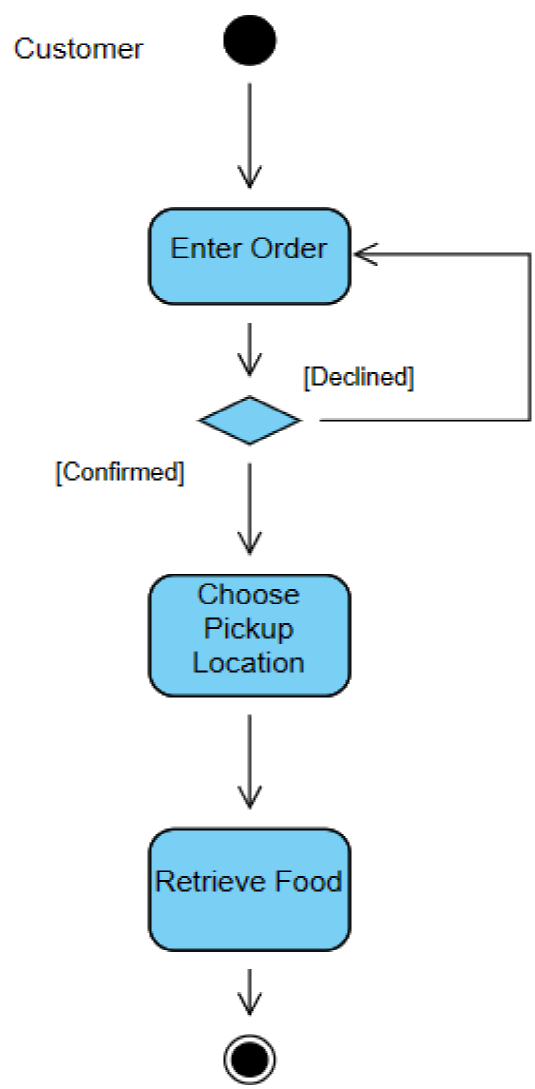


Figure 7.2

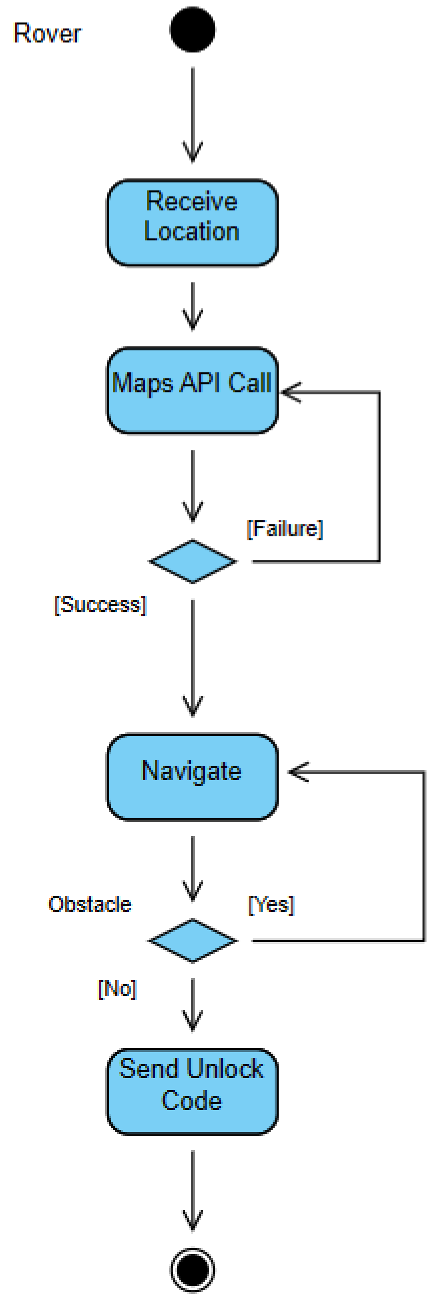


Figure 7.3

A state diagram, also known as a state machine diagram, is a visual representation in Unified Modeling Language (UML) that illustrates the various states an object or system can go through during its lifecycle. It describes the behavior of a system by depicting its states, events triggering transitions between states, and actions performed in each state.

The initial illustration portrays a state diagram for the user, or customer, of the FoodieRover service. This diagram is straightforward, depicting the user placing an order, selecting an appropriate destination, and, upon arrival, utilizing the received code to unlock the container and retrieve their food.

The subsequent diagram delineates the state transitions for FoodieRover throughout this operational phase. Following the reception of order information, which includes the designated drop-off location, FoodieRover initiates an API call to Google Maps' servers to obtain a suitable route. While in transit, should FoodieRover encounter any obstacles, it will interrupt its progress and endeavor to reorient itself around the impediment. Upon reaching the specified destination, FoodieRover utilizes its proprietary API to transmit a code to the customer, facilitating the disarming of the lock and enabling access to the delivered food.

Class Diagram

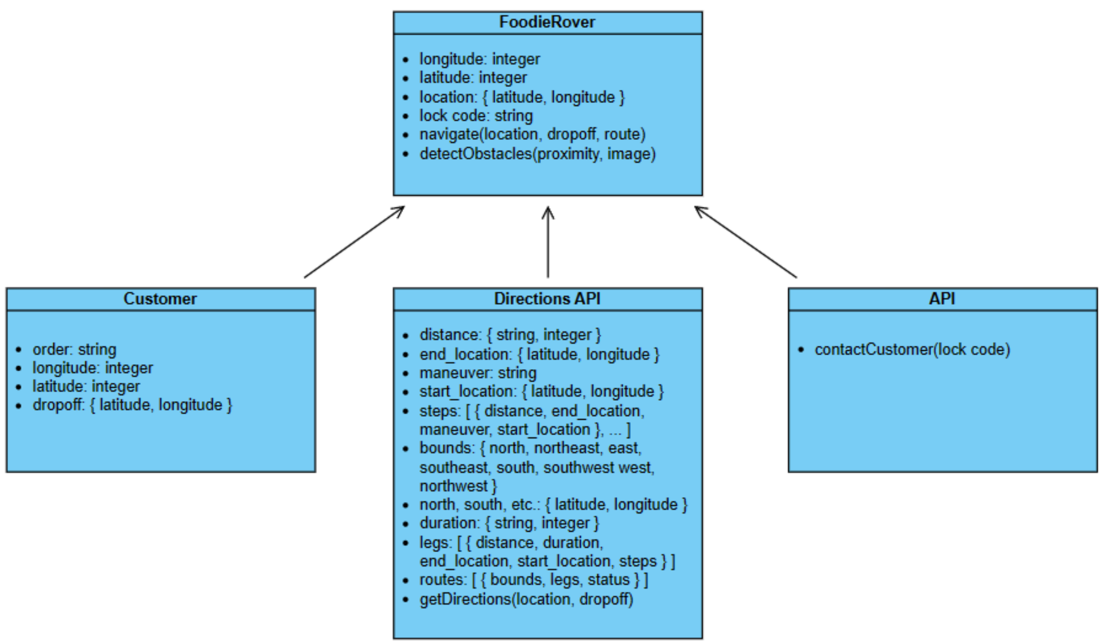


Figure 7.4

A class diagram is a visual representation in Unified Modeling Language (UML) that depicts the structure and relationships within a system by illustrating the classes, attributes, methods, and associations between objects. As evident from the above, the functionality of FoodieRover is reliant on three subordinate classes to ensure proper system operation. The customer class facilitates the acquisition of the drop-off location coordinates. The Google Maps' Directions API class plays a pivotal role by furnishing a comprehensive set of information utilized in constructing the route object for FoodieRover. The routes array, comprising legs, serves as extended segments of the pathing information, and each leg further encompasses steps. These steps, being the smallest increments of pathing information, are delineated based on alterations in maneuver. Lastly, a local API is employed to convey the lock code to the customer.

Firmware

A key aspect of the software within FoodieRover is its applications on a hardware level. These include the ultrasonic sensors, the GPS module, the magnetometer, and the DC motors.

HC-SR04 Ultrasonic Sensors

The initiation of the ultrasonic sensors' code development will be executed using Python, leveraging its extensive collection of readily available libraries for streamlined implementation. Primarily, the Raspberry Pi library designated for GPIO pins, denoted as RPi.GPIO, will be installed. This installation can be achieved through a pip command. Several crucial functions from the GPIO library, including setmode(), setup(), output(), input() and cleanup(), will be employed to facilitate the development process.

- GPIO.setmode(mode: int)
    - Description: sets the GPIO identification mode to either GPIO.BCM for their Broadcom SOC channel numbers or to GPIO.BOARD for association with the physical numbers presented on the board
    - Return Value: none
- GPIO.setup(pin: int, direction: int)
    - Description: configures a GPIO pin for input or output
    - Return Value: none
- GPIO.output(pin: int, state: int)
    - Description: sets a GPIO pin to GPIO.HIGH or GPIO.LOW
    - Return Value: none
- GPIO.input(pin: int)
    - Description: reads the input state of the specified GPIO pin
    - Return Value: int, 0 for GPIO.LOW and 1 for GPIO.HIGH
- GPIO.cleanup()
    - Description: resets all GPIO pins and releases GPIO resources
    - Return Value: none

In addition to the GPIO methods, consideration must be given to the pins of the ultrasonic sensor, encompassing VCC, TRG (trigger), ECH (echo), and GND. In this project, n GPIO extender is employed to accommodate all 8 sensors. On this extender board, the TRG and ECH pins are individually assigned GPIO pins for each sensor, while VCC and GND are grouped together, respectively.

The development of a function for actively measuring the distance of nearby objects from the sensors is imperative. A practical approach involves implementing a while-loop that continuously monitors the time elapsed between the echo pin transitioning from low to high. At the inception of this function, the GPIO pin associated with TRG is set to high and then low. Subsequently, a while-loop is entered, persisting as long as the ECH pin maintains a low reading. Upon the ECH pin registering a high signal, the initial while-loop is terminated, leading to another while-loop that iteratively records the time until the ECH pin reverts to a low state.

While it may initially appear redundant to repeatedly record the time when ECH reads high, this redundancy is intrinsic to the functionality of the TRG and ECH pins. When the TRG pin is elevated to a high state, a brief sonic burst of 8 cycles at 40KHz is emitted, and then the ECH pin is triggered to a high state, not by the code but by its electrical configuration. The ECH pin returns to a low state once the sensor receives feedback from the emitted waves, necessitating the measurement of the duration during which the ECH pin remains high.

NEO-M8P-2 & Polaris API

At the crux of FoodieRover's ability to navigate is the NEO M8P-2 board. This board comes equipped with a chip capable of real time kinematic (RTK) calculations. RTK positioning is a sophisticated satellite navigation technique that significantly enhances the accuracy of position data derived from satellite systems such as GPS, GLONASS, Galileo, or BeiDou. Unlike standard GPS which offers meter-level accuracy, RTK improves positioning to centimeter-level precision by utilizing the phase of the signal's carrier wave in addition to the content of the signal itself. This advanced method is integral to applications requiring high precision such as surveying, autonomous vehicles, and precision agriculture.

The primary signal used in GPS includes two components: the carrier phase and the code phase. The code phase, which most consumer GPS units rely on, can determine a receiver's position to within a few meters by calculating the time delay of the signal travel from the satellite to the receiver. However, the carrier phase—the actual wave component of the satellite signal—offers a much finer granularity because it includes the entire wavelength of the signal, which has a much shorter cycle than the code itself.

The carrier phase is crucial for high precision because it allows the receiver to observe fractions of a wavelength with high accuracy. This fraction can tell us more precisely where the receiver is relative to the satellite. However, the carrier phase introduces a new complexity: it only measures the fractional part of the total number of wavelengths from the satellite to the receiver, not the integer number of complete wavelengths. This unknown integer number is what we refer to as the integer ambiguity, and it must be accurately resolved to use the carrier phase measurement effectively.

Once the integer ambiguity is resolved, the RTK system can lock onto a fixed integer number, which it uses to reference the subsequent carrier phase changes, effectively tracking the receiver's minute movements relative to the satellite. This results in an ability to measure positions down to the centimeter scale, significantly enhancing accuracy over standard GPS methods, which typically resolve positions to within larger margins of error.

The core of RTK technology lies in its use of both a base station and a rover. The base station is a fixed GPS receiver located at a known coordinate, which continuously monitors the satellite signals and calculates correction factors. These corrections account for GPS signal errors including delays caused by the ionosphere and troposphere, satellite orbital errors, and inaccuracies in the satellite clock. The base station then broadcasts this correction data to the rover—a mobile GPS receiver—via radio or other communication links.

Upon receiving these corrections, the rover faces the challenge of resolving the integer ambiguity—the unknown number of full wavelengths between the satellite and the receiver—which is crucial for precise positioning. The integer ambiguity resolution is a sophisticated process that uses complex algorithms to count the exact number of wavelengths, thus significantly enhancing positional accuracy.

Besides integer ambiguity resolution, RTK technology effectively corrects several other sources of GPS errors. These include ionospheric and tropospheric delays that affect the speed and path of GPS signals, orbital errors pertaining to the satellite's reported vs. actual position, and timing discrepancies between the satellite's and the receiver's clocks. Multipath errors, which occur when signals bounce off surfaces before reaching the receiver, are also mitigated.

As stated earlier, RTK requires both a rover and a base station. Base stations can go for upwards of $1,000 and were not a realistic option for this project. Instead, we opted to use a service by the name of Polaris. Polaris provides APIs that enable access to Radio Technical Commission for Maritime (RTCM) correction streams via their network of base stations. This service allows users to obtain the necessary data for high precision GPS without the need to invest in physical base station hardware. The integration of Polaris services into our project was facilitated through a custom-made C++ library, specifically designed to streamline the process of making API calls to Polaris servers. Here’s an example of how to initialize the Polaris client:

```
PolarisClient polaris;

void initPolaris()
{
        polaris.SetRTCMCallback([this](const uint8_t *buffer, size_t size_bytes)
                { rtcmPort.Write(buffer, size_bytes); });

        polaris.RunAsync();
}
```

This initialization function configures the Polaris client to receive RTCM data. The SetRTCMCallback function is used to define a callback that handles the data once received—it writes the RTCM data to rtcmPort, which is then used to correct GPS signals. The RunAsync method is particularly crucial as it allows the data retrieval process to occur asynchronously. This means that the RTK system can continue to operate efficiently, processing incoming serial data from the NEO-M8P-2 board which contains the recalibrated coordinate positions, without delay or interruption from the RTCM data handling process.

HMC5883L Magnetometer/Digital Compass

FoodieRover will be using the HMC5883L compass module to enhance its own position system. This module uses I2C to communicate and we need to ensure that this method of communication is activated on the Raspberry Pi 4. We can do so by navigating to the Raspberry Pi configuration settings and entering the Advanced Options. Another resource that is going to be necessary to the functionality of the code regarding the compass is Python3 and the Git client. The purpose of the Git client is to clone GitHub repositories that contain necessary libraries to assist with controlling the module. We can install these tools by using the following lines:

sudo apt-get install git i2c-tools python-smbus python3 python-pip python-virtualenv python3-setuptools
git clone https://github.com/quick2wire/quick2wire-python-api
git clone https://bitbucket.org/thinkbowl/i2clibraries.git

The I2CLibraries GitHub repository contains functions that allow us to read the HMC5883L compass by importing the class i2c_hmc5883l. The i2c_hmc5883l class provides useful methods to easily facilitate different functionalities of the HMC5883L compass. Some of the methods are listed below.

- i2c_hmc5883l(port, addr=0x1e, gauss=1.3)
    - Description: this is the constructor, and the integer argument being passed in as a port indicates which bus the device's address can be found on for I2C communication
    - Return Value: i2c_hmc5883l Object
- setContinuousMode()
    - Description: sets the compass module to continuously read measurements
    - Return Value: none
- setDeclination(degree, min)
    - Description: sets the values necessary to calculate for the offset between magnetic north and true north (magnetic declination) such as the degrees east or west of true north and the arc minutes
    - Return Value: none
- getAxes()
    - Description: retrieves the current measurement of the axes with respect to the 3D frame described by the compass in terms of x, y, and z
    - Return Value: tuple (x, y, z)

L298N DC Motor Driver

FoodieRover will employ the L289N DC Motor Driver Module to facilitate the functionality of its four DC motors, encompassing operations such as forward mobility, backward mobility, acceleration, and deceleration. The selected programming language for interfacing with these motors is Python, and for this purpose, the RPi.GPIO library will be imported.

In addition to utilizing GPIO methods for configuring the necessary pins to transmit signals to the motors, the library will also be employed to implement the Pulse Width Modulation (PWM) technique. PWM is employed to regulate the rotational velocity of the motor by alternating the signal between active and inactive states, thereby enhancing energy efficiency. Instead of supplying a constant voltage, PWM achieves control by manipulating the average voltage over time, determining the effective voltage applied to the motor. The distinctive characteristic of PWM lies in its frequency, denoting the rate at which the signal oscillates.

Furthermore, the RPi.GPIO methods encompass the management of the duty cycle, defined as the duration during which the signal performs its duty, or work, which, in this context, involves maintaining the power supply. The duty cycle is expressed as a percentage of the total cycle time during which the signal remains high. The higher the percentage that duty cycle represents the faster the motor will spin. These two crucial methods encapsulated within the RPi.GPIO library are as follows:

- GPIO.PWM(pin, frequency)
  - Description: this method initializes a PWM object for the specified pin (integer) at the given frequency (float) which will enable the pin for PWM functionality
  - Return Value: PWM Object
- GPIO.PWM.ChangeDutyCycle(duty_cycle)
  - Description: duty_cycle is a float representing the percentage of the total cycle to set the signal to on for and is a method of PWM objects that is used to augment the PWM signal for the pin specified by the object
  - Return Value: none

Wi-Fi/LTE

A notable challenge inherent in this project pertains to ensuring the consistent availability of a stable internet connection for FoodieRover. Such connectivity is imperative as FoodieRover necessitates the capability to initiate API calls not only to Google Maps' services but also to the server hosting the mobile application. Leveraging the integrated Wi-Fi adapter of the Raspberry Pi 4, the establishment of this connection can be executed through the Graphical User Interface (GUI). In the case of the GUI, the procedure is relatively straightforward, involving the selection of the Wi-Fi symbol, perusing the list of available connections, selecting the desired network, and, if secured with WPA-2 (Wi-Fi Protected Access 2), entering the requisite password when prompted.

An additional consideration is the decision of whether FoodieRover should establish a connection to UCF's on-campus Wi-Fi. This choice is influenced by the observed intermittent coverage drop-offs experienced across various locations on the campus Wi-Fi network. Consequently, an alternative approach involves establishing a Long-term Evolution (LTE) connection with the Raspberry Pi 4. This entails the utilization of a Mini PCIe (Peripheral Component Interconnect Express) card, capable of interfacing with a SIM card equipped with a data plan. Subsequently, a Mini PCIe WWAN Card to USB adapter and an antenna become necessary components for this configuration.

Upon completion of the aforementioned configurations, the subsequent step involves the setup of the Qualcomm MSM Interface (QMI), specifically designed for cellular modules predicated on Qualcomm chipsets. To instantiate this interface for our Mini PCIe module, it is imperative to install libqmi initially. Following this, the modem is to be ensured in an online state. Subsequently, configuration for the raw-ip protocol is necessary. Subsequent to these steps, the mobile network can be established by modifying the Access Point Name (APN), username, and password to align with the settings stipulated by the SIM card and the respective mobile operator. Lastly, the IP address and default route can be configured through the utilization of udhcpc (ParryTech, 2022). The ensuing representation elucidates these procedures within the terminal:

```
sudo apt install libqmi-utils udhcpc
sudo qmicli -d /dev/cdc-wdm0 -dms-set-operating-mode='online'
sudo ip link set wwan0 down
echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
sudo ip link set wwan0 up
sudo qmicli -p -d /dev/cac-wdm0 -device-open-net="net-raw-ip|net-no-qos-header" -wds-start-network="apn='YOUR_APN',username="YOUR_NAME",password="YOUR_PASSWORD",ip-type=4" -client-no-release-cid
sudo udhcpc -q-f-i wwan0
```

As a third alternative for FoodieRover, we can consider establishing a connection by tethering the Raspberry Pi 4 to a hotspot generated by one of our mobile devices, which already accesses an LTE connection through a Data plan. Among the three options, I perceive this approach as the most viable. The primary challenge lies in identifying an optimal location to affix the device on FoodieRover, ensuring that the connection remains uninterrupted and unaffected by distance limitations.

Robot Operating System

Running the infrastructure of FoodieRover is ROS. The Robot Operating System (ROS) is an open-source, flexible framework for writing robot software. It provides a structured communications layer above the host operating systems of a heterogeneous compute cluster. Although ROS itself is not an operating system (OS), it provides services designed around a distributed computing model that one would expect from an OS, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.

ROS is designed to be modular at a fine-grained scale. A ROS-based system is composed of numerous independent nodes, each of which communicates with other nodes using a publish-subscribe messaging model. For example, one node controls a robot's wheel motors, another node might handle laser range-finding, while another might manage high-level tasks like navigation and path planning. This modularity makes ROS very flexible and powerful in robotic research and development.

ROS also includes tools and libraries that help in the creation and running of complex and robust robot behavior across a wide variety of robotic platforms. Whether for academic or commercial development, ROS provides drivers, algorithms, and tools to help developers create robot applications. It can handle everything from flying robots at scale to robotic arms involved in precise manufacturing. ROS supports a vast array of sensors and actuators and integrates with advanced algorithms for functions like kinematic path planning and point cloud processing. In industries ranging from mobility to agriculture, ROS is being used to enhance the capabilities of all kinds of robots, providing the roboticist with the ability to build more capable and exciting robotics applications.

The Raspberry Pi 4 is running off of Ubuntu 22.04, a Linux distribution. For this version of Ubuntu 22.04, ROS 2 Humble is recommended. A particularly notable difference between ROS 1 and ROS 2 is the package building process. In ROS 1, the build system utilized is catkin, which integrates with CMake to facilitate building packages. The typical command used is catkin_make, which allows developers to compile packages one CMake file at a time, giving them control over the build process and making it relatively straightforward to manage dependencies and build paths.

In contrast, ROS 2 uses a tool called colcon as its standard build tool. Colcon is designed to improve the building of packages not just for ROS, but for other software projects that use different programming languages and build systems alongside ROS. The command colcon build effectively manages multiple packages by accessing all the CMake files in the workspace directory. This approach simplifies the building process for large projects with complex dependencies, supporting a more modular and decoupled architecture recommended for modern software practices.

To integrate many of the custom libraries, the proper dependencies needed to be met. To satisfy these requirements we created our own CMake files to ensure that the proper libraries were linked together. Once the appropriate packages were set up, foodie_driver and foodie_polaris, ROS was ready to start communicating across the custom node we made.

Flowchart

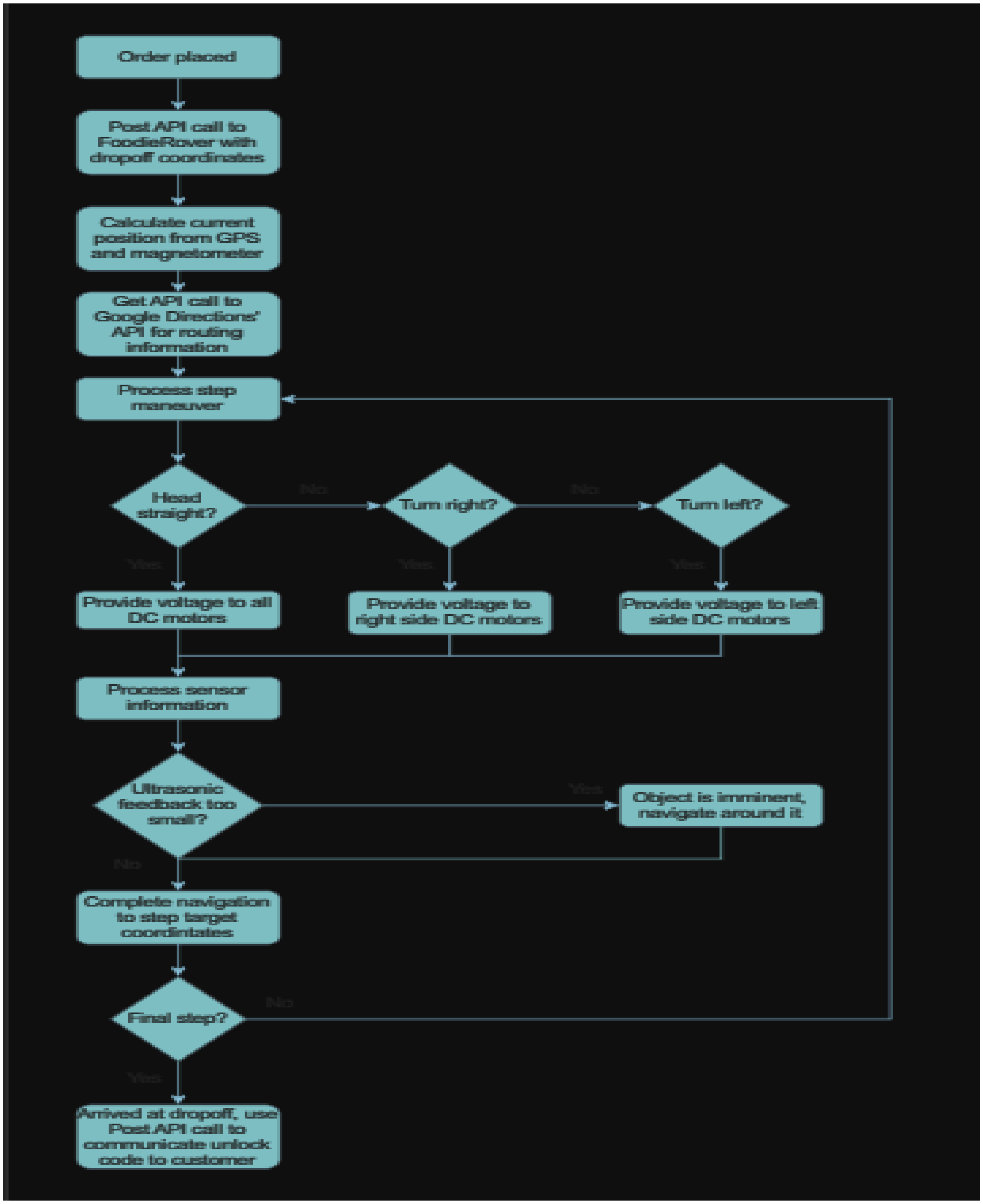


Figure 7.5

This software flowchart delineates the operational sequence within FoodieRover's computer during a delivery attempt, providing a comprehensive overview of firmware interactivity and the requisite server-side interactions for seamless execution. Upon placement of an order through the FoodieRover application, the associated dropoff location is communicated to FoodieRover. Subsequently, the rover calculates its current position, a crucial step for obtaining comprehensive routing information from Google's Directions API. It is imperative to communicate both the starting and ending coordinates for this purpose.

The information retrieved from the API call encompasses an array of steps, fragmenting the entire route into discrete segments characterized by changes in direction. FoodieRover then processes each step, examining details such as maneuver descriptions and the step's target coordinates. Based on this information, it determines the requisite operations for the DC motors and initiates obstacle detection protocols. In the event of an encountered obstacle, FoodieRover executes basic evasive maneuvers. Upon completion of processing all steps, it may be inferred that FoodieRover has arrived at the designated dropoff location. At this juncture, the customer is notified with a code, enabling them to unlock the food storage and retrieve their meal.

Mobile Development

The mobile application for FoodieRover will be crafted utilizing React Native, a highly regarded cross-platform framework developed by Meta. Leveraging JavaScript as its primary programming language and React as its framework, React Native offers the advantage of code reusability across multiple platforms. It excels in providing a responsive and intuitive user interface, essential for an engaging mobile experience. Complementing this, Node.js is chosen for backend development due to its seamless integration with JavaScript, featuring an event-driven, non-blocking I/O model. This ensures efficient handling of concurrent requests, critical for the real-time data processing and communication required by FoodieRover.

Express, a minimal and flexible web framework for Node.js, will be employed for API implementation, showcasing its simplicity and scalability. Express streamlines the creation of robust and RESTful APIs, facilitating seamless communication between the frontend and backend components of the application. In terms of database management, MongoDB, a prominent NoSQL database, aligns well with JavaScript, providing flexibility and scalability for data storage. MongoDB's document-oriented model allows for easy representation and storage of complex data structures, crucial for managing diverse information related to FoodieRover's functionalities.

To ensure the robust deployment and scalability of the application, Google Cloud Platform (GCP) has been selected as the hosting solution. GCP offers a comprehensive suite of scalable and fully managed services, providing a reliable infrastructure for the seamless operation of FoodieRover's mobile application. This strategic alignment of React Native, Node.js, Express, MongoDB, and GCP forms a cohesive and effective technological stack.

User Interface

FoodieRover incorporates a mobile application that allows users to place orders for their preferred food items. This application holds pivotal significance in FoodieRover's service, as the data transmitted from the application to FoodieRover includes crucial information regarding the delivery destination. In the design phase of the mobile application, I utilized Figma to create an intuitive and user-friendly interface.

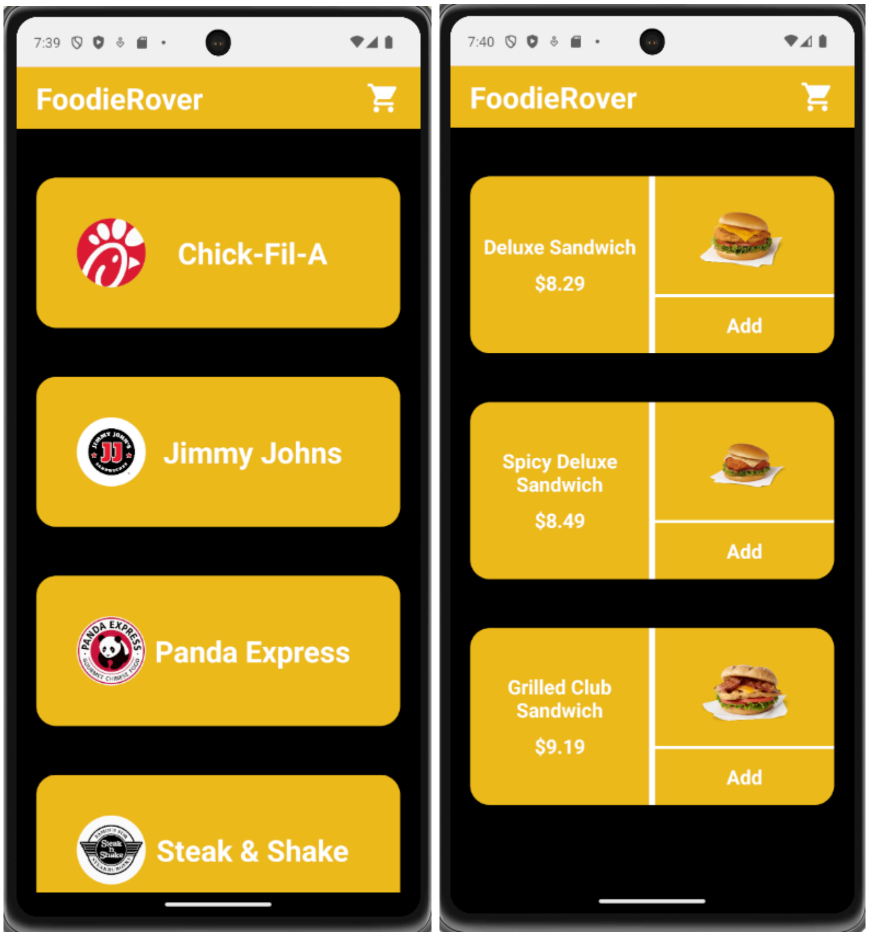


Figure 7.6

The left-hand display serves as the home page of the mobile application. Upon the application's initiation, users are prompted to specify their desired on-campus restaurant from which they intend to place an order. This initial screen serves as the gateway for users to navigate and make selections.

Upon selecting a restaurant, the right-hand display emerges as a follow-up screen, presenting users with a list of menu items associated with the chosen establishment. Each menu item is accompanied by an option to add it to the user's cart, facilitating a streamlined and intuitive ordering process.

This two-step interface design ensures a clear and user-friendly experience, enabling users to seamlessly navigate restaurant options, make selections, and progress through the ordering process with ease.

Figure 7.7

This interface represents the user's shopping cart and is accessed through the selection of the "Cart" navigation bar element. It presents comprehensive information pertaining to the pricing of each item within the cart and provides a designated area for users to input their credit card details. Additionally, it prompts users to specify the critical dropoff location. Once both sets of information are furnished, users can proceed to finalize their order by selecting the "Pay" option.

8. System Fabrication/Prototype Construction

In this section, we delve into the process of PCB fabrication and the subsequent construction of a prototype, both important to the realization of FoodieRover. The fabrication of a printed circuit board (PCB) stands as a foundational element of the system.

PCB

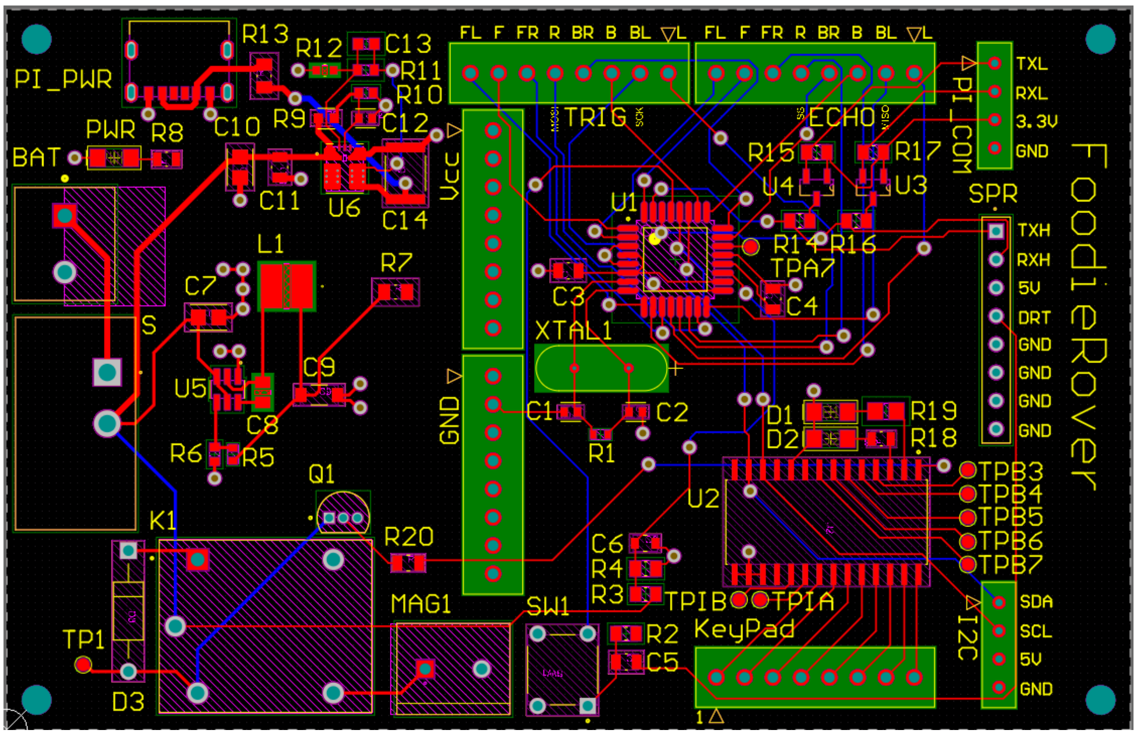


Figure 8.1

The schematic diagrams section shows the specific components as well as the circuitry design that will be part of the PCB. Every major component will be powered through the PCB and all the data that contains the critical information will also move through it. Additionally, the PCB offers protection to the overall circuit and offers an organized basis for the wires that will connect the components at different locations to the Raspberry Pi. The process of PCB fabrication involves the precise application of materials and techniques to create the physical representation of the envisioned electronic circuitry.

During the component selection process, meticulous consideration was devoted to scrutinizing the specifications of each part. Parameters such as tolerance, voltage limits, and power dissipation held particular significance in ensuring the chosen components met the precise requirements of the design. In tandem with these technical considerations, availability and cost emerged as pivotal factors influencing the selection process. Altium Designer's Manufacturer Part Search panel facilitated this discerning process by providing robust filtering capabilities for specific parts. Notably, the platform not only aids in identifying suitable components but also offers supplementary information, including the status of the part's availability and the existence of associated models. This comprehensive approach to part selection, bolstered by Altium's intuitive tools, ensures a judicious balance between technical compatibility, practical feasibility, and cost-effectiveness in the overall design process.

Once the schematics were made using Altium Designer software, they were used to create a linking PCB project. Components were placed in a strategic way to allow for routing to run over and under the board through vias placed. The program offers design rule checks that were used to validate the results.

After placing all components on the top layer, the mounting component was strategically positioned on the bottom layer. This decision was deemed essential to prevent the obscuration of designators for the connectors to the components once mounted. Subsequently, a dedicated layer was generated to delineate the PCB's outline. While opting for a compact board for enhanced handling convenience, careful consideration was also given to prevent components from clustering together, potentially leading to overheating. To ensure adequate support beyond the mounting component, drill holes were deliberately added to each corner of the PCB, facilitating additional securing mechanisms for its placement within the designated compartment.

Stackup

#	Name	Material	Type	Weight	Thickness	Dk
	Top Overlay		Overlay			
	Top Solder	Solder Resist	Solder Mask		0.01016mm	3.5
1	Top Layer	CF-004	Signal	1oz	0.035mm	
	Dielectric 1	FR-4	Dielectric		0.32004mm	4.8
2	Bottom Layer	CF-004	Signal	1oz	0.035mm	
	Bottom Solder	Solder Resist	Solder Mask		0.01016mm	3.5
	Bottom Overlay		Overlay			

Figure 8.2

The Stackup table presented above illustrates the distinct layers comprising the PCB, predominantly characterized by a top and bottom layer demarcated by a dielectric material. Each layer is clearly delineated, indicating both the material composition and its corresponding thickness. Notably, a deliberate choice was made to adopt a thickness of 1.378 mil for both the top and bottom layers. This decision aligns with established recommendations for 2-layer PCBs, specifically accommodating 1 ounce of copper. This particular thickness strikes a balance conducive to the optimal performance and cost considerations inherent to such configurations.

Top Layer

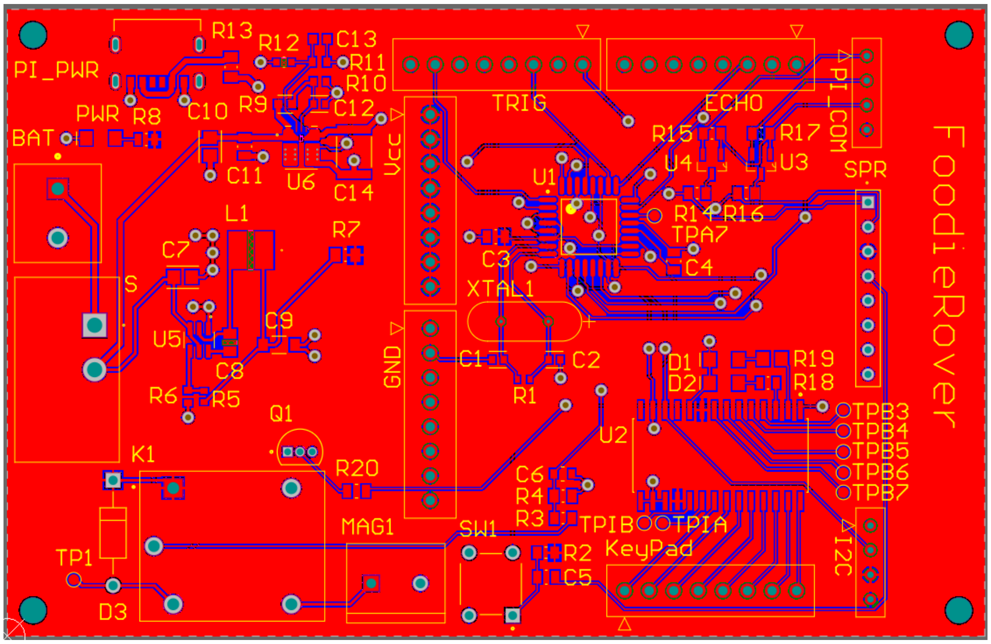


Figure 8.3

Bottom Layer

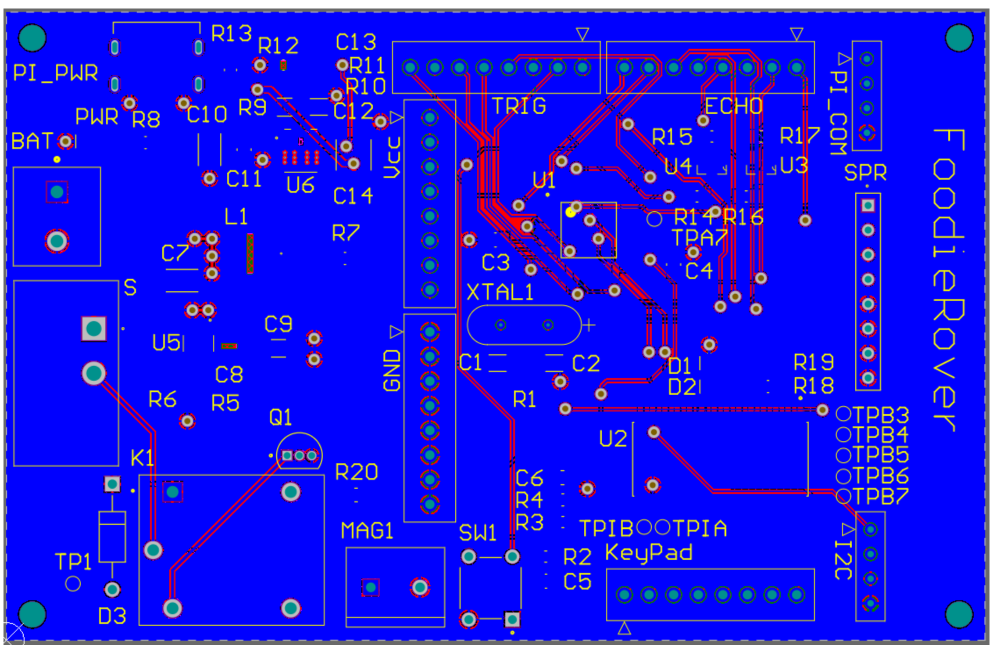


Figure 8.4

The layered design of the FoodieRover's PCB, with dedicated power (Vcc) and ground layers, is a strategic choice that enhances the board's electrical performance and integrity. The Vcc layer ensures that there is a consistent and robust power delivery to all the components, which is especially important for the microcontroller and power-hungry peripherals like the Raspberry Pi.

Having a separate ground layer serves multiple purposes. It acts as a return path for current, it helps with noise reduction by providing a reference for the entire circuit, and it can minimize electromagnetic interference (EMI), enhancing the PCB's signal quality. The ground plane also helps with thermal management, acting as a heat spreader across the board's surface.

In high-speed or complex circuits, maintaining signal integrity and reducing EMI is crucial. This kind of PCB layout, where power and ground are given their own layers, creates a low-inductance path for signals and helps with decoupling, making it a design seen in high-quality electronics.

The use of these layers in the FoodieRover PCB shows careful attention to the electronic design's reliability and functionality, ensuring that the system can operate efficiently and safely in various conditions. It's a testament to the thoughtful engineering work that has gone into developing a robust platform for the FoodieRover.

Altium allows you to see a 3D model of the PCB board. This 3D view feature is instrumental in gaining a more immersive understanding of the physical layout and spatial arrangement of components on the board. A dynamic 3D view facilitates a holistic perspective that aids in design verification, component placement optimization, and overall system comprehension.

<u>Top View</u>

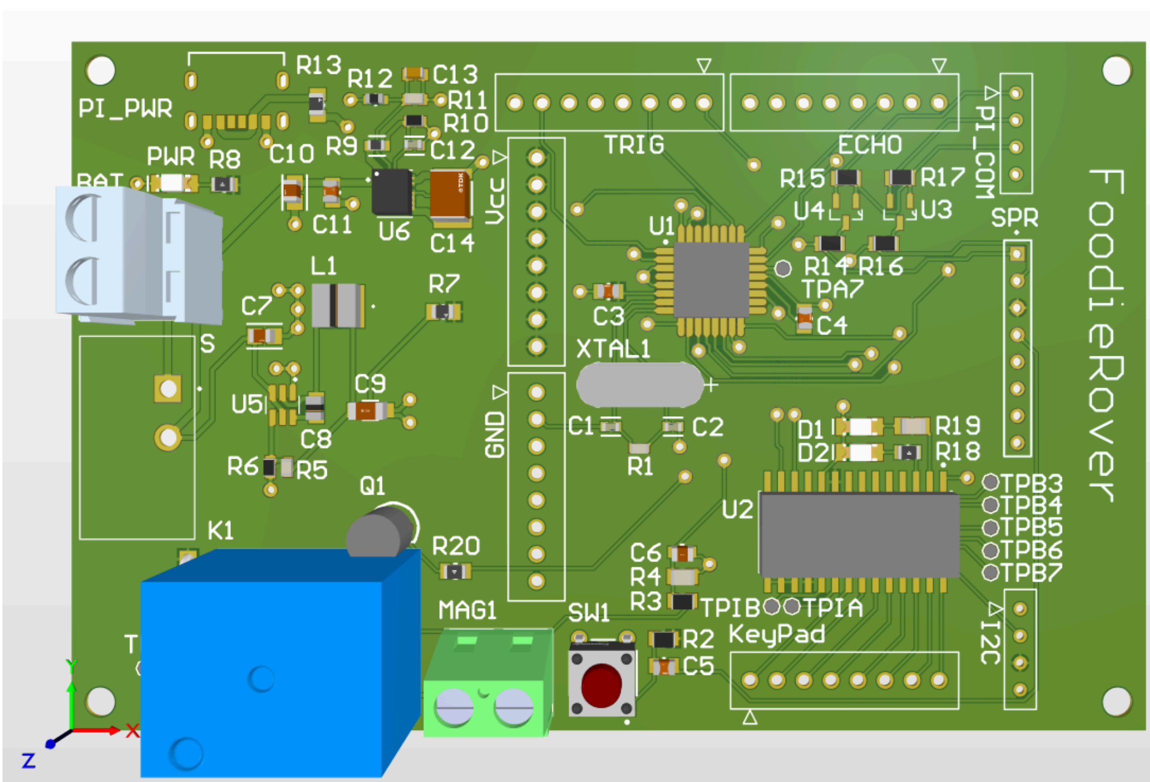


Figure 8.5

In the three-dimensional model, it is discernible that right-angle connectors have been employed, and notably, they do not extend all the way to the edge. This deliberate design choice is intended to enhance the structural integrity of the connections, considering the multitude of essential connections present. The strategic utilization of right-angle connectors ensures a robust and well-supported arrangement, addressing potential stress points and contributing to the overall durability of the interface.

Furthermore, it is noteworthy that additional ground pins have been incorporated into the design. This precautionary measure is implemented to account for scenarios where the standard allocation of ground pins might be insufficient. The inclusion of surplus ground pins serves as a contingency, offering flexibility and adaptability in scenarios where an expanded grounding infrastructure may be required. This meticulous approach to connector design aims to optimize both the mechanical and electrical aspects, aligning with best practices in ensuring a resilient and versatile connection system within the electronic assembly.

Bottom View

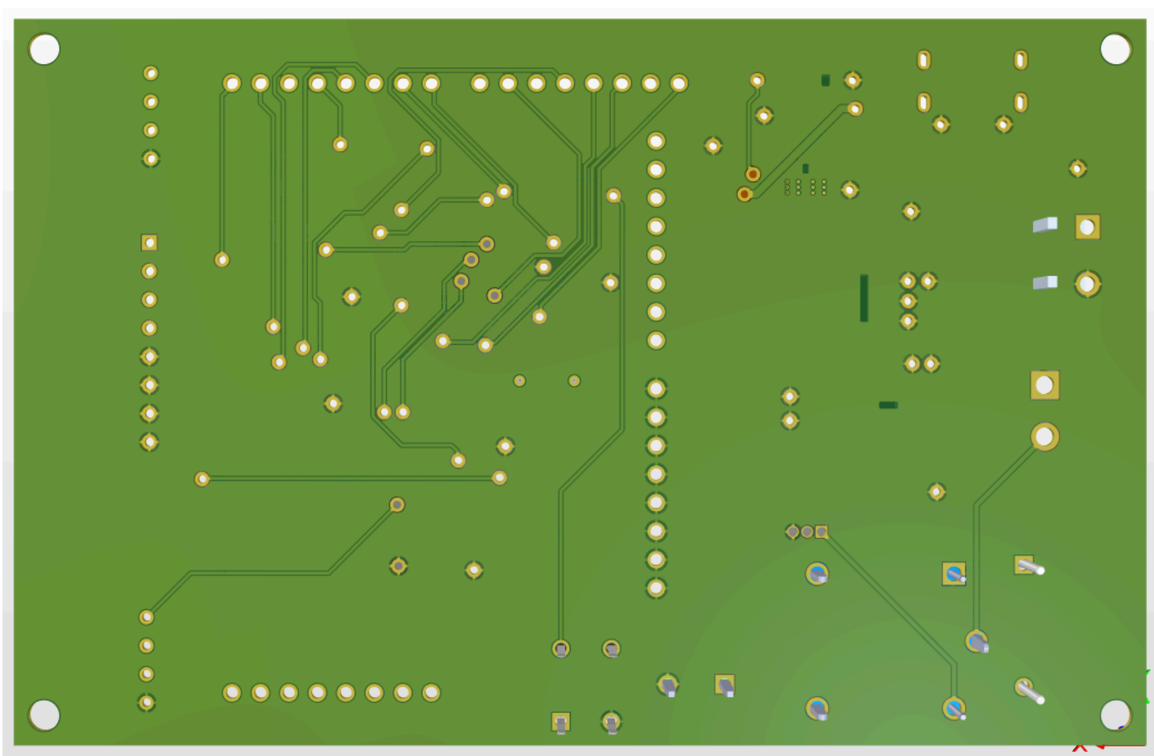


Figure 8.6

Within the 3D view, Altium enables users to explore their PCB from various angles, rotate it, and zoom in or out for detailed inspections. This capability is particularly valuable in identifying potential interference issues, checking for component collisions, and ensuring that the design aligns with the intended mechanical constraints. Getting this type of visualization ultimately contributes to the efficiency and accuracy of the design process, ensuring that the electronic system is not only functionally sound but also physically optimized for manufacturing and assembly processes.

Prototype

The first version of the FoodieRover PCB, depicted here, marks a significant step in the project's evolution. This prototype encapsulates the initial concepts and design approaches we took in integrating the various components necessary for the rover's operation. It served its purpose by allowing us to assess the spatial requirements and identify areas where space optimization could be improved. Notably, the design featured more through-hole components, which, while easier to solder and handle, contributed to the PCB's larger footprint.

We encountered challenges such as an inadequate number of ground connections and the presence of redundant components that, while not detrimental to functionality, presented an opportunity to streamline the design. The learning curve was steep, but invaluable lessons were learned regarding component selection and PCB layout efficiencies. This prototype was fully operational and integrated into the system, providing a solid foundation for refinement.

Drawing on this experience, the subsequent PCB iteration was redesigned with a focus on space efficiency, surface-mount technology (SMT) to reduce size, and the removal of superfluous elements. The outcome was a more compact, elegant, and efficient PCB that better suits the spatial constraints of FoodieRover, demonstrating the benefits of iterative design and continuous improvement.

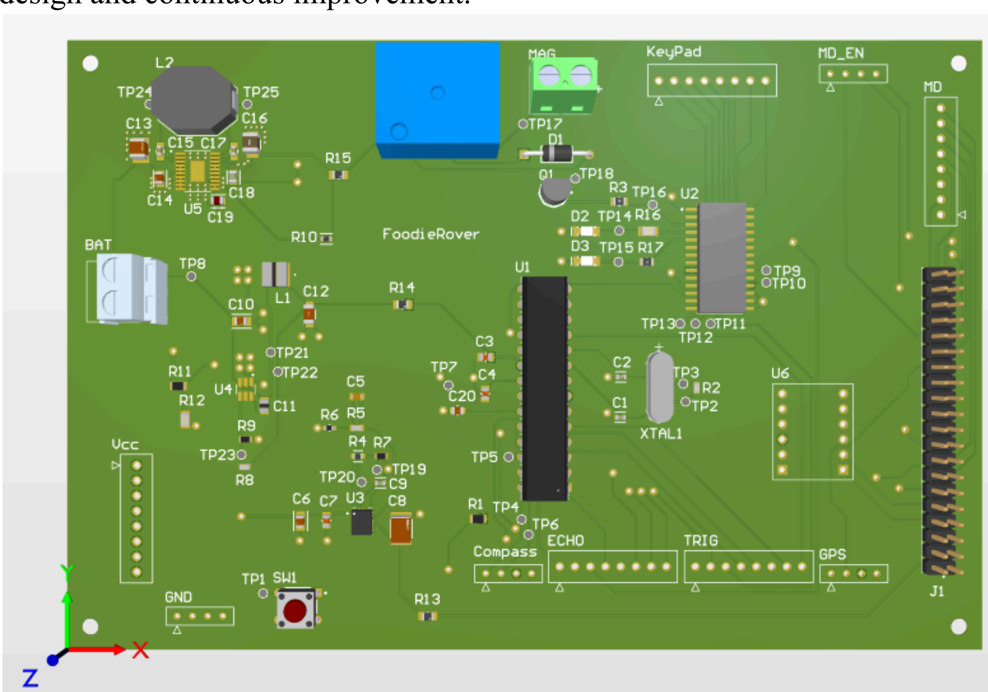


Figure 8.7

9. System Testing

Testing Compass Module 3-Axis HMC5883L

The sensor module detects low magnetic fields and translates its readings into differential voltage. Its capacity to register low fields enables it to perceive the Earth's magnetic field. The digital interface outputs voltage on three axes, facilitating the computation of headings. In our application, the module will be employed for robotic navigation. To assess the module's functionality as a compass, both its hardware and software will undergo testing.

Hardware:

The hardware testing involves interfacing the module with an Arduino Uno board. Our primary focus is to evaluate its I2C communication and reading accuracy; hence, the use of the Raspberry Pi 4 board is currently unnecessary. The module should be capable of detecting field strengths in the micro-Tesla range to measure the Earth's magnetic field. Operational requirements include a supply voltage, ground connection, and two I2C lines, with no need for external hardware.

Upon unpackaging, the module requires pin soldering. A concern arises regarding the potential impact of ferromagnetic pins on reading accuracy; this will be tested. If found relevant, the ferromagnetic pins will be replaced with copper pins, a non-ferromagnetic material. Subsequent testing will assess performance concerning proximity to other hardware, live wires, and movement.

Using a breadboard, the SCL, SDA, supply voltage, and ground were connected to the module via jumper cables. The wires were intentionally positioned as far as possible from the module and connected to the microcontroller. However, inaccurate readings resulted, likely due to significant magnetic fields generated by nearby current in the breadboard.
Upon revisiting the connection, the module was elevated using long wires, resulting in much more accurate readings. The program yielded readings accurate within five degrees, deemed acceptable. Given the module's sensitivity to nearby magnetic fields, it was decided, based on testing, to solder copper pins and position the module far from any components causing fluctuations in accuracy. Additionally, placing the module at an elevated position replicated the optimal results observed during testing.

Software:

The software uploaded to the microcontroller reads data from I2C communication, measures magnetic field strength in micro-Tesla units, and converts the collected Tesla values (in x, y, and z directions) into headings in degrees. The code utilizes libraries such as Wire for I2C and applicable Adafruit libraries for the HMC5883 module.

After initializing serial communication and detecting the module, the code employs a function to retrieve and print information, including the device ID, maximum and minimum magnetic values, and resolution. In the main loop, the function repetitively reads from the module for the x, y, and z axes. For each iteration, heading calculation is performed using arctangent to obtain radians. Compensation using the third axis, z, determines the difference between true north and magnetic north. Adjustments are made for negative values, and the heading is finally converted to degrees, printed to the serial monitor, with a 0.5-second delay for the next reading.

Testing Ultrasonic Distance Sensor HC-SR04

The HC-SR04 module facilitates non-contact distance measurements within a range of 2cm to 400cm or 4 meters. A thorough evaluation of this module is essential to ascertain its accuracy and reliability, as the functionality of FoodieRover relies heavily on the precision of the readings obtained from these modules. Furthermore, comprehensive testing, including assessments of the measurement angle and identification of potential inaccuracies, contributes to the enhancement of both hardware and software design.

Hardware:

To assess the hardware, an Arduino Uno was utilized, prioritizing the performance evaluation of the HC-SR04 module. The Raspberry Pi was excluded from the testing as it was not deemed necessary. The module was securely placed on a breadboard to ensure stability, maintaining a 90-degree angle to the horizontal surface. Four connections were established: the module connected to a 5V supply voltage, ground, and two GPIOs for trigger (output) and echo (input) signals. Multimeter measurements confirmed normal module operation, with the supply voltage close to 5V, trigger pin at 1.33V, and echo pin at 0.5V. It was observed that operational noise emanates from the module during its functioning.

The setup was employed to examine the module's range, angles of interest, and its response to different object sizes. Results indicated that larger objects, equivalent to the size of a person or book, were detected more reliably. Smaller objects, smaller than a book, were detected at greater angles when placed on a flat horizontal surface. Introducing a towel for a rougher surface rendered readings at greater angles insignificant. This observation is significant as it reflects the conditions that FoodieRover may encounter. Some readings were identified as inaccurate and were disregarded, highlighting the necessity for a higher volume of readings for improved accuracy. Accurate readings were found to occur at an angle of around 20 degrees, subject to variations based on conditions such as the proximity of larger objects and the texture of the objects. Notably, the most accurate readings were obtained at closer distances, reinforcing the preference for relying on readings closer to the module for increased certainty, particularly regarding objects in close proximity to FoodieRover.

The module setup was modified to incorporate LEDs with resistors, serving as indicators for object detection. Connected to the board's GPIOs, the LEDs facilitated the establishment of arbitrary distances of interest. Rapidly moving an object parallel to the module was effectively detected through LED flickering, demonstrating the module's capability to pick up objects moving at a speed of approximately 8 miles per hour with significant accuracy.

Software:

Understanding the software used for testing requires familiarity with the functioning of the ultrasonic sensor. The sensor operates by sending a high-level trigger signal of 10 microseconds, followed by eight cycles of a 40kHz sonic burst to detect the reflected pulse signal by raising the echo. The distance is then calculated based on the time of travel from ultrasonic to return. The formula involves multiplying the high-level time by the speed of sound and dividing by two. The code performs this calculation iteratively to interpret the distance readings. The variables 'duration' and 'distance' store information from the readings and perform the necessary calculations.

Within the loop function, the trigger signal is sent using 'digitalWrite' to the corresponding pin. The echo pin is then utilized to measure the duration of the pulse. The calculation is performed, and the results are printed to the serial monitor. If statements, controlled by thresholds determined by arbitrary distances, interpret the results. If the calculated distances are less than or equal to the threshold value, the corresponding LED is illuminated; otherwise, it remains off.

Testing of Robot Chassis

The T900 Robot Tank Car Chassis is a sophisticated piece of machinery designed for various applications, integrating 4 DC encoder motors, 4 plastic tracks, 5 metal frame components, 40 bearings, and an assortment of screws and other components. Extensive testing protocols are employed in the evaluation process to guarantee the built chassis's best performance and dependability.

Introduction:

The T900 Robot Tank Car Chassis represents a versatile platform for robotics enthusiasts, offering a sturdy construction and a variety of components for diverse robotic projects. The testing process aims to check the individual parts and their collective performance under different conditions.

Assembly Integrity Check:

Before commencing the tests, a crucial step involves verifying the assembly integrity. Even if the pieces weren't assembled when they arrived, careful assembly is necessary to guarantee correct operation. Every motor, track, frame part, bearing, and screw kit was meticulously put together in compliance with the manufacturer's instructions. To verify that all parts were securely attached, components were aligned, and the assembly was done correctly, a comprehensive inspection was carried out.

Load and Stress Testing:

To evaluate the durability and load-bearing capacity of the chassis, stress tests were conducted. Various weights were incrementally added to the chassis to assess its ability to withstand and maneuver under different loads. Stress testing involved both static and dynamic loads to simulate real-world scenarios where the robot might encounter uneven terrain or carry additional equipment.

Component Reliability Assessment:

The reliability of individual components such as tracks, bearings, and frame parts was evaluated separately. The tracks were subjected to abrasion tests on different surfaces to determine their wear resistance. Bearings underwent rotational tests to examine their smoothness and longevity under continuous operation. Frame components were assessed for structural integrity and resistance to deformation under stress.

Quality Assurance:

A meticulous inspection was carried out to identify any potential issues such as loose connections, wear and tear, or manufacturing defects. Additionally, the screw kits and other parts were examined for compatibility, completeness, and quality.

Conclusion:

The testing process of the T900 Robot Tank Car Chassis involved a comprehensive evaluation of its assembly, functionality, durability, and component reliability. The results indicate a well-assembled and robust chassis capable of performing various tasks. However, continuous monitoring and maintenance will be essential to ensure prolonged and efficient operation in practical applications. This assessment provides valuable insights for both end-users and manufacturers to enhance the design and performance of future iterations of the T900 chassis.

<u>Testing of DC Motor</u>

The evaluation involves a careful examination of the motors' speed, torque, current consumption, and response to varied voltage inputs within the specified parameters. By analyzing these attributes, this study aims to provide a comprehensive understanding of the motors' capabilities and their role in driving the robot chassis efficiently.

Introduction:

The DC Encoder Motors within the T900 Robot Chassis are vital components responsible for enabling movement and maneuverability. Understanding their performance parameters, such as speed, torque, and response to varying electrical inputs, is essential for optimizing the robot's functionality across diverse applications.

Methodology:

The experimental procedure involved a systematic setup wherein each DC Encoder Motor was mounted onto the chassis, connected to a wheel to simulate practical movement, and interfaced with a motor driver for controlled power regulation. Voltage inputs were incrementally adjusted within the specified working range of 3.5-12V, and corresponding current levels were monitored to analyze the motors' speed and torque behavior.

Results and Analysis:

The 25mm gear motors exhibited a diverse range of performance characteristics. In-depth analysis revealed a direct correlation between the applied voltage/current and the rotational speed of the motors. As the voltage within the specified range increased, the motors' rotational speed exhibited a proportional rise. Similarly, the torque output showed a corresponding increase concerning the current supplied to the motors.

Moreover, the noise emission during motor operation was not to loud, indicating relatively quiet performance suitable for various applications and environments.

Conclusion:

The comprehensive testing and analysis of the 4 DC Encoder Motors in the T900 Robot Chassis yielded valuable insights into their performance characteristics. These motors showcased adherence to specified speed, torque, and current parameters across varying voltage inputs, establishing their capability to efficiently drive the robot chassis within the designated voltage range.

The detailed data obtained from this analysis holds significant promise for optimizing motor control strategies, enhancing the overall functionality, and providing guidance for potential improvements in future iterations of the T900 Robot Chassis. This deeper understanding of the motors' operational attributes is critical for leveraging their capabilities across a spectrum of robotics applications.

Testing of Motor Driver

This section presents an extensive evaluation of the L298N Dual H-Bridge Motor Driver, a pivotal component in the control of DC motors. The testing methodology involved a detailed setup process, integrating the motor driver with DC motors, a +5V Arduino, and specific Arduino code to assess its capabilities. By carefully following wiring instructions and implementing code, this study scrutinizes the motor driver's potential, enabling independent control of two motors, each capable of handling up to 2A, in both forward and reverse directions.

Introduction:

The L298N Dual H-Bridge Motor Driver, leveraging the widely used L298 Integrated Circuit, serves as a critical element in robotics, automation, and various electronic applications. This section aims to conduct an in-depth analysis to elucidate the operational functionalities of this motor driver, crucial for precise control over the speed and direction of DC motors.

Methodology:

The testing methodology involved a step-by-step setup procedure to ensure accurate interfacing between the L298N module, DC motors, and the Arduino platform. Initially, the wires of the stepper motor were identified and connected to their corresponding module connections: A+ to 1, A- to 2, B+ to 13, and B- to 14. This precise wiring is fundamental in ensuring proper motor operation.

The module configuration was completed by placing jumpers over the pairs at module points 7 and 12, followed by connecting the power supply to points 4 (positive) and 5 (negative/GND) of the module.

Connections between the L298N module and the Arduino were established. Pins IN1, IN2, IN3, and IN4 of the module were linked to digital pins D8, D9, D10, and D11 on the Arduino, respectively. Additionally, to maintain a stable electrical connection, Arduino's GND was connected to point 5 on the module, and Arduino's 5V was linked to point 6 because sourcing 5V from the module was necessary.

For validating the motor driver's functionality, the Arduino code "stepper_oneRevolution" was uploaded to the Arduino. This code facilitated the execution of a single revolution by the stepper motor in one direction, followed by a return, effectively demonstrating the motor driver's precise control over motor movement.

Results and Analysis:

The setup process and execution of the "stepper_oneRevolution" code resulted in successful motor control using the L298N Dual H-Bridge Motor Driver. The demonstration showcased the driver's capacity to effectively control the connected DC motors, executing a complete revolution in one direction and then returning, affirming its accuracy in managing motor speed and direction.

The controlled movement of the motors validated the functionality of the motor driver in maintaining synchronization and precise control, ensuring smooth and accurate motor operations.

Conclusion:

The comprehensive testing and successful demonstration of the L298N Dual H-Bridge Motor Driver affirmed its capability to independently control two DC motors with precision and reliability. The systematic setup process, combined with the utilization of specific Arduino code, provided a clear depiction of the motor driver's proficiency in managing motor speed and direction, solidifying its relevance in diverse applications within robotics, automation, and electronic systems.

This detailed analysis offers valuable insights into the operational efficiency and reliability of the L298N motor driver, emphasizing its crucial role in various motor control systems and highlighting its potential for further advancements and applications in the realm of robotics and electronics.

Testing Keypad Module

This section outlines the procedures carried out in order to establish confidence in the proper operation of the 4x4 Keypad Module. This generic membrane keypad unit is developed to read input from a user through a series of 4 rows and 4 columns of soft-shell buttons, totaling 16 unique buttons. These 16 potential input values include digits 0 through 9, letters A through D, an asterisk (*), and a pound sign (#). This component will be utilized in the FoodieRover system to read input from the delivery recipient, with the purpose of controlling the locking and unlocking of our electromagnetic lock security feature. Upon reaching the delivery destination, the consumer will input the designated passcode, the detection of which will initiate the process, via the microprocessor, of removing supply power from the locking module, thus releasing the applied holding force and allowing the consumer to access their delivery materials.

Hardware:

The physical Keypad Module unit is a flexible membrane, roughly 2 inches long by 2.5 inches tall, containing 4 rows and 4 columns of slightly elevated buttons, with each button delineating the digit, letter, or symbol that it represents. This clarity is ideal given that this component will be externally attached to the robot to serve as an important part of the user-system interface, so there is no need for additional functional or aesthetic adjustments. Additionally convenient is the removable adhesive paper on the back side of the keypad membrane, eliminating the need to consider mode of attachment to the FoodieRover (unless the integrity of this adhesive is deemed unsuitable).

The keypad comes equipped with a long, thin attachment housing the internal wiring needed for the transfer of data between the keys and the external interface. This attachment terminates with 8 female-type attachment ports. To interconnect these GPIO ports to the microcontroller GPIO ports specified in the GPIO Expander Schematic detailed previously, 8 male-to-male type connectors will be utilized.

For the purposes of testing the physical keypad hardware, these 8 attachment ports were connected via male-to-male jumper ports to GPIO ports D2 through D9 of the Arduino Uno development board, given that this was the unit accessible at the time of testing, and that the functionality of the keypad can be evaluated equally as proficiently as with the Raspberry Pi microprocessor that we plan to implement in the final system. These attachments were secured without issue, verifying that the ports included in the keypad unit are compatible with the wiring techniques we intend to use and that none are blocked or otherwise misshapen.

The buttons on the keypad membrane are soft in material but somewhat stiff to press, necessitating the need to apply intentional but not distressing or excessive force. Additionally, applying pressure roughly to the center of each button is much more likely to result in an accurate reading than attempting to press the button on any side or corner. Confirmation that a button has been sufficiently pressed is assured through a noticeable “clicking” feeling and sound, providing confidence that most users should be fully capable of operating the unit properly.

Verification of proper operation of the internal hardware system of the keypad was achieved through the correct identification of button presses for each of the 16 unique buttons in the software test. This ensures that the valid designated digital signals were sent through the GPIO connections in each of the 16 test cases.

Software:

The software aspect of the keypad system integration encompasses the function of reading digital data received from the keypad’s output pins and determining the key that was pressed by the user as a function of the status of these 8 pins. The 8 GPIO pins correspond to one row or column each, allowing enough combinations of received data to identify each of the 16 possible buttons. On the software front, this is achieved through a process called keypad scanning. Each row and column pin is set to HIGH (5V or 3.3V) by default and can be connected to GND through the pressing down of internal resistor components within each button, setting the corresponding column and row pins to LOW (0V). Thus, keypad scanning works by recognizing which row and column pin is LOW during each press and using the intersection of the two to properly identify the button at that location.

The software program implemented to identify the button presses defines the array of button display characters in a 2x2 character array “keys” delineated by the row and column identifying numbers. ROW_NUM and COLUMN_NUM arrays are also defined by the order of the GPIO pins corresponding with the numerical order of rows and columns. The program contains a loop that continuously identifies the “key” character returned by scanning the GPIO pins for the non-active row and column, and applying this information to the “keys” array to return the character on the pressed key.

Upon testing of each of the 16 individual buttons via these methodologies, the correct character value associated with the pushed button was returned each time to the Arduino IDE Serial Monitor. This successful result verifies that the hardware of the keypad is operating properly, the connections are adequately interfaces, and the software applied is equipped to handle the information and return the desired values. Thus, the keypad module has successfully undergone the testing process and is prepared for integration with the FoodieRover system.

Testing Electromagnetic Lock and Relay

The AGPTek Electromagnetic (EM) Lock component serves as the principal manner of securing the delivery container component while the system is en route. This unit is designed to utilize incoming applied power to generate a magnetic field in the electromagnetic component, which, when active, attracts an associated metal armature plate with a holding force of 60 kg (130 lbs). Affixing these components to the lid and body of the delivery container will guarantee that no one can access the meal being delivered until the proper passcode is entered into the keypad, deactivating the lock and removing the holding force. The Lock Relay is an important intermediary between the EM Lock and the Microprocessor given that the lock operates at a voltage of 12 V, which cannot be directly interfaced with the Microprocessor without burning the circuitry. Additionally, the Relay provides the control mechanism for activating and deactivating the lock through digital information received from the processor. For this reason, the Relay is wired on one side to the Microprocessor for control of status, and wired on the opposing side to the 12V power source and the electromagnet for resulting passing or blocking of power to the unit. This section will detail the testing of each of these interconnected components individually, as well as in connection to verify the interface and software control of the locking system.

Hardware:

As mentioned, the EM Lock is composed of an electromagnet and an armature plate (as well as metal screws and other affixing materials). Upon inspection of the hardware units, each one appears to be as described during component selection, with a positive and negative wire emerging from the electromagnet for interconnection with a power source. The armature plate is simply a small rectangular metal box with a hole in the center for attachment to the physical structure.

Prior to involving the Lock Relay, the functionality of the EM Lock components was tested via connection with a 12V DC Power Source through a breadboard. Upon application of 12 Volts at the positive wire of the electromagnet, with the negative wire affixed to a ground node, the armature plate was drawn into contact with the magnet. This connection was of sufficient strength as to not allow separation of the components with the application of physical force from the team members, thus verifying that the EM Lock components themselves operate as expected.

The Lock Relay component was also tested individually through the measurement of input and output voltages at the respective nodes, based on the proposed interconnections with the Microcontroller and the electromagnet. On the input side, DC+ and DC- were wired to 5V and GND nodes, respectively, on a breadboard. On the output side, the COM (Common) pin was wired to a 12V node, representing the 12V power source. When the IN pin was also wired to the 5V node, representing the ON status of a GPIO pin, the voltage at the NO (Normally Open mode) node was measured as 12V. Thus, the electromagnetic connection would successfully be activated. When the IN pin was wired to GND, there was no voltage measured at the NO node, representing the lack of activation of the EM Lock under these conditions. Thus, the Lock Relay works as expected and as required to allow for control of the locking mechanism.

Software:

To evaluate the overall locking system and the control capabilities obtained through the relay, the components were wired together as planned in the overall design and a software program was developed to verify that locking and unlocking procedures could be completed via microprocessor control.

The positive terminal of the EM Lock was wired through the Relay's NO and COM pins to the 12V power source. The Relay's input pins DC+, DC-, and IN were wired to 5V, GND, and GPIO nodes of the Arduino Uno development board, respectively. A simple software program was then developed to test the ability to lock and unlock the system via a microcontroller. The program initialized the GPIO pin connected to the Relay as an output pin and then entered a loop that switches this pin's value between HIGH and LOW with a small delay to allow for observation. If implemented correctly, this loop code should result in the electromagnet being activated and deactivated roughly every 3 seconds, and thus the system should be locked and unlocked following the same pattern.

It was determined that this program operates as expected, with the lock being active while the GPIO was set to HIGH and non-active while the pin was set to LOW. Thus, it has been confirmed that a simple programming block such as this program can be applied to a microprocessor in order to switch the status of the locking system between being locked while undergoing delivery procedures to being unlocked at all other times. Evidently, this code will not exactly mirror the final version, which will activate and deactivate the locking system not on a time-based schedule, but rather activate it when a delivery is initiated and deactivate it when the unlocking passcode is identified from the keypad.

The successful application of this program and observation of correct results in the hardware components permit the conclusion that the EM Lock and Lock Relay components function as expected, that the interfaces between components are valid and compatible, and that software control of the system's status is not only possible but even relatively simple. Consequently, these components are deemed fit for the next stage of advancement in the development and implementation of this product.

10. Administration

In this chapter, we delve into the critical aspects of our project management strategy. First, we discuss the costs incurred for all components purchase and the allocation of funding and budget management, as financial resources are a cornerstone of project success. Next, we outline the key milestones that will mark our journey towards project completion, ensuring that progress is tracked and objectives are met in a timely manner. Finally, we address the distribution of work and responsibilities among team members, optimizing our collective efforts to achieve project goals efficiently and with the highest quality.

10.1 Budget and Funding

This section will summarize the cost analysis of the components that have been identified and bought for this project implementation. Prices will be updated as components are bought in order to ensure the final cost summary is accurate to all purchases made by team members and the resulting total costs can be divided in an equal and fair manner.

ITEM	QUANTITY	PRICE ESTIMATE	TOTAL COST
Camera	4	$14 for 6	$14
Magnetometer	1	$10	$10
GPS module	1	$25	$25
Motors*	2	$25	$50
Robot chassis	1	$170	$170
Microprocessor	2	$30	$60
Motor driver	2	$15	$30
Ultrasonic sensor	8	$9 for 5	$18
Key pad	1	Acquired	$0
Power source	1	$40	$40
Magnetic lock	1	$20	$20
Cooler	1	$30	$30
API	1	$30	$30
		Total cost:	$497

Figure 10.1

* Many chassis' come pre-equipped with motors.

We expect the funding for this project to come from self-funding from the team members. Team members will purchase the design components that are needed for the completion of this project and will make an effort to equally share the costs incurred among all members for fairness.

10.2 Project Milestones

This section summarized the key milestones that are part of this project, from brainstorming project ideas to researching, purchasing, and testing components, and finally consolidating the components into the designed system for testing and modifications when necessary. Each milestone represents a significant stride towards our goal of making a revolutionary food delivery device. This layout of important events will help our team stay organized and on schedule across the Fall 2023 and Spring 2024 semesters so as to not fall behind on important dates and deadlines.

Project Milestone					
Task	START DATE	DUE DATE	%COMPLETE	DONE	NOTES
Group Discussions	9/5/23	5/1/24	100%		Group meets every week to talk about project
Parts for Prototype Ordered	11/5/23	11/20/23	100%		Parts for Prototype ordered
Parts for Prototype Received	11/10/23	12/2/23	100%		When the Parts are received
Testing of Products	11/20/23	12/2/23	100%		Time for which the Robot will be built
Assembly of Parts	12/3/23	1/20/24	100%		First wave of tests for robot
Changes to Prototype Made(1)	1/20/24	2/13/24	100%		Learned from mistakes from first tests
Second Tests	2/14/24	2/18/24	100%		Second wave to text changes made from first mistake
Changes to Prototype Made (2)	2/18/24	2/22/24	100%		Learned from mistakes from second test
Third Tests	2/26/24	3/11/24	100%		Third wave to test changes made
Final Changes Made	3/12/24	3/20/24	100%		Last minor changes made for finished product
Last Tests	3/21/24	3/31/24	100%		Final test for prototype
Final Demonstrations	4/1/24	Senior Design demonstration day	100%		Demonstration of robot

Figure 10.2

10.3 Work Distribution

The following table delineates the allocation of work among the members of Group 29 for a specific component of the FoodieRover design, encompassing both hardware and software aspects. Each member is responsible for all related concepts within their assigned component. The distribution of responsibilities is rooted in the individual interests and strengths of the team members.

While the primary member shoulders the majority of the responsibilities for each component, a secondary member is designated to provide assistance when the workload becomes burdensome for the primary member. The secondary member is expected to maintain familiarity with the project's progress and offer support when required.

Work Distribution		
Component	Primary Member	Secondary Member
Camera	Mauricio	Antonia
Communication Module		Alexis
Software		Everyone
Frame/Motors	Chidoziri	Alexis
Magnetometer		Alexis
Distance Sensors		Antonia
Microprocessor	Antonia	Mauricio
Motor Driver		Chidoziri
Navigation Module		Mauricio
Locking Mechanism	Alexis	Chidoziri
Power Source		Chidoziri

Figure 10.3

11. Summary and Conclusion

The concept of the FoodieRover integrated system design originally emerged from the need for students on college campuses to optimize their time management with regard to accessing meals from on-campus vendors during breaks between important classes and other relevant obligations. Many students struggle with laborious schedules, and the time devoted to obtaining a meal from busy food vendors distracts from educational commitments and sometimes results in skipping meals altogether. This common struggle among students could be alleviated with the application of an intelligent meal delivery service, such as the one being developed from the design outlined in this report. The successful deployment of such a system on a large scale would relieve stressors frequently encountered by students who understand the value of the time they spend each day walking to and from food vendor locations and waiting in line, and how this time could be better spent on more fulfilling and pressing matters.

The most crucial design objective that encapsulates the scope of this project would be the robot's self-navigation capabilities. It is this feature that permits a food delivery system to reduce the workload of the users involved without requiring more work for another person in terms of delivering the order to the specified location. Self-navigation in the FoodieRover system arises from an array of complex and technologically advanced subsystems operating in a fully integrated manner to exchange relevant information with a carefully innovated software analysis program, brought to fruition by an intelligent central microprocessor unit.

The features requisitioned by the team during the initial design stage are to be implemented with the help of many peripheral units that communicate data information for analysis and reconfiguration of status and behavior when necessary. The major auxiliary units employed include ultrasonic distance sensors, a GPS navigation module, a magnetometer, and a camera, which are used for navigation, obstacle detection, and rerouting functionalities. Another significant aspect of this system is that it is designed with security in mind, prompting the inclusion of the electromagnetic lock, lock relay, and keypad units that will require the delivery container to remain securely closed until a specified passcode is input by the ordering user.

The current stage of the design process outlined within this report document encapsulates a meticulously detailed and thoughtfully considered blueprint which serves to guide the design team in applying the ideas set forth for the realization of the fully operating system in the coming months. Each physical and digital aspect of this robust design has been thoroughly researched and considered in conjunction with its associated components to ensure that not only is the most applicable item being chosen for each given task, but also that each item is capable of interconnection with the others without compromising function or safety in any way.

Several additional factors have been investigated by the design team in relation to the development of this delivery system. Methodologies implemented by similar delivery systems designed by others have been considered and compared with those put forth by the team in order to define similarities and differences between each, as well as to gain a high level of understanding of the related technologies that exist on the market today. Related standards and other constraints have been researched to develop a robust comprehension of the limitations which this type of device faces, not only technologically, but socially, ethically, and economically. These fields of research have yielded a multitude of additional considerations that are intricately interwoven into the aggregate system design, including factors of realistic manufacturability and digital information security.

After considerate selection of the hardware and software elements to be incorporated into the FoodieRover system design, the development team detailed the interconnections between these components through the creation of detailed hardware and software system design documentation. Among hardware components, units which are to be interfaced together to serve a common purpose are grouped into subsystems, where their complex interconnections are described thoroughly. These include the Navigation Subsystem, which champions the efficient and accurate routing of the robot along the specified delivery path, and the Locking Subsystem, which provides the user access control platform to unlock the secure container that stores their order. This section of the report provides several well crafted schematics detailing the physical connections between each component. Careful considerations of these interfaces gave rise to the development of a sophisticated PCB Board design, described in detail in Chapter 8. The operating software functions are also illustrated to gain an understanding of how each directive envisioned is to be realized through deliberate programming of the microprocessor unit. This section details how the processor is to attain the many desired operations, such as the procedures used for gathering and evaluating information from the several peripheral components including the Ultrasonic Sensors and the GPS Module.

At this point in time, the prototype of the FoodieRover design has been successfully constructed and tested for all major design features. The integration of all hardware components and software programming allowed for the robot to demonstrate advanced capabilities in mobility, self-navigation, obstacle detection, and security. The final design was able to successfully move at the desired speeds, move and turn in all directions, use API connectivity to obtain extremely precise location data and route to the destination coordinates, detect objects in its movement path, and implement a working locking mechanism. FoodieRover has shown incredible potential for the successful deployment of an on-campus self-navigating food delivery system.

With the world becoming more automated than ever, it is certainly not an unrealistic thought that college students may one day experience the luxury of sitting down at their favorite campus study spot and ordering a meal to be delivered to them automatically, maximizing the efficiency of the time which they devote to education and eliminating the wasted time of traversing the campus grounds and loitering in line at the ordering counter. Wide-scale applications of intelligent, automated, self-operating systems such as FoodieRover would increase convenience and decrease stressors experienced by students, and would overall make college campuses more efficient places to spend purposeful, goal-oriented time.

12. Appendix A - References

[1] 2013, K. K. 8. (2019, April 17). Basic robotics - power source for robots. AZoRobotics.com. https://www.azorobotics.com/Article.aspx?ArticleID=139

[2] AdobeFirefly (2023). "Design the most efficient small sleek but ..." firefly.adobe.com https://firefly.adobe.com/public/t2i?id=urn:aaid:sc:US:491ded12-6bf1-4694-88eb-ba33354a50c7

[3] All About Circuits. (n.d.). What is a Magnetometer? Retrieved from https://www.allaboutcircuits.com/technical-articles/what-is-a-magnetometer/

[4] Arduino Forum. (2014, May 19). Servo vs DC motors for a robot. Arduino Forum. Retrieved from https://forum.arduino.cc/t/servo-vs-dc-motors-for-a-robot/238047/2

[5] Botland Store. (n.d.). Wybór właściwego silnika - DC vs krokowy vs serwo. Botland Store. Retrieved from https://botland.store/content/230-wybor-wlasciwego-silnika-dc-vs-krokowy-vs-serwo

[6] CARUGATI, C. (2023). ANTITRUST ISSUES RAISED BY ANSWER ENGINES. Bruegel. http://www.jstor.org/stable/resrep51438

[7] ChatGPT, & Journal of International Affairs. (2022). OPENAI'S CHATGPT AND THE PROSPECT OF LIMITLESS INFORMATION: A Conversation with ChatGPT. Journal of International Affairs, 75(1), 379–386. https://www.jstor.org/stable/27203141

[8] Circuit Globe. (n.d.). Difference between Servo Motor and DC Motor. Circuit Globe. Retrieved from https://circuitglobe.com/difference-between-servo-motor-and-dc-motor.html

[9] De Spiegeleire, S., Maas, M., & Sweijs, T. (2017). WHAT IS ARTIFICIAL INTELLIGENCE? In ARTIFICIAL INTELLIGENCE AND THE FUTURE OF DEFENSE: STRATEGIC IMPLICATIONS FOR SMALL- AND MEDIUM-SIZED FORCE PROVIDERS (pp. 25–42). Hague Centre for Strategic Studies. http://www.jstor.org/stable/resrep12564.7

[10] Feller, B. (2023). Artificial Intelligence for Security Practitioners: A Conversation with ChatGPT. Daniel K. Inouye Asia-Pacific Center for Security Studies. http://www.jstor.org/stable/resrep49232

[11] GeeksforGeeks. (2023, May 2). What's difference between Microcontroller (µC) and microprocessor(µP)? GeeksforGeeks. https://www.geeksforgeeks.org/whats-difference-between-microcontoller-%C2%5c-and-microprocessor-%C2%B5p/

[12] GMW Associates. (n.d.). Magnetometers. Retrieved from https://gmw.com/magnetometers/

[13] GPK-32 tracked Inspection Robot: Superdroid robots: Inspection robots, tactical robots, Custom Robots. SuperDroid Robots | Inspection Robots, Tactical Robots, Custom Robots. (2023, October 5). https://www.superdroidrobots.com/product/gpk-32-tracked-inspection-robot/?utm_term=&utm_campaign=ET%2B%7C%2BPMax&utm_source=adwords&utm_medium=ppc&hsa_acc=2041907023&hsa_cam=20349989458&hsa_grp=&hsa_ad=&hsa_src=x&hsa_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAjw1t2pBhAFEiwA_-A-NBgkem9ZKSEdFdnSscm_kwBtEMy_5AdBYAWgNF6533FFC2kaUVqoERoCf68QAvD_BwE

[14] How GPS Works. (n.d.). Www.maptoaster.com. https://www.maptoaster.com/maptoaster-topo-nz/articles/how-gps-works/how-gps-works.html#:~:text=GPS%20uses%20a%20lot%20of

[15] K, H. P. (n.d.). Choosing batteries for Robots. Engineers Garage. https://www.engineersgarage.com/choosing-battery-for-robots/

[16] Kelechava, B. (2021, July 19). SAE Levels of Driving Automation. The ANSI Blog. https://blog.ansi.org/sae-levels-driving-automation-j-3016-2021/#gref

[17] Kiwibot autonomous delivery robots, revolutionizing the future of robotic delivery. (n.d.). https://www.kiwibot.com/

[18] Manyika, J. (2022). Getting AI Right: Introductory Notes on AI & Society. Daedalus, 151(2), 5–27. https://www.jstor.org/stable/48662023

[19] Murray, C. (2023, September 12). U.S. Data Privacy Protection Laws: A Comprehensive Guide. Forbes. https://www.forbes.com/sites/conormurray/2023/04/21/us-data-privacy-protection-laws-a-comprehensive-guide/?sh=346803825f92

[20] NOAA Office of Ocean Exploration and Research. (n.d.). Magnetometer. Retrieved from https://oceanexplorer.noaa.gov/technology/magnetometer/magnetometer.html

[21] Oitzman, M. (2021, May 18). Mobile Robot Standards. Mobile Robot Guide. https://mobilerobotguide.com/2021/05/18/mobile-robot-standards/

[22] Robocraze. (n.d.). What is the difference between servo motor vs DC motor? Robocraze. Retrieved from https://robocraze.com/blogs/post/what-is-the-difference-between-servo-motor-vs-dc-motor

[23] Robotmotor. (n.d.). The Difference Between AC Servo Motor and DC Servo Motor. Robotmotor. Retrieved from https://www.robotmotor.com/news/the-difference-between-ac-servo-motor-and-dc-servo-motor/

[24] Scharre, P., Horowitz, M. C., & Work, R. O. (2018). What is Artificial Intelligence? In ARTIFICIAL INTELLIGENCE: What Every Policymaker Needs to Know (pp. 4–9). Center for a New American Security. http://www.jstor.org/stable/resrep20447.5

[25] ScienceDirect. (n.d.). Magnetometer. In Materials Science. Retrieved from https://www.sciencedirect.com/topics/materials-science/magnetometer

[26] Shack, H. (2022, December 26). Make an autonomous “Follow me” cooler. Hackster.io. https://www.hackster.io/hackershack/make-an-autonomous-follow-me-cooler-7ca8bc

[27] Strategic Studies Institute, US Army War College. (2023). Uses and Ethics of Artificial Intelligence. In 2023 Annual Estimate of the Strategic Security Environment (pp. 48–55). Strategic Studies Institute, US Army War College. http://www.jstor.org/stable/resrep53246.18

[28] Teel, J. (2023, September 20). Microcontroller or microprocessor: Which is right for your new product?. PREDICTABLE DESIGNS. https://predictabledesigns.com/microcontroller-or-microprocessor-which-is-right-for-your-new-product/

[29] The Pi Hut. (n.d.). What's the Difference Between DC, Servo, & Stepper Motors? The Pi Hut. Retrieved from https://thepihut.com/blogs/raspberry-pi-tutorials/whats-the-difference-between-dc-servo-amp-stepper-motors

[30] US Department of Commerce, N. O. and A. A., US Department of Commerce, N. O. and A. A., & US Department of Commerce, N. O. and A. A. (n.d.). The Global Positioning System: Global Positioning Tutorial. Oceanservice.noaa.gov. https://oceanservice.noaa.gov/education/tutorial_geodesy/geo09_gps.html

[31] What is LIDAR? Learn How Lidar Works. Velodyne Lidar. (2022, June 3). https://velodynelidar.com/what-is-lidar/

[32] What is Motor Driver: Complete Guide. (2022, June 9). Robocraze. https://robocraze.com/blogs/post/what-is-motor-driver

[33] What is the history of Artificial Intelligence (AI)?. Tableau. (n.d.). https://www.tableau.com/data-insights/ai/history#:~:text=Birth%20of%20AI%3A%201950%2D1956&text=Alan%20Turing%20published%20his%20work,and%20came%20into%20popular%20use

[34] YoungWonks. (n.d.). What is a Magnetometer and How Does It Work? YoungWonks. Retrieved from https://www.youngwonks.com/blog/What-is-a-Magnetometer-and-How-Does-It-Work

[35] *Arduino - Electromagnetic Lock: Arduino tutorial*. Arduino Getting Started. (2023, October 25). https://arduinogetstarted.com/tutorials/arduino-electromagnetic-lock

[36] Shawn. (2021, June 29). *Types of distance sensors and how to select one?*. Latest Open Tech From Seeed. https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/

[37] Madsen, J. (2023a, October 6). *The different robot power supply types: Which is right for your operation?*. Bravo Electro Power Supply and Fan Experts. https://www.bravoelectro.com/blog/post/robot-power-supply-types

[38] Murray, C. (2023, September 12). *U.S. Data Privacy Protection Laws: A Comprehensive Guide*. Forbes. https://www.forbes.com/sites/conormurray/2023/04/21/us-data-privacy-protection-laws-a-comprehensive-guide/?sh=70e532df5f92

[39] *Choosing the Right Robot Battery: A ultimate guide*. MANLY. (2023, August 23). https://manlybattery.com/choosing-the-right-robot-battery-a-ultimate-guide/

[40] Matan. (2023, October 26). *Electromagnetic locks: How it works, application & advantages*. Electricity. https://www.electricity-magnetism.org/electromagnetic-locks/

[41] McMerkin, M., V, J., Felix, C., J., Paul, 42601, J. V.-, Derek, M, C., Kaan, John, Mysticovl, KonstantinosH, Fernandes, R., Mother, Y., Karina, Kvist, D., Street, O., Emiliano, Neto, P., … Mark. (2023, July 18). *Using a raspberry pi distance sensor (Ultrasonic Sensor HC-SR04)*. Tutorials for Raspberry Pi. https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/

[42] ParryTech, P., & ParryTech. (2022, November 11). *Adding 4G LTE connectivity to Raspberry Pi*. Blog. https://www.parrytech.net/blog/adding-4g-lte-connectivity-to-raspberry-pi/#:~:text=Raspberry%20Pi%20Setup%201%20Post%20succesful%20insertion%20of,LED%20will%20glow%20once%20power%20ON%20is%20successful.

[43] Minh, L. C., Lee, N., & Michael. (2023, July 18). *Build your own raspberry pi compass (HMC5883L)*. Tutorials for Raspberry Pi. https://tutorials-raspberrypi.com/build-your-own-raspberry-pi-compass-hmc5883l/

[44] Administrator. (2018, February 12). *Raspberry Pi L298n interface tutorial: Control a DC motor with l298n and Raspberry Pi*. ElectronicsHub. https://www.electronicshub.org/raspberry-pi-l298n-interface-tutorial-control-dc-motor-l298n-raspberry-pi/

[45] *Arduino - keypad: Arduino tutorial*. Arduino Getting Started. (2023, November 17). https://arduinogetstarted.com/tutorials/arduino-keypad

[46] *Arduino - Relay: Arduino tutorial*. Arduino Getting Started. (2023b, November 17). https://arduinogetstarted.com/tutorials/arduino-relay