Autonomous Package Delivery Robot

College of EECS - Capstone Project
Fall 2022 - Spring 2023

Bolivar Beleno
Joseph Brenneman Borisch
Junjie Sun
Budsakol Tiangdah

Table of Contents

1 Overview

This section will establish our project's goals and our team contribution strategy. It will discuss the summary of the Autonomous Package Delivery Robot, set standards for organizing our collaborative process, and pinpoint the specific area of technology that this project will influence. Another goal is to establish a general and specific timeline of completion deadlines for the course of this year. References and related documentation will be presented in the last section.

1.1 Executive Summary

The objective of this project is to create a robotic package delivery system operating in the context of an environment with well-developed pedestrian-tailored infrastructure, such as a college campus. The Autonomous Package Delivery Robot (APDR) will be capable of carrying packages while autonomously navigating along sidewalks and avoiding obstacles to reach its final destination. The scope of this project also contains a user interface in the form of a website that will allow individuals to initiate and receive deliveries at a specified destination.

The goal of this project is to update the current conditions of the Autonomous Package Delivery Robot (APDR). Hardware wise, the power supply module will be embedded in the controller PCB, the SBC (i.e., Single Board Computer) will be replaced and retrofitted to improve rendering and graphics processing capabilities. On the software side, our objective is to complete the autonomous implementation of the robot via ROS scripts that will interact with live imaging pre-processed on-board and sent to be processed on a main driver/host. This project will also introduce a new update to the APDR's communication protocol, the embedded system, and the level of computation it can handle. In the meantime, repair the broken front sensor bumpers and increase the power supply efficiency of the current power source to match the new computation needs. The developed product will be incredibly aware of stationary objects and dynamically moving pedestrians and vehicles, as well as provide an intuitive and reliable courier service to distributors and customers alike.

This project was inherited from a previous Oregon State University EECS Capstone group (2021-2022). This team will be working with Andrew Pehrson, project partner, and previous team member on this project. In its current state, the robot is capable of movement under manual control and waypoint creation using GPS. The technical goals for the team inheriting this project are developing an obstacle detection method and increasing the capability for autonomous outdoor travel of the APDR. The developed product will be incredibly aware of stationary objects and dynamically moving pedestrians and vehicles, as well as provide an intuitive and reliable courier service to distributors and customers alike.

1.2 Team Protocols and Standards

Table 1: Team Protocols and Standards table

Topic	Protocol	Standard
Documentation Standard	Finalized and intermediate documentation should be presented in a Microsoft Document File.	Before closing out a Trello ticket, the work done should be properly documented. Minimum of 1 sentence per 30 minutes spent working. Team members should use best judgement to make sure adequate information is written.
Coding Standard	Code will be properly commented and actively synchronized via GitHub.	All function and classes must be properly commented on and documentation present. Every functional block of code has to be described with at least one line of comment.
Hardware Design Standard	Team will employ Fusion360 for 3D models and KiCad for PCB design.	To maintain collaborative transparency, Fusion360 should be used, because of the cloud storage capabilities. We will use Fusion360 for 3D Modeling. KiCad was proposed as all team members have previous experience with this design environment. All modules should have proper schematics starting with a block diagram and working up to specific component schematics (Low Level − > High Level).
Team Meeting Standard	Weekly two-hour long Friday meeting.	Team members will arrive 5 minutes before meetings start and will bring to the discussion an update from the previous meeting's tasks. If for any reason a team member is not able to attend, immediate notification should be provided.
Communication	Discord will be the main communication method.	A single communication method was chosen to streamline the communication pipeline.
Task Management	A Trello board will be used to keep track of tasks progress and deadlines.	For all Trello submission, each member must document a brief description of their work on an hourly basis.
Interpersonal Conflict	All interpersonal conflicts will be resolved on a 1-1 private basis. If no resolution can be achieved, the project	If a situation arises where there is a team conflict, this hierarchy of actions should be followed.

	supervisor will be notified.	
On-time Deliverables and Team Collaboration	A pre-deadline will be implemented with deadlines to allow for team members to review any possible submission.	Work judged as complete will include all necessary content and formatting requirements listed on Canvas and will be nearly error-free.
Documentation Standard	Finalized documentation should be compiled in LaTeX using Overleaf. Intermediate documentation should be written in a Google document.	All documented changes must be communicated and tabulated on the Trello board.
Expenses and Purchases	All expenses will be presented and approved by the project partner.	All project expenses should be approved by the project sponsor as well as all team members prior to purchase. Once ap- proved, expenses will be tracked in the linked spreadsheet. All expenses exceeding the project budget will be asked to be financed by the project sponsor. If financing is not pro- vided, they will be evenly split amongst team members.
Project Partner Communication	The project partner will have official updates via email.	Email will be the main communication medium with occasional emails documenting project progress.

1.2.1 Communication Analysis and Contact Information

"Group 13" Discord Server is the communication interface for sharing resources, planning, scheduling meetings, voicing project concerns, ideas, and opinions outside of meetings. No alternate communication methods will be used with the concrete purpose of streamlining the group communications pipeline. Document Control will be handled via the "Group 13" shared Google Drive. Deliverables which require a group submission must be approved by all members of the team prior to being submitted to Canvas.

Table 2: Team Contact Information

Contact Information		
Name	Roles	Email
Bolivar Beleno	Leader	belenosb@oregonstate.edu
Joseph Borisch	Document Revision	borischj@oregonstate.edu
Junjie Sun	Communicator	sunjun@oregonstate.edu
Budsakol Tiangdah	Coordinator	tiangdab@oregonstate.edu

1.3 Gap Analysis

While the pandemic changed the way we live by limited physical human contact which also meant limited the logistics operated by humans. The demand for autonomous delivery robots went up profoundly during that time since it can guarantee contactless delivery. The global delivery robot market was valued at $3.53 billion in 2020 and is projected to reach $30.05 billion by 2030 [1].

While there are the autonomous delivery robots out there in the market, this project aims to enhance the opportunities available in the market by incorporating the recycled technology, using the materials that reached their end-of-life cycle which are the wheelchair frame, battery, motors, and wheels. Also, the goal is to run the robot in fully autonomous mode along with the safety functionalities which mean the robot could perform its tasks and operate in the environment independently, without any human intervention [2]. Regarding full autonomy capabilities, this project aims to develop a robot able to operate without monitoring, including when crossing intersections and avoiding obstacles. Existing implementations such as Starship's delivery robots [3] and Tesla's Autopilot [4] will be used in reference.

Autonomous delivery robots can be seen on the sidewalk on campus, which is a perfect environment for testing the robot as it is an enclosed ecosystem, and in some areas of some cities. While in the future, it could have abilities to be compatible with all the streets and run freely all over the city. Currently, there are autonomous delivery robots that are used for everyday deliveries such as food and packages, but in the future the market could expand to more complex, high value applications. Therefore, the potential customers could be from logistic service providers such as UPS and FedEx to small retail stores.

1.4 Timeline

Below is the projected timeline for the group, categorized by term.

			PHASE ONE		
			OCTOBER	NOVEMBER	DECEMBER
1	FALL TERM				
1.1	Study Robot Operating System 2	TEAM			
1.1.1	Study Previous Implementation of code	TEAM			
1.2	Problem Definition	TEAM			
1.3	Block Defintions	TEAM			
1.4	Assigning Responsibilities	TEAM			
1.5	Learn PCB Design	TEAM			
1.6	Final Block Definitions	TEAM			

Figure 1. Project Gantt Chart for Fall Term

			PHASE TWO		
			JANUARY	FEBRUARY	MARCH
2	WINTER TERM				
2.1	Test current state with ROS2	TEAM			
2.2	Integrate New Code and Test	TEAM			
2.3	Enhance Edge Detection	TEAM			
2.4	Improve Processing Power	Junjie			
2.5	PCB Bill of Materials	Junjie/ Joe			
2.6	Design New Replacement PCB	Junjie/ Joe			
2.7	Build PCB	Junjie/ Joe			
2.8	Order and Integrate PCB	Junjie/ Joe			
2.9	Construct Website for Project	TEAM			
2.1.1	Integrate Final Robot	TEAM			

Figure 2. Project Gantt Chart for Winter Term

			PHASE THREE		
			APRIL	MAY	JUN
3	SPRING TERM				
3.1	Final Fixes/ Adjustments	TEAM			
3.2	Presentation Preparation	TEAM			
3.2	Project Documentation	TEAM			

Figure 3. Project Gantt Chart for Spring Term

1.5 References

[1] S. M. Prateek Y, "Delivery Robot Market by Load Carrying Capacity," Freight Logistics, Chicago, 2021.

[2] J. Walker, "What are Autonomous Robots?," Locus Robotics, New York, 2022.

[3] D. Carrillo, "How many engineers does it take to make a robot sing?," Starship, 10 March 2022. [Online]. Available: https://medium.com/starshiptechnologies/how-many-engineers-does-it-take-to-make-a-robot-sing-b9a8e92e55f7. [Accessed 10 11 2022].

[4] Tesla, "Autopilot and Full Self-Driving Capability," Tesla, 10 11 2020. [Online]. Available: https://www.tesla.com/support/autopilot. [Accessed 10 11 2022].

1.6 Revision Table

Table 3: Section 1 Revision Table

Section 1 – Overview Revisions	
Date	Revision
11/11/2022	Bolivar Beleno: Updated Section 1.5 References
11/05/2022	Bolivar Beleno: Updated Section 1.1 Executive Summary
10/27/2022	Bolivar Beleno: Updated Section 1.2 and section structure
10/14/2022	Junjie Sun: Update Goal and Time
10/14/2022	Joseph Borisch: Gantt chart creation
10/14/2022	Budsakol Tiangdah: Gap Analysis, revisions
10/14/2022	Bolivar Beleno: Initial Document Creation

2 Impacts and Risks

2.1 Design Impact Statement

2.1.1 Public Health, Safety, and Welfare

The autonomous delivery robot's main functionality is to navigate itself from one place to another. Most of the time the robot uses the sidewalk to travel and complete deliveries. The fact that it has to share the sidewalk with pedestrians and also interact with traffic raises two main concerns. The first is the safety and convenience of the pedestrians. Another is the decision making of the robot to complete the path finding when it encounters obstacles, whether this is human, traffic, or other obstacles.

When the robot interacts with the pedestrians, we have to consider the diversity of all pedestrians as well. We have to be aware that human obstacles are not just adults, but can be children, elders, pregnant women, or any other person with a special health condition that may need assistance or special equipment to navigate along the sidewalk. For example, people who are using a wheelchair will need extra space to move along the sidewalk, or people who are vision impaired won't be able to avoid the robot if it is moving unexpectedly, which can lead to accidents and injuries. The speed that the robot moves when it is navigating across different environments (slow crowd/fast and close crowd/no crowd) is also another thing to monitor. Though the autonomous robot is small and can go at a much slower speed than other obstacles, it can still cause injury if it runs into people or is going at a different speed than the crowd which might cause the crowd to trip [1].

The robot is able to decide the best path to go and decide when it should adjust its movement according to the environment since it has to interact with the crowd and the traffic to ensure that it does not violate any law or make the city more chaotic. One case study in New York City covered allowing delivery robots to operate in their streets and sidewalks in Manhattan in November 2019. On the same day, the New York City officials promptly issued a cease-and-desist order, claiming that the delivery robots violated vehicle and traffic laws that prohibit self-driving cars and motor vehicles on sidewalks, and citing job loss and traffic congestion as their main concerns [2]. Therefore, implementation of the human aware navigation algorithms required consideration of social norms and proxemics rules to minimize the hazards of using the autonomous robot in public spaces.

2.1.2 Cultural and Social

The Autonomous Package Delivery Robot project is able to merge and blend with pedestrian society. This analysis is especially important for this project as the implementation and integration of autonomous vehicles in pedestrian spaces (i.e., parking lots, sidewalks, crosswalks, etc..) is still in the early stages of integration. This results in a culture which is learning to interact and coexist with these entities. To make a positive impact on this integration process, it was of paramount importance to evaluate possible negative impacts resulting from the implementation of this novel component in society.

Over the last decade, the development of artificial intelligence has been revitalized thanks to the renewed interest in an old technology, Convolutional Neural Networks [3]. As a result, there has been a boom in new, ingenious projects. The autonomous robot idea has been implemented more and more over the past five years. In a study published in 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI) [4] researchers outlined the frictions experienced by medical staff as they tried integrating delivery robots. Three key findings of the study were that medical units had a low tolerance for interruptions, they did not perceive the robots providing enough of a benefit to justify the cost, and the robot would break down due to high traffic within the hospital. All of this led to the delivery robots having a negative impact on workflow and caused lots of staff to resist working with this technology.

One key takeaway from this study that our team considered is that interactions with delivery robots are extremely impactful. This is because the robot is in contact with the general public more often compared to long-distance autonomous courier delivery. A 2016 report by McKinsey & Company [5] estimates that autonomous vehicles will make up 85% of last-mile deliveries by 2025. The people who share the sidewalks with these robots are not the ones actively sending or receiving the delivery, so there is a high likelihood that they will have a lower tolerance for being interrupted by these deliveries. This means that extra consideration had to be put into how the robot travels to its destination. Another key takeaway is that because the ultimate goal is complete pedestrian integration, we wanted to reach an equilibrium between clutter and low visibility for this project implementation. This means, we didn't want the robots to be perceived as clutter or an inconvenience for pedestrians, but we also didn't want them to have such low visibility that they become susceptible to accidents because a pedestrian was unaware of their presence.

2.1.3 Environmental

The materials used to construct any electronic device play a large role in its environmental impact. Electronic devices use refined metals and materials that could pollute the earth if not disposed of properly. In order to help combat this issue, this project recycles an electric wheelchair chassis and batteries to be used for the body of the robot. Batteries, specifically Lithium-ion batteries like those used in this design, are typically not disposed of properly. According to an article titled "It's Time to Get Serious About Recycling Lithium-Ion Batteries," posted to *Chemical and Engineering News*, "only 2–3% of Li-ion batteries are collected and sent offshore for recycling… The recycling rates in the European Union and the US—less than 5%—aren't much higher," [6]. Lithium-ion batteries are recyclable, there is just no good system currently in place to deal with them. These devices should be reused when possible, or properly disposed of. In order to help combat this issue, this team ensured proper disposal of all waste components of the project, especially including the lithium-ion batteries.

Using electrically powered package delivery robots could have an impact on the pollution produced in an area. Many package delivery systems currently in place, such as trucks and vans, burn fuels in order to operate. According to an article titled "What Are the Possible Solutions to Reduce the Environmental Impact of Last-Mile Deliveries," the author states

“Studies have shown that sidewalk autonomous delivery robots (SADRs) have the highest efficiency in terms of energy consumption per customer when used in small, densely populated service areas, compared to drones or autonomous delivery cars or vans. Drones’ and autonomous vans’ efficiency increases in large service areas with a lower population density,” [7]. It is important to keep in mind that this applies to relatively short, distanced deliveries in populated cities. Therefore, this kind of product is best designated to populated towns where delivery travel time will be relatively short.

Another concern regarding this project was the fact that autonomous vehicles can possibly disrupt wildlife and become invasive to their living space. Wildlife can be unpredictable and therefore it was important to program and interpret what to do in a given situation. In order to combat this, our team incorporated a navigation system that is robust enough to handle interactions with non-human obstacles in a way that will not pose harm to them.

2.1.4 Economic

Autonomous delivery vehicles are beginning to become more common in daily life. Based on Mark Fagan’s team from Harvard Kennedy School, they state, “In Houston, Texas, residents can get their groceries from Kroger delivered via an autonomous vehicle operated by the AV company Nuro, which brings your order directly to your door. Some of those groceries may have been shipped in from Dallas on an automated long-haul truck, as UPS, FedEx, and JB Hunt Transport Services are all conducting pilot projects with self-driven semis running themselves at high speeds for hundreds of miles (with a safety driver in place) between these two Lone Star hubs.” [8]. In the current world, several companies are starting to develop autonomous package delivery. As this technology develops, it can possibly fundamentally change the current transportation industry and improve people's quality of life.

The goal of this team was to capitalize on this increase in desire for autonomous vehicles. There are currently many resources being developed and studies regarding the use of autonomous vehicles. This was useful for our team to develop a design that can be useful to the current world of autonomous vehicles.

Autonomous delivery vehicles can also revolutionize the last mile deliveries. Last mile deliveries refer to the last step of the delivery process, in which the package is actually delivered to the recipient. This is typically the most labor and resource intensive part of the delivery process. Our autonomous robot furthers this path and makes some differences in the industry by attending to this issue. Our team focused on energy allocation and improving computation efficiency. By doing this, it accelerated the progress of solving the last mile delivery problem. As the Nuro Final Report points out, “Delivery AVs can provide an effective solution to carry goods from local stores and restaurants through an on-demand, last-mile delivery service. These vehicles are equipped with lidar sensors, radar, and cameras to monitor surroundings, and are designed to carry goods rather than people. Custom on-road delivery AVs, like Nuro’s R2, are space-efficient electric vehicles, which are smaller than a standard passenger car. They are built to operate on urban and suburban streets” [9]. Using the knowledge of the last mile delivery, our team designed a package delivery robot that takes this into account and implements a navigation system that is able to better tackle delivery issues.

The autonomous vehicle market is expected to increase substantially by the year 2035. According to the Nuro Final Report “Total Economic Activity worth $3.4 trillion could be potentially generated in the US economy across different sectors (retail, automobile manufacturing, warehousing and storage, etc.)” [9]. This report also states “60% ($2.0 trillion) is expected to consist of additional Value Add to GDP” [9]. From this report, we can see that the autonomous vehicle delivery market has a lot of room to expand, and companies are beginning to look for autonomous alternatives to delivery options. With this in mind, our team hoped to produce a design that can compete with current autonomous vehicles and fit into the current market.

2.2 Risks

Table 4: Risk assessment table

Risk ID	Risk description	Risk category	Risk probability	Risk impact	Performance indicator	Action Plan
R1	Vendor Delays	Timeline	30%	Low	Lead times, in-stock quantity	Clarify which components are necessary for this project
R2	Documentation	Technical	20%	Low	GitHub compilation	Sort clearly about the related code. Keep the record of revising the current schematics
R3	Hardware physically damaged	Cost/ Technical	30%	Medium	Inspection of components	3D print components are exchangeable.
R4	Lacking relative knowledge to activate the program	Education	50%	High	Lack of the constructive input to this project	Set up the routine meeting with instructors and mentors
R5	Computational Limitations	Technical	30%	High	Process runtime	The potential to upgrade the current chip into FPGA chip or computer chips such as intel, AMD
R6	Circuit design and manufacturing limitations and debugging	Technical	20%	Medium	Component malfunction	Intensive simulation and tests are needed.
R7	Procrastination	Timeline	40%	High	Due dates not being met	Delegate the specific tasks clearly to each team member. Tag each team member their responsibility clearly
R8	Communication	Technical	40%	Medium	Communication signal is unstable	Redesign the antenna and figure out how to stabilize the communication bandwidth

2.3 References

[1] A. Y. &. A. W. Troianos, "Legal Considerations Before Deplaoying Autonomous Delivery RObots," 23 February 2020. [Online]. Available: https://thespoon.tech/legal-considerations-before-delploying-autonomous-delivery-robots/.

[2] D. P.-G. &. A. B. Pericle Salvini, "Safety Concerns Emerging from Robots Navigating in Crowded Pedestrian Areas," International Journal of Social Robotics, 2022.

[3] S. P. P. S. Kumar Chellapilla, "High Performance Convolutional Neural Networks for," Tenth International Workshop on Frontiers in Handwriting Recognition,, La Baule (France), 2006.

[4] B. M. a. J. Forlizzi, "Robots in organizations: The role of workflow, social, and environmental factors in human-robot interaction," 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 287-294, 2008.

[5] B. P. &. J. Lewis, "Urban Technology Studies:," Chaddick Institute, Chicago, IL, 2019.

[6] M. Jacoby, "It’s time to get serious about recycling lithium-ion batteries," 14 July 2019. [Online]. Available: https://cen.acs.org/materials/energy-storage/time-serious-recycling-lithium/97/i28.

[7] M. Alexandrova, "Can Autonomous Robots help reduce CO2 emissions?," LMAD, Chicago, 2022.

[8] B. G. a. E. G. Mark Fagan, "Autonomous Delivery Vehicles:," Harvard Kennedy School, Massachussetts, 2022.

2.4 Revision Table

Table 5: Risk assessment table

Section 1 – Overview Revisions	
Date	Revision
11/11/2022	Bolivar Beleno: Updated Section 2.4 References
11/05/2022	Bolivar Beleno: Updated Section 2.2.3 Economic Factors
10/29/2022	Junjie Sun: Update Risk Table
10/29/2022	Joseph Borisch: Risks Section
10/29/2022	Budsakol Tiangdah: Design Impact Statement
10/29/2022	Bolivar Beleno: Project Scope and Requirements Sections
10/25/2022	Bolivar Beleno: Initial Document Creation

3 Top-Level Architecture

3.1 Block Diagram

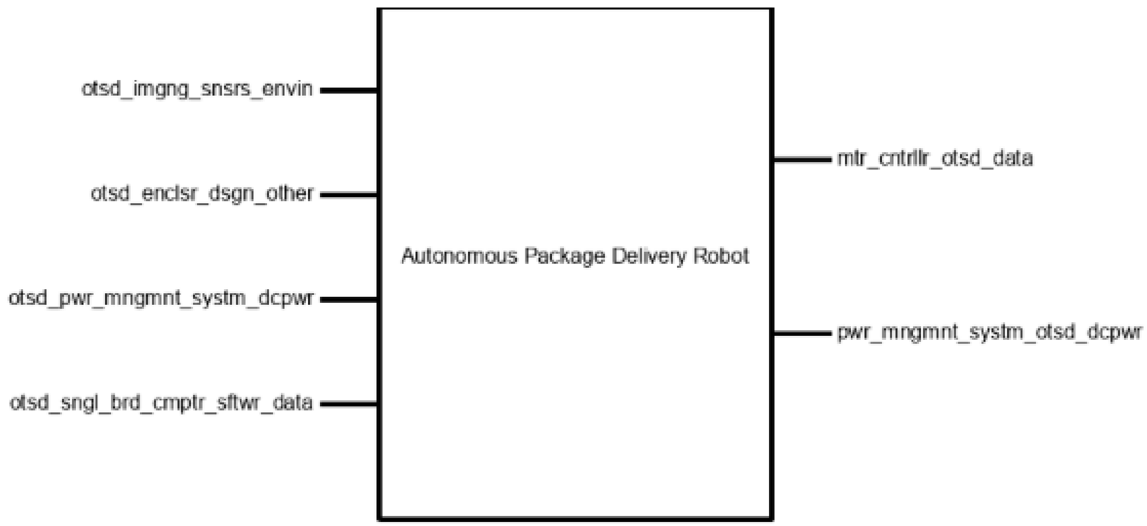

Figure 4: Black Block Diagram of Autonomous Package Delivery Robot

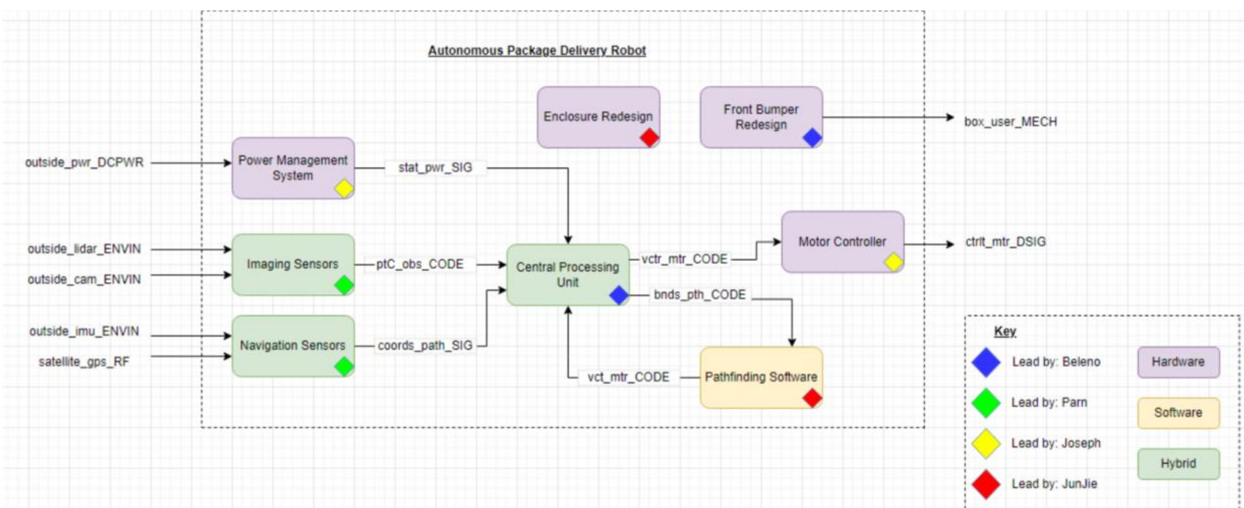

Figure 5: Top Level Block Diagram of Autonomous Package Delivery Robot

3.2 Block Description

Table 6:Block Description

Name	Description
Imaging Sensors Champion: Parn Tiangdah	The image sensor is the block that takes in the environmental data from outside of the system which tells the robot what the environment around it looks like including obstacles, edges, paths, signs, and free spaces. The robot will get the environmental data through both Lidar (Light Detection and Ranging) sensor that we currently have on board and the camera that will be added to the robot according to the recommendation from our project partner in order to incorporate image processing capabilities, reinforce object identification, improve the accuracy in edge detection, and enhance some safety features. Since the Lidar sensor was already implemented and used on board, this block will mainly focus on implementing and utilizing the camera capabilities. The camera will capture its environment as still images. Then sending those digital data to the onboard processor to perform image processing. The data from a camera will be used by other blocks including the edge detection block and pathfinding block allowing the ability to fulfill the requirement of doing edge detection, object identification and avoidance, object collision prevention, and path generation. The robot will use these data to control the motor controller block to perform desired autonomous behaviors and accomplish its tasks. Overall, this image sensor block on an autonomous robot plays a crucial role in allowing the robot to perceive and understand its environment, make decisions, and perform tasks in a more autonomous manner.
Edge detection Champion: Parn Tiangdah	This edge detection block takes in data from the imaging sensor block as RGB images at the capturing rate of 30 frames per second. It takes those images into the process of image processing and then sends the output as images (edge map) to the mapping block. The main goal of this block is to detect the edge of the pathway and obstacles. Then send the processed data to the mapping block for further usage in determining the best route.

Single Board Computer Hardware Champion: Bolivar Beleno	The Single Board Computer Hardware Block is championed by Bolivar Beleno. It is a hardware centered block that is mainly focused on the power interphase and enclosure dimensioning. Because this system component is comprised exclusively by the Nvidia Jetson Nano Single Board Computer a mechanical connection interface will be defined.
Pathfinding Software Champion: Junjie Sun	This block takes into the user choose destination and source, and import the map that build from the camera and LIDAR sensor, it figure out a optimized route and send that optimized route as an array of command to motor control board
Enclosure Design Champion: Junjie Sun	The vehicle needs a solid enclosure to protect itself from the weather. There are several key area needs to have a strong cover, for example like, the case for the computer or power supply source. Fastening: drill through the enclosure of the Power& CPU enclosure protection and bolted on the Lego surface.
Motor Controller Champion: Joe Borisch	The motor controller block is responsible for receiving directional data from the computer and controlling the signals that are sent to the motor controller board in order to ensure proper robot movement functionality. This is a software focused block that will be championed by Joseph Borisch. The code for this block is implemented via arduino IDE and will act to interpret the data received by the computer and send the correct movement data to the SmartDriveDuo. This communication will occur over UART connection. The directional movement from the computer will be interpreted as a step function, where the highs and lows are read as binary values.
Power Management System Champion: Joe Borisch	This block is responsible for handling all of the different power requirements for the robot. A redesign of this block from the previous teams was deemed necessary, as the current power management system present on the robot is inefficient and does not provide the appropriate requirements we have for the robot. This block is equipped with three different DC-DC converters. The block is designed with modularity in mind for future additions to the robot.

Single Board Computer Software Champion: Bolivar Beleno	The Single Board Computer Software Block is a software centered block that controls the movement by receiving action commands from the controller and then transmits this data over serial communications to the motor controller. As per fundamental architecture of the ROS2 software, this block will function as a node in an ROS2 topic framework and will transmit this data over a UART interface to the system's motor controller: the "SmartDrive Duo". A standard topic, "cmd_vel" will be used to pull navigation information from theROS2 stack. Then, this data will be transformed into a left and right motor speed. Finally, this data will be repackaged according to the serial packetized instructions expected by the SmartDrive Duo interface standards.

3.3 Interface Definition

The following tables will define the specific properties associated with the interfaces defined in Figure 4. These definitions will provide a specific profile of the interactions between user input and output and define the scope of intermediate steps required to fulfill those I/O specifications.

Table 7: Interface definitions

Name	Properties
otsd_imgng_snsrs_envin	• **Other:** maximum range of 2 meter • **Other:** Field of view 60°H x 49.5°V x 73°D • **Other:** minimum range of 0.4 meter
otsd_enclsr_dsgn_other	• **Other:** Dimension: 122.6mmx69.9mm. it contains Φ3.00mm holes to attach the surface • **Other:** Fasteners: Φ3.00mm holes. • **Other:** It requires PLA filament to print the components out • **Other:** Maximum allowed board size is:: 120x68mm. • **Other:** Fasteners: x4 fasteners

otsd_pwr_mngmnt_systm_dcpwr	• **Inominal:** 600 mA • **Ipeak:** 1.0 A • **Vmax:** 26 V • **Vmin:** 22 V
otsd_sngl_brd_cmptr_sftwr_data	• **Messages:** System expects string: "Hello World". • **Other:** Wifi implementation • **Protocol:** ROS2 protocol will be used to interface between server and system.
imgng_snsrs_edg_dtctn_data	• **Datarate:** Minimum of 30 frames per second • **Other:** RGB image • **Other:** .png file type
imgng_snsrs_mp__data	• **Datarate:** Minimum of 30 frames per second • **Other:** RGB image • **Other:** .png file type
edg_dtctn_mp__data	• **Datarate:** Minimum of 30 frames per second • **Other:** vector data of contour in coordinates formatting in .txt file for each frame • **Other:** grayscale contour image
sngl_brd_cmptr_hrdwr_enclsr_dsgn_mech	• **Other:** Dimension: 70mmx45mm • **Other:** Maximum allowed board size is: 120x79mm • **Other:** Attaching holes: 3.00mm holes • **Other:** x4 Fasteners

pthfndng_sftwr_mtr_cntrllr_data	**Datarate:** It must show that under that baud rate 9600, python set up with motor control board **Messages:** When the command for the motor control board is finished, the successful sent message shows on python terminal **Other:** For the motor control board, it is able to show the command signals through 4 direction LEDs turn on
mtr_cntrllr_otsd_data	**Datarate:** 9600 Baud **Messages:** Serial data (8-bits): 0bMDSSSSSS. **Protocol:** Serial Simplified
pwr_mngmnt_systm_otsd_dcpwr	**Inominal:** 1A **Ipeak:** 3A **Vmax:** 3.5V **Vmin:** 3.0V
pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr	**Inominal:** 500 mA **Ipeak:** 1.5A **Vmax:** 5.2V **Vmin:** 4.8V
sngl_brd_cmptr_sftwr_mtr_cntrllr_data	**Datarate:** 9600 baud. **Messages:** Serial data (8-bits): 0bMDSSSSSS. **Protocol:** Serial Simplified.
mp__pthfndng_sftwr_data	**Messages:** It must show the end location for the user choice **Messages:** It must show the start location for the user choice **Other:** specify obstacles location in the graph window

3.4 References and File Links

[1] “Canny edge detection,” OpenCV. [Online]. Available: https://docs.opencv.org/4.x/da/d22/
tutorial py canny.html. [Accessed: March 12, 2023].

3.5 Revision Table

Table 8: Revision table for section 4.3

Date	Author	Description
3/12/2023	Budsakol Tiangdah	Initial Draft: Outlined document and filled in each section with relevant research content.

4 Block Validation

4.1 Single Board Computer Hardware

Block Champion: Bolivar Beleno

4.1.1 Description

A single board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output interfaces and other components integrated onto the board. Some popular examples of SBCs include the Raspberry Pi and the BeagleBone or in our case a Nvidia Jetson Nano. These boards are designed to be compact, inexpensive, and low power, making them a popular choice for hobbyists, makers, and educators. They can be used for a variety of projects and applications, such as media centers, game consoles, home automation systems, and even industrial control systems. Some SBCs come with a pre-installed operating system, while others allow users to install their preferred OS, such as Linux or Windows IoT. The Raspberry Pi, for example, supports several different operating systems, including Raspbian, a Debian-based distribution specifically tailored for the Raspberry Pi. SBCs offer a cost-effective and flexible platform for a wide range of projects, and with the increasing popularity of the Internet of Things (IoT), they have become an increasingly popular choice for building connected devices.

It is a hardware centered block that focuses on the power management system and enclosure mechanical connection interface design

4.1.2 Design

The black box schematic is relatively simple. We have a single input and a single output. The power supply delivers the required current at the defined interface voltages under the region Vmin <V <Vmax. The interface design is explained in more detail in section 4.1.4.

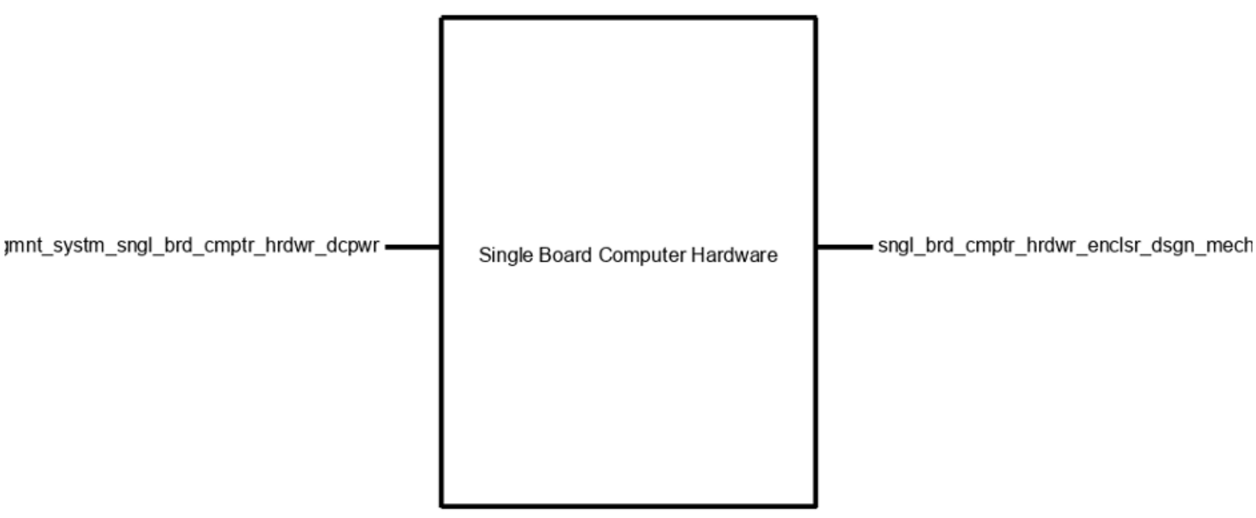

Figure 6: Black Box for Single Board Computer (Hardware) Function Block

The NVIDIA Jetson Nano 2GB Developer Kit is a powerful tool for developing and deploying AI applications. One important aspect of the kit is its input power interface. The kit features a USB-C port that is capable of accepting power from a 5V/3A USB-C power supply. The USB-C port can also be used for data transfer and connecting to a host computer. This makes the Jetson Nano 2GB Developer Kit a versatile and convenient tool for developers, as it allows them to power the device and transfer data with a single cable. It's important to note that the power supply used must be capable of providing the required voltage and current to avoid potential damage to the device. In addition to its power input interface, the Jetson Nano 2GB Developer Kit also includes a 40-pin GPIO header, a MIPI CSI camera connector, a MIPI DSI display connector, HDMI and USB 3.0 ports, and an Ethernet port for networking. These features make it a flexible and powerful platform for developing and deploying AI applications in a variety of contexts.

4.1.3 General Validation

The validation will consist of two parts, the validation of the correct power supply interface and static mechanical enclosure interface. In general, The NVIDIA Jetson Nano is a single-board computer designed for running AI and robotics applications. It has several power input options, each with specific voltage and current requirements. Here are the operating voltages and currents for the different power input options:

- Micro-USB power input: The Jetson Nano can be powered using a 5V DC power supply connected to the micro-USB port. The maximum current required for this power input option is 2A.
- GPIO header power input: The Jetson Nano can also be powered using the GPIO header, which accepts a voltage range of 5V to 19V DC. The maximum current required for this power input option is 3A.
- DC jack power input: The Jetson Nano also has a DC jack, which accepts a voltage range of 9V to 19V DC. The maximum current required for this power input option is 3A.

For our application, the USB-C power input will be implemented. It's important to note that the actual current required by the Jetson Nano will vary and depend on the specific usage scenario and the load on the device. It's recommended to use a power supply that can provide at least the minimum required current to ensure stable operation.

The architecture and code from the system passed down from the previous team is functional for the most part but due to a lack of computing capability the pathfinding functionality has never been tested. The Nvidia Jetson Nano system is proposed to overcome this serial communication bottleneck. The system is set to verify that the Nvidia Jetson Nano is capable of receiving data through a ROS2 network via ethernet. While also being able to output messages that are understandable for the motor controller (i.e., SmartDrive Duo Serial Packetized).

Subsequently, the power supply interphase will be defined which includes: Number of fasteners, diameter of fasteners, bidimensional dimensioning and on board chip dimensioning.

4.1.4 Interface Validation

sngl_brd_cmptr_hrdwr_enclsr_dsgn_mech

Table 9: Block Output

Property	Description	Reasoning
Other: Attaching holes: 3.00mm holes	PCB has holes for enclosure interface with diameter of 3.00mm	The definition of this value is important to integrate with the enclosure.
Other: x4 Fasteners	PCB has 4 fastening holes.	Number of fasteners possible is important to know the layout of the enclosure interface.
Other: Dimension: 70mmx45mm	On-board CPU chip has planar dimensions of: 70mmx45mm.	Important for enclosure interface considerations.
Other: Maximum allowed board size is: 120x79mm	The CPU chip has planar dimensions of: 100mmx79mm.	Essential for enclosure interface.

pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr

Table 10: Block Input

Property	Description	Reasoning
Inominal:	500mA	Nominal current of 500mA. This value was proposed as empirical typical use is of approximately this value.

Ipeak:	1.5A	Maximum peak current of 1.5A. We have seen current spikes of up to 1 A, for this reason a 50% buffer seemed reasonable.
Vmax:	5.2V	Maximum input voltage of 5.2V.Whilst the input can receive up to 24V we have tested and validated the use up to 5.2V.
Vmin:	4.8V	Minimum input voltage of 4.8V. Whilst the input can receive up to 2V we have tested and validated the use up to 4.8V.

Due to multiple methods to power this system (i.e., micro-USB, GPIO and DC power jack) we will use a USB-C implementation for verification purposes, but DC power jack and micro-USB have also been tested for correct interface validation

4.1.5 Verification Plan

An Nvidia Jetson Nano will be supplied power via DC power supply, with voltage ranging from Vmin < V < Vmax. Throughout this test the control feedback will be the power LED found on board.

Power Supply (Input):

1. Setup DC Power Supply to:

- V=Vmin=4.8V
- V=Vmax=5.2V

2. Power Jetson Nano via USB-C port.

3. The system is powered for both voltage values without trespassing Imax=1.5A.

4. Confirm Inominal is approximately the expected value Inominal=300mA.

Enclosure (Output):

1. Planar dimensions will be measured with a ruler.

- Board dimensions < 120x79mm
- On-board Chip dimensions < 70mmx45mm

2. Diameter dimensions will be measured with Vernier Caliper.

- Hole Diameter < 3.00mm

3. Number of holes check.

- Number of holes = 4

4.1.6 References and File Links

4.1.6.1 References

[1] “NVIDIA JETSON NANO DEVELOPER KIT,”nvidia.com. [Online]. Available: https://siliconhighway.com/wp-content/gallery/jetson-nano-devkit-datasheet-936542-US-hr.pdf. [Accessed: 15-Dec-2022].

[2] “Wiki,” *ros.org*. [Online]. Available: http://wiki.ros.org/rosserial_arduino/Tutorials/Blink. [Accessed: 22-Jan-2022].

[3] “30Amp 7V-35V SmartDrive DC Motor Driver,” *citron.io*. [Online]. Available: https://www.cytron.io/p-30amp-7v-35v-smartdrive-dc-motor-driver-2-channels. [Accessed: 15-Dec-2022].

4.1.7 Revision Table

Table 11: Revision table for section 4.1

2/01/2023	Bolivar Beleno: Modified contents with consideration of Professor Rachael's comments and feedback from fellow course mates.
1/17/2023	Bolivar Beleno: Added “Block general validation”
1/03/2023	Bolivar Beleno: Added “Block testing process”
12/15/2022	Bolivar Beleno: Added “Block overview”, “Block design”, and “References and links”
12/09/2022	Bolivar Beleno: Initial draft creation

4.2 Single Board Computer Software

Block Champion: Bolivar Beleno

4.2.1 Description

A single board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output interfaces and other components integrated onto the board. Some popular examples of SBCs include the Raspberry Pi and the BeagleBone or in our case a Nvidia Jetson Nano. These boards are designed to be compact, inexpensive, and low power, making them a popular choice for hobbyists, makers, and educators. They can be used for a variety of projects and applications, such as media centers, game consoles, home automation systems, and even industrial control systems. Some SBCs come with a pre-installed operating system, while others allow users to install their preferred OS, such as Linux or Windows IoT. The Raspberry Pi, for example, supports several different operating systems, including Raspbian, a Debian-based distribution specifically tailored for the Raspberry Pi. SBCs offer a cost-effective and flexible platform for a wide range of projects, and with the increasing popularity of the Internet of Things (IoT), they have become an increasingly popular choice for building connected devices.

It is a software centered block that controls the movement by receiving action commands from the server and then transmits this data over serial communications to the motor controller. ROS2 is a set of libraries and tools for building and operating robotic systems. It is a free, open-source software platform that provides a comprehensive framework for developing, testing, and deploying robotic applications. ROS2 is the latest version of the Robot Operating System (ROS), which is widely used in the robotics community. The main goal of ROS2 is to provide a more robust, secure, and scalable architecture for robotics applications, while maintaining compatibility with the existing ROS ecosystem. ROS2 provides a number of key features and benefits, including:

- Middleware: ROS2 includes a middleware layer that abstracts communication between nodes and provides features such as message passing, service calls, and action execution.
- Modular architecture: ROS2 is designed to be highly modular, with a number of separate packages that can be used to build robotic applications.
- Real-time performance: ROS2 is designed to support real-time applications, with low latency and high-frequency communication between nodes.
- Cross-platform support: ROS2 is supported on a number of platforms, including Windows, Linux, and macOS, making it possible to develop and deploy robotics applications on a variety of hardware.
- Interoperability: ROS2 is designed to work with other robotic frameworks and tools, such as Gazebo, MoveIt!, and OpenCV.

An Arduino UNO will be used to mimic the software aspect of output messages.

Using ROS2, this block will function as a node in an ROS2 topic framework and will transmit this data over a UART interface to the system's motor controller: the "SmartDrive Duo". A standard topic, "cmd_vel" will be used to pull navigation information from the ROS2 stack. Then, this data will be transformed into a left and right motor speed. Finally, this data will be repackaged according to the serial packetized instructions expected by the SmartDrive Duo interface standards, which can be interpreted by the motor controller to output speed.

4.2.2 Design

The black box schematic is relatively simple. We have two inputs and one output. We have the data generated by an external server which will command and direct the robot system as the input. We then have a signal going to the motor controller as an output. The interface design is explained in more detail in section 4.2.4.

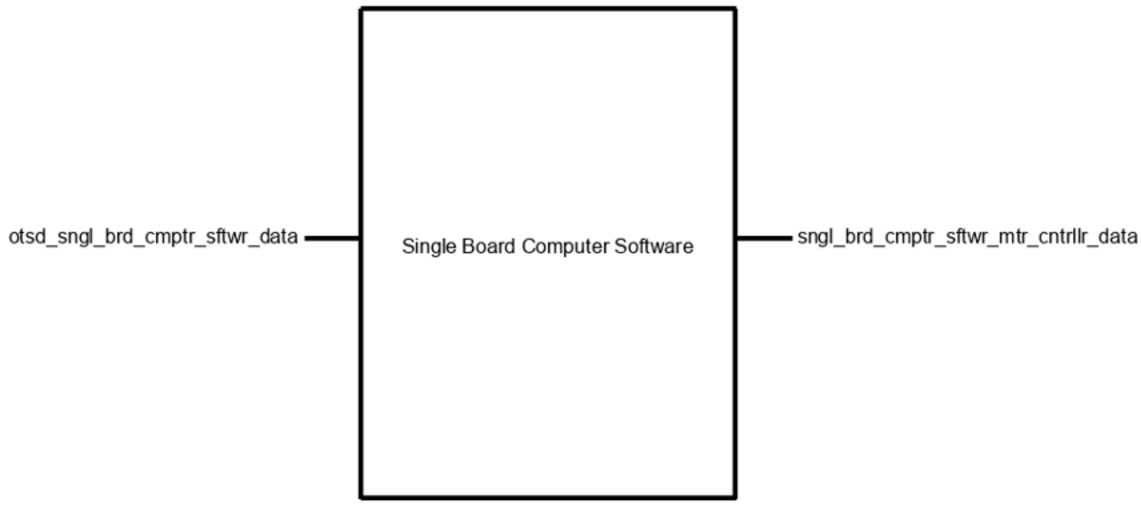

Figure 7: Black Box for Single Board Computer block

Table 12 expands on how the Nvidia Jetson Nano system interacts with the outside environment and other function blocks.

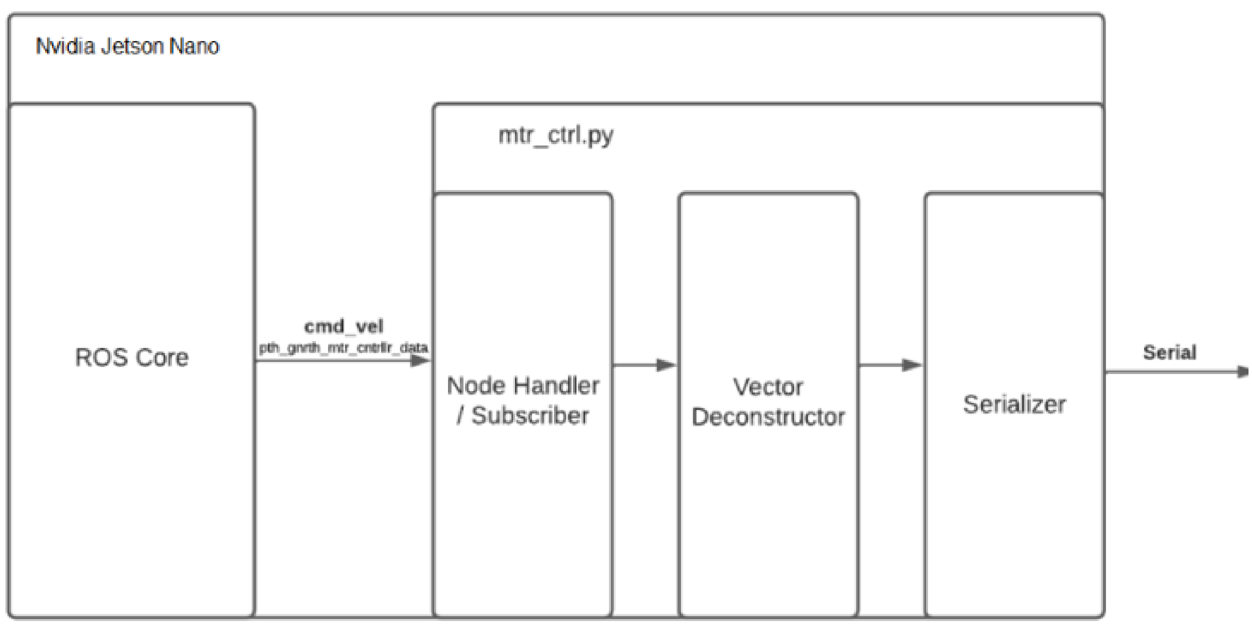

Figure 2: Flow Diagram for the Single Board Computer block

In ROS2, a topic is a named data stream that enables nodes to exchange messages using the publish-subscribe communication model. Nodes can publish data on a topic and subscribe to it to receive messages. Topics have unique names, support multiple message types, and can be used for one-to-many and many-to-one communication. The ROS2 communication infrastructure ensures delivery of messages to the correct nodes and provides features for robust communication.

The format of the cmd_vel message varies depending on the specific implementation, but it typically contains linear and angular velocity values, as well as header information. The header information can include information such as the frame of reference, timestamp, and sequence number. In our validation and verification process a string topic will be published and received by our client-server system. To simplify the process, we post via a C++ script a string containing “Hello World” with a counter tagging the message as an ID. The system will also be able to receive the same string plus integer message via a python script that hears the “myString” topic on the ROS2 network.

4.2.3 General Validation

The validation will consist of two parts, the validation of correct ROS2 network interface and serial communication for motor controller interface. In general, ROS2 topics are one of the key communication mechanisms in the Robot Operating System 2 (ROS2) framework. A topic is a named data stream that allows nodes to publish or subscribe to data. In ROS2, topics are used to exchange messages between nodes, with nodes being able to publish or subscribe to any number of topics.

ROS2 topics are unidirectional, meaning that data flows from a publisher to one or more subscribers, and they use a publish-subscribe communication pattern. Each topic has a specific message type associated with it, which defines the structure of the data being transmitted.

ROS2 topics are a crucial component of the ROS2 architecture and are used for various purposes, such as sending sensor data, sending control signals, and exchanging state information. They are an important aspect of the ROS2 framework and are used to build complex robotic systems by connecting multiple nodes and coordinating their behavior. Furthermore, validation of serial communication involves setting up the connection between two devices, such as a microcontroller and a computer, with a serial cable and ensuring that the communication settings match on both devices. Communication can be tested using a terminal program and the transmitted data can be verified for accuracy. It is important to repeat the process with different data to ensure reliability and consistency of the communication. Monitoring for errors during transmission is also necessary to guarantee correct transmission and reception of the data.

The architecture and code from the system passed down from the previous team is functional for the most part but due to a lack of computing capability the pathfinding functionality has never been tested. The Nvidia Jetson Nano system is proposed to overcome this serial communication bottleneck. The system is set to verify that the Nvidia Jetson Nano is capable of receiving data through a ROS2 network via ethernet. While also being able to output messages that are understandable for the motor controller (i.e., SmartDrive Duo Serial Packetized).

The command line interface (CLI) can be used to input commands in ROS2. ROS2 provides several command line tools that can be used to interact with the system, such as ros2 node, ros2 topic, and ros2 service. These tools allow you to control and monitor the behavior of your ROS2 system directly from the command line.

For example:

- Launching nodes: You can use the ros2 node command to launch nodes in your system.
- Listing topics: You can use the ros2 topic command to list all the topics in the system.
- Publishing messages: You can use the ros2 topic pub command to publish messages on a topic.

Listing services: You can use the ros2 service command to list all the services in the system.

4.2.4 Interface Validation

otsd_sngl_brd_cmptr_sftwr_data

Table 12: Block Input

Property	Description	Reasoning
Messages: System expects string: "Hello World".	The test subscribes to a topic. Published data is a string.	A string was chosen as the example data because it can contain both, characters and integers in a single message.
Other: Wifi implementation	The ROS2 network will be accessed by the system via WiFi.	Any network configuration is viable, as long as all users are connected to it. Ethernet and WiFi were tested but Wifi was chosen for validation as this is the logical implementation for the expected system performance.
Protocol: ROS2 protocol will be used to interface between server and system.	Communications between system and server will be handled via ROS2 protocol networking.	ROS2 protocol was chosen due to the amount of tools and accessible resources available. Furthermore, the previous iteration of this project had been implemented via this interface.

sngl_brd_cmptr_sftwr_mtr_cntrllr_data

Table 13: Block Output

Property	Description	Reasoning
Data Rate: 9600 baud.	Messages will be sent at 9600 baud or 104 us per bit.	9600 baud is a popular standard communication rate.
Messages: Serial data (8-bits): 0bMDSSSSSS.	8 bits will be output of the form: “0bMDSSSSSS.”	The structure of this message is already defined by the Serial Simplified Protocol and is expected by the motor controller.
Protocol: Serial Simplified.	Serial Simplified serial communication protocol will be used.	Serial Simplified serial communication was discussed by our team and selected to streamline communication methods between system and server. This was necessary as one of the issues of this system was the slow rate of data transfer.

4.2.5 Verification Plan

An Nvidia Jetson Nano running ROS 2 will be connected to the ROS2 network and receive a single string via the “Hello World” topic. For the output, the Jetson Nano will output a string which will confirm an appropriate ROS2 interface response.

Server Input:

1. Power on the Jetson Nano

2. Startup ROS2 on Nvidia Jetson nano and server

3.Run “demo_nodes_cpp” listener

3. Receive a series of “Hello World” string messages via the ROS2 network through command line interface (CLI).

Output:

1.Connect Arduino Uno to Jetson Nano for internal power interface

2. Check the Arduino script is running.

3. Use an oscilloscope to evaluate output serial communication messages. Proposed message is “170” due to its binary representation “0b10101010” making the detection easier for confirmation.

4.2.6 References and File Links

4.2.6.1 References

[1] “NVIDIA JETSON NANO DEVELOPER KIT,”nvidia.com. [Online]. Available: https://siliconhighway.com/wp-content/gallery/jetson-nano-devkit-datasheet-936542-US-hr.pdf. [Accessed: 15-Dec-2022].

[2] “Wiki,” *ros.org*. [Online]. Available: http://wiki.ros.org/rosserial_arduino/Tutorials/Blink. [Accessed: 22-Jan-2022].

[3] “30Amp 7V-35V SmartDrive DC Motor Driver,” *citron.io*. [Online]. Available: https://www.cytron.io/p-30amp-7v-35v-smartdrive-dc-motor-driver-2-channels. [Accessed: 15-Dec-2022].

4.2.7 Revision Table

Table 14: Revision table for section 4.2

2/01/2023	Bolivar Beleno: Modified contents with consideration of Professor Rachael's comments and feedback from fellow course mates.
1/17/2023	Bolivar Beleno: Added “Block general validation”
1/03/2023	Bolivar Beleno: Added “Block testing process”
12/15/2022	Bolivar Beleno: Added “Block overview”, “Block design”, and “References and links”
12/09/2022	Bolivar Beleno: Initial draft creation

4.3 Motor Controller

Block Champion: Joseph Borisch

4.3.1 Description

The motor controller block is responsible for receiving directional data from the computer and controlling the signals that are sent to the motor controller board in order to ensure proper robot movement functionality. This is a software focused block that will be championed by Joseph Borisch. The code for this block is implemented via arduino IDE and will act to interpret the data received by the computer and send the correct movement data to the SmartDriveDuo. This communication will occur over UART connection. The directional movement from the computer will be interpreted as a step function, where the highs and lows are read as binary values.

4.3.2 Design

As stated, the motor controller block is responsible for receiving directional data from the on board computer, and converting the data into a binary value that can be sent to the motor controller board.

The data received from the computer will take the form of a step function, where the highs and lows are read as binary 0 and 1, respectively. The motor controller block will receive this data over baud 9600 and expect to form an 8-bit binary number.

Once the data is received from the computer, it will be converted to an 8-bit binary byte and sent to the SmartDriveDuo motor controller via UART connection. The protocol used for this communication is serial simplified. This protocol allows each binary number of a byte to be interpreted as a separate command. For instance, the protocol used for this project is 0bMDSSSSSS, where M is left or right motor, D is direction, and S is speed.

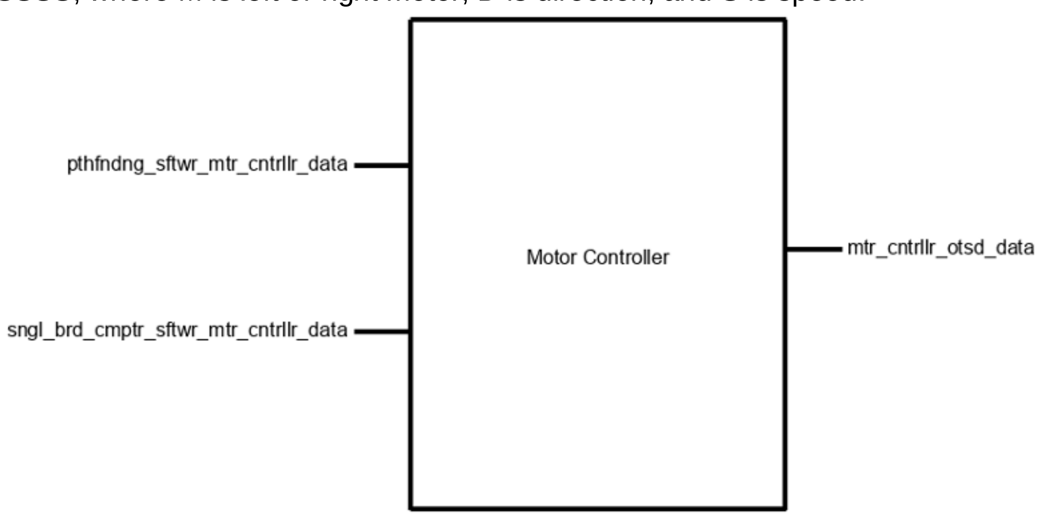

Figure 8: Black Box Diagram for Motor Controller

4.3.3 General Validation

Unfortunately, the code for the motor controller was not passed down from the previous team. However, our designs required a slightly different implementation for the motor controller, so a redesign was necessary regardless. The previous team used serial packetized for their communication protocol. This protocol is similar to serial simplified, but requires more data management and processing power. Our team decided to opt for serial packetized as it offers good motor control with simple implementation. Communication over PWM is also possible, however in the event that the robot is disconnected from the computer, the motors will be randomly driven. Serial simplified offers the best control of the motors without using too much of the processing power of the robot.

Currently, the arduino used for the motor controller block is entirely dedicated to the function of the motor controller board. This means that the on board computer does not have to use its resources to drive the motors. In addition, the arduino is a good choice as it can be easily implemented to read analog signals and send signals over UART connection.

The data received and sent by the Arduino will have a 9600 baud rate. While this isn’t the fastest communication protocol, it ensures that the data will be properly transmitted and read without skipping.

In order to reduce the bill of materials for this project, the arduino and motor controller board from the previous year is being used for this block.

4.3.4 Interface Validation

Pthfinding_sftwr_mtr_cntrllr_data : Input

Table 15: Block Input

Interface Property	Why This Value Was Chosen	Why The Design Meets or Exceeds the Property
Data Rate: It must show that under that baud rate 9600, python set up with motor control board	A baud rate of 9600 is generally reliable and can provide good data transfer rates for many applications	Arduino is capable of serial communication with baud rate of 9600 according to the datasheet

Messages: When the command for the motor control board is finished, the successful sent message shows on python terminal	Determining successful transmission of signals	Arduino capable of outputting data to monitor
Other: For the motor control board, it is able to show the command signals through 4 direction LEDs turn on	Determining successful transmission of signals	Arduino capable of controlling LED through output pins

mtr_cntrllr_otsd_data : Output

Table 16: Block output

Interface Property	Why This Value Was Chosen	Why The Design Meets or Exceeds the Property
Data Rate: 9600 Baud	A baud rate of 9600 is generally reliable and can provide good data transfer rates for many applications	Arduino is capable of serial communication with baud rate of 9600 according to the datasheet
Messages: Serial data (8-bits): 0bMDSSSSSS	Value given by datasheet for SmartDriveDuo	Arduino capable of transmitting bytes over serial communication.
Protocol: Serial simplified	Easy to implement protocol given by SmartDriveDuo datasheet	N/A

Sngl_brd_cmptr_sftwr_mtr_cntrllr_data : Input

Table 17: Block Input

Interface Property	Why This Value Was Chosen	Why The Design Meets or Exceeds the Property
Data Rate: 9600 Baud	A baud rate of 9600 is generally reliable and can provide good data transfer rates for many applications	Arduino is capable of serial communication with baud rate of 9600 according to the datasheet
Messages: Serial data (8-bits): 0bMDSSSSSS	Value given by datasheet for SmartDriveDuo	Arduino capable of transmitting bytes over serial communication.
Protocol: Serial simplified	Easy to implement protocol given by SmartDriveDuo datasheet	N/A

4.3.5 Verification Plan

A Jetson nano will be connected to the motor controller over UART serial. Motor controller code is saved on the arduino.

1. Power on Jetson Nano.
2. Run program for remote manual controls.
3. Drive the robot forward, backward, left, and right

4.3.6 References and File Links

4.3.6.1 References

[1] Abidin, Idris Zainal. “Let's Arduino Control Your Smartdriveduo-60.” *Cytron Technologies*, Cytron Technologies, 25 Jan. 2017, https://www.cytron.io.

4.3.6.2 File Links

- DeskMini Computer: https://store.minisforum.com/products/um350?variant=42279017611509

- SmartDriveDuo-60 Datasheet: https://www.technobotsonline.com/Datasheets3/1519-060-User.pdf
- Jetson Nano Developer Kit Datasheet: https://www.ximea.com/support/wiki/apis/Jetson_Nano_Benchmarks#:~:text=The%20critical%20point%20is%20that,the%20J28%20Micro%2DUSB%20connector

- Arduino Data sheet: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

4.3.7 Revision Table

Table 18: Revision table for section 4.3

Revision	Date	Author	Description
1.0	3/12/2023	Joseph Borisch	Initial Draft: Outlined document and filled in each section with relevant research content.

4.4 Power Management System

Block Champion: Joseph Borisch

4.4.1 Description

The power management block is responsible for handling the step down and regulation of the supply voltage in order to be used by the various electronic components within the robot. A 24V supply voltage comes from the acid lead batteries installed on the robot. These batteries have been recycled from an old electric wheelchair, along with the chassis. This is in an effort to reduce the impact on the environment for this project. The power management block utilizes three DC-DC converters in order to step the voltage down for use within the microcontroller and different sensors. These converters all step the voltage down to 3.3V, 5V, 12V. The decision to split the microcontroller and sensor power into different converters was made in an effort to save the power board in case of short circuits or failure. In these cases, the entire board will not fail, but only a part of it. Additionally, the power management block outputs an additional floating point value that can be digitized using an ADC and used by the microcontroller in order to find the battery level within the robot. This block is an important part of the system as a whole as it will provide necessary power to all of the various components.

4.4.2 Design

The block diagram for the power management system has several key design components. In summary, as stated previously, the power management block is responsible for powering and regulating the various sensors and microcontrollers present on the robot. In addition, this block will have an option for reading the battery life of the robot as a percentage, if we decide to incorporate this in the final design. This block is made with modularity in mind so that we could customize it in the future to fit our needs.

To begin, according to the specs of the batteries used on this robot, the batteries will output 24V. Given that the various components and sensors on the robot all require a lower voltage and will all be powered by this battery, the goal of this block is to incorporate all of the power regulation to one single PCB. If done correctly, this block should greatly simplify the wiring and power connections for this robot.

Furthermore, there are various sensors located on the robot, such as the camera, lidar, and bumpers, that need to be powered. These sensors all operate with either 3.3V or 5V. To accomplish this, the block will include a DC converter to step the voltage down to the appropriate value. The converters used in this design are buck converters, given their efficiency. The team discussed using flyback converters instead for their isolation, but the team instead opted for buck converters with over voltage protection. Each buck converter on the board is equipped with a separate circuit to protect the board from over current or over voltage.

In addition, the microcontroller present on the robot will also require a voltage step down. According to the datasheet of the DeskMini computer being used by this robot, the operating voltage should be 5V. Given this information, another DC converter will be added to the block in order to regulate this voltage. Again, a buck converter is used here for the same reasons as before.

Finally, this block will include several connection headers in order to make the board more modular. Because the robot is not fully operational in its current state, our team is not 100% certain what the final product will look like. In addition, given the fact this project was handed down to our team, we wanted to try to make a power board that would allow for easier modification for any future teams.

Below is the block diagram for the Power Management System, which shows the internal diagram for the block.

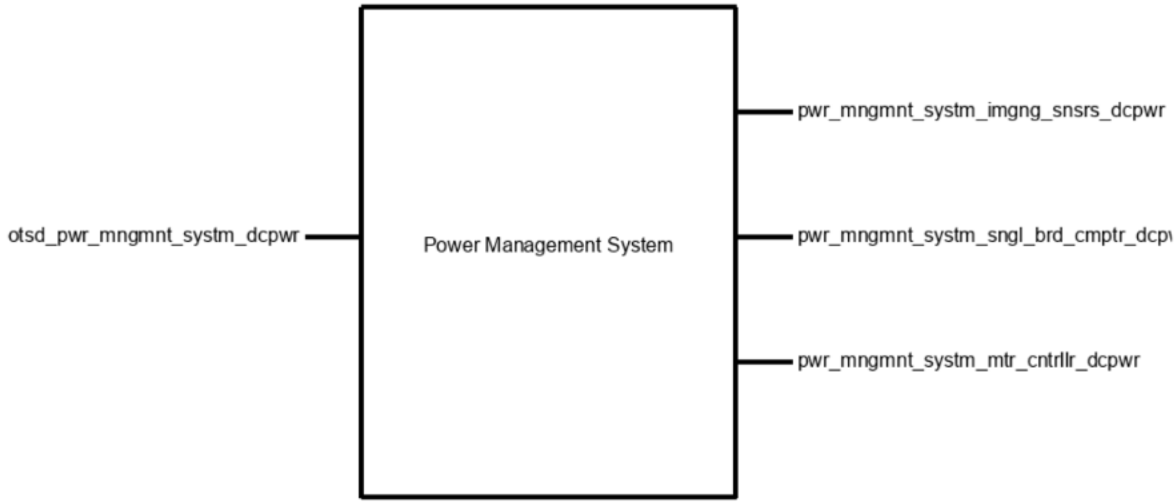

Figure 9: Block Diagram for Power Management System

Below is the schematic for the block, demonstrating how the components will be connected.

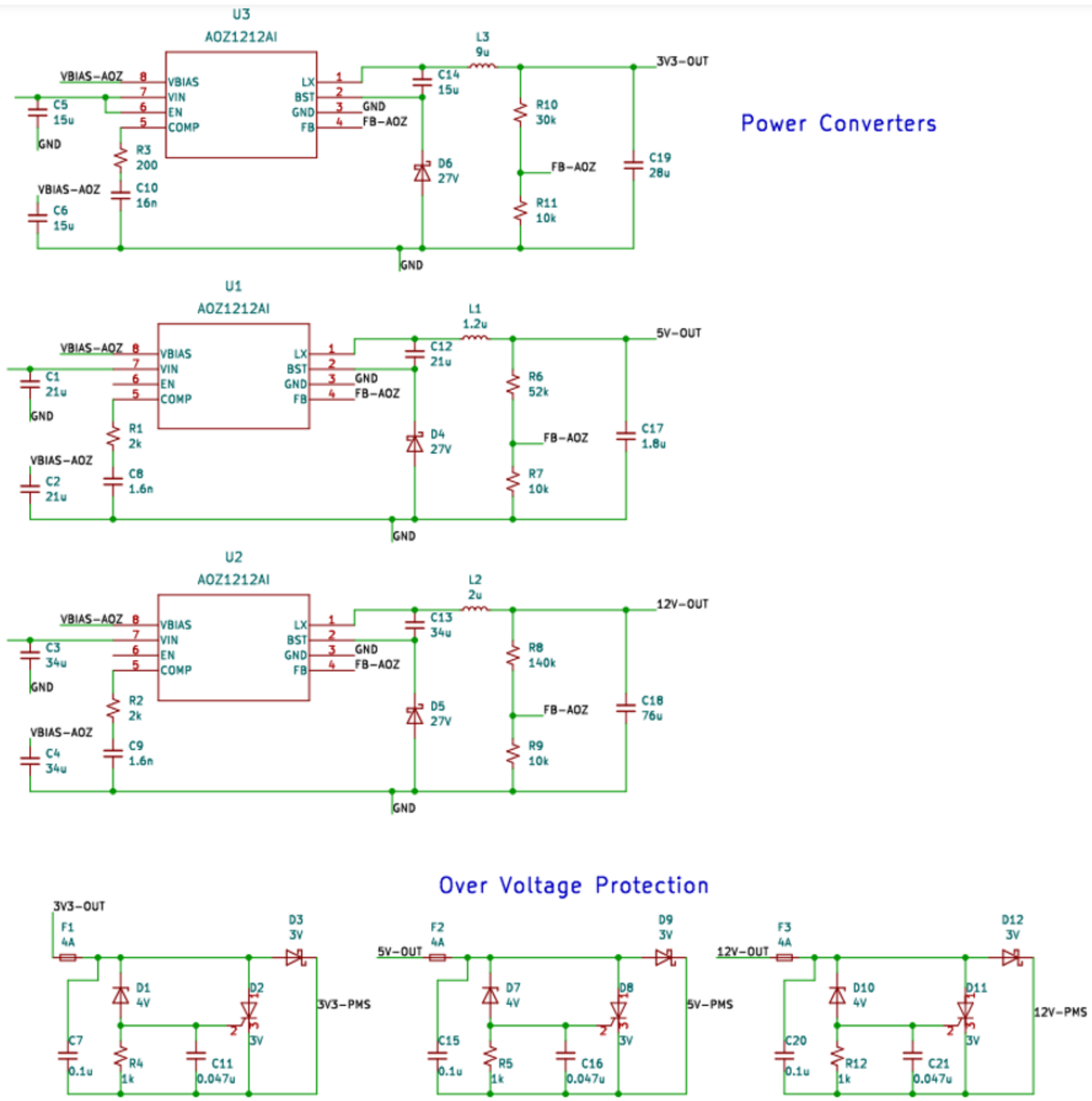

Figure 10: Schematic for Power Management System

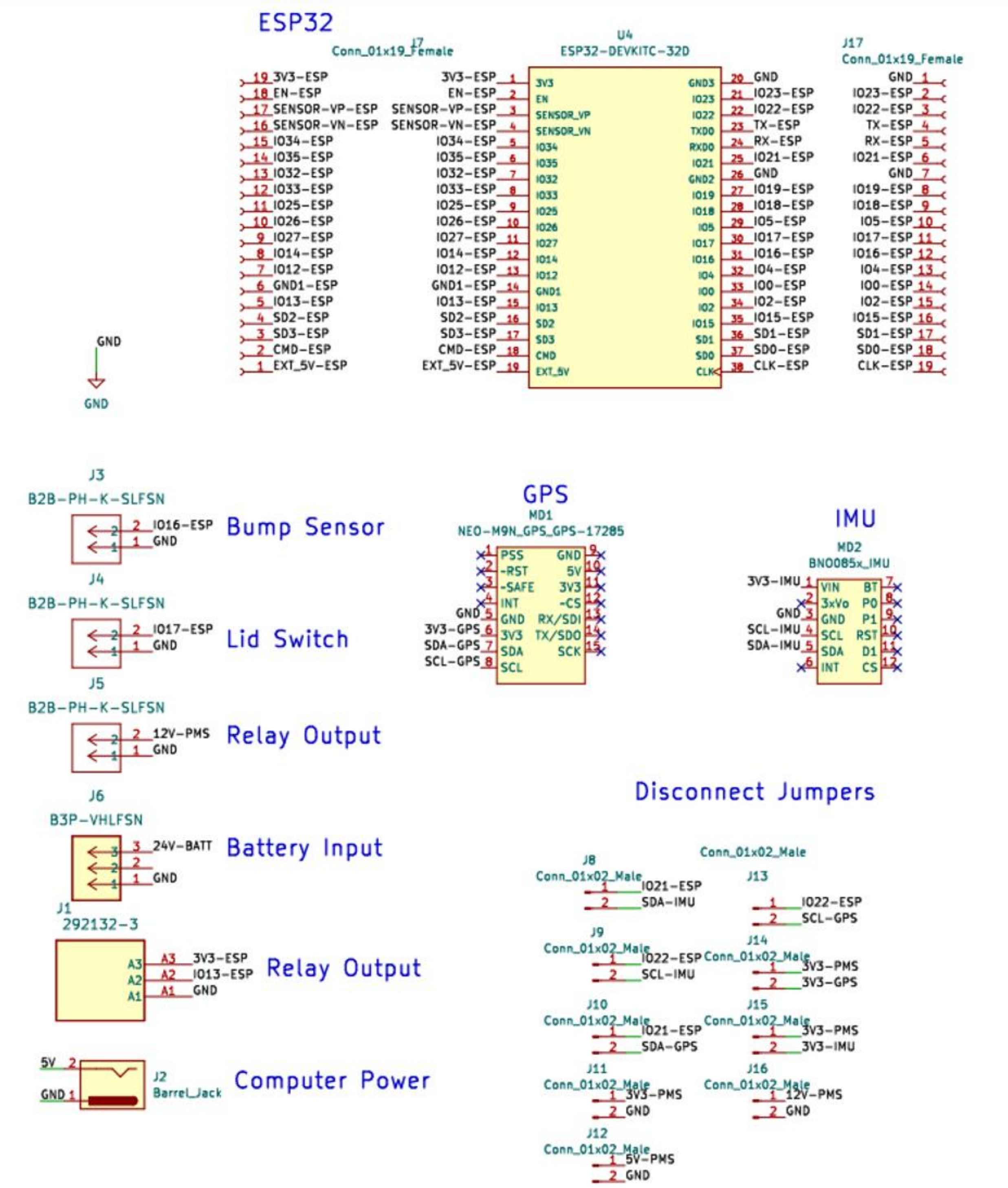

Figure 11: Schematic for Power Management System

4.4.3 General Validation

The power management block has several key roles that it must carry out for the robot to function properly.

The DC converters that regulate the voltages for the various sensors on the robot must meet the 3.3V and 5V ratings specified in the engineering requirements. This converter must regulate and step down the 24 volts from the batteries on the robot. To accomplish this, a buck converter with overprotection was chosen for the fact that it is efficient. For overprotection of this converter, a separate circuit is used with a Schottky diode. Having an isolated converter is important to this design considering the case where one of the circuits shorts, the entire PCB will not be unusable. Hopefully this will not be an issue, but previous documentation for this project noted this as a concern.

The DC converter that regulates the voltage for the computer on the robot must meet the 5V rating given by the engineering requirements. The computer used for this project is the DeskMini computer, which functions as the robots main processing system. According to the datasheet for the DeskMini, this is the voltage that is recommended to operate. As with the other DC converters, a buck converter will be used to regulate the voltage.

The final DC converter is used to power the relays present on the robot. This converter must meet the 12V rating given by the engineering requirements. As with the other converters, a buck converter will be used here.

In order to reduce the bill of materials for this project, the same buck converter module was used for all three converters. This buck converter is adjustable and handles all of the switch logic for the converter.

The power management block will require a significant part of the team’s budget. However, it has been decided that this is a vital component for the design of the robot. The current power management system on the robot is thrown together and is not efficient for what we want.

Additionally, the manufacturing and shipment of these parts will also have an environmental impact, as stated in the Design Impact Statement. However, this risk is deemed necessary for constructing the robot.

4.4.4 Interface Validation

otsd_pwr_mngmnt_systm_dcpwr: Input

Table 19: Block input

Interface Property	**Why This Value Was Chosen**	**Why The Design Meets or Exceeds the Property**
Inominal: 0.6A	The combined expected nominal power output of the 5V converters is 15W. This results in an input current of 0.6A.	The 24V lead acid battery source is capable of supplying large amounts of current to operate the wheelchair batteries. The current requirements for the other electronics is quite small in comparison.
Ipeak: 3.0A	The peak current draw from the batteries is not expected to exceed 3A	The peak load is more considerable than it is for nominal, but this peak current is only expected to occur at startup when the motors are not moving and there is no other load on the batteries.
Vmax: 26V	The nominal lead acid battery voltage is 12V. Therefore, the max value should never exceed 12.6V when fully charged. Given that this system has two batteries, this gives a maximum voltage of 25.2V. This value is rounded up to the 26V value.	The onboard lead acid batteries for the system follow the described characteristics explained in detail in [1].

Vmin: 22V	A lead acid battery with a voltage of 11V is significantly discharged and should be recharged before continued use to maximize its lifespan. This is the minimum battery voltage the system is therefore expected to operate at before recharge.	The onboard lead acid batteries follow the typical characteristics described and will need charging at 22V.

pwr_mngmnt_systm_otsd_dcpwr: Sensor Voltage output

Table 20:Block output

Interface Property	Why This Value Was Chosen	Why The Design Meets or Exceeds the Property
Inominal: 1A	The sum of the nominal current draw of all the 3.3V sensors and electronics is expected to be below 1A.	The associated buck converter is capable of an output current of 3A.
Ipeak: 3A	The sum of all the peak current draws of all the 3.3V sensors and electronics is not expected to exceed 3A.	The associated buck converter's nominal current can handle the load peak.
Vmax: 3.5V	The supply voltage for the sensors cannot exceed 3.3V; 3.5V was chosen to provide safety margin.	Output for the buck converter will not exceed 3.5V.
Vmin: 3V	The supply voltage for the sensors cannot drop below 3V; 3V was chosen to provide safety margin.	Output for the buck converter will not drop below 3V.

pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr : Jetson Nano

Table 21: Block output

Interface Property	Why This Value Was Chosen	Why The Design Meets or Exceeds the Property
Inominal: 0.5A	Estimated nominal current value.	The associated buck converter is capable of an output current of 3.0A.
Ipeak: 1.5 A	Recommended peak value given by Jetson Nano datasheet when operating in default (10W) mode	The associated buck converter's nominal current can handle the load peak.
Vmax: 5.2V	Conservative estimate of the max power draw for the Jetson Nano Developer kit	Output for the associated buck converter will not exceed 5.2V
Info: Max power consumption	Jetson Nano has two operation modes: default and low power. Default mode offers 10W max power while low power mode offers 5W. Our team has decided that the 10W default mode should accomplish what we need to do	Information only
Info: Max current	The Jetson Nano datasheet specifies that under normal operation the Jetson nano should not exceed 2A. However, it also specifies how to configure the board alternatively to accept 5V at 4A by using the external power connector on the board.	Information only
Vmin: 4.8V	Minimum voltage requirement given by the Jetson Nano datasheet.	Output for the associated buck converter will not drop below 4.8V.

4.4.5 Verification Plan

Power Source Voltage Test Procedure

1. Use a lab bench power supply to provide input otsd_pwr_mngmnt_systm_dcpwr with 22V.
2. Use a voltmeter to verify that pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr is within 4.8-5.2V.
3. Use a voltmeter to verify that pwr_mngmnt_systm_otsd_dcpwr is within 3-3.5V.
4. Repeat steps 1 through 5 with input otsd_pwr_mngmnt_systm_dcpwr set to the maximum 26V.

Power Source Current Test Procedure

1. Connect a 1A load to pwr_mngmnt_systm_otsd_dcpwr .
2. Connect a 1A load to pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr .
3. Use a lab bench power supply to prove input otsd_pwr_mngmnt_systm_dcpwr with 22V.
4. Leave the system running for at least 10 seconds.
5. Verify with an amp-meter that for the various outputs are still outputting their specified current ratings.
6. Use the lab bench power supply's amp-meter to verify that otsd_pwr_mngmnt_systm_dcpwr does not exceed 1.0A.
7. Turn off the power supply.
8. Connect a 3A load to pwr_mngmnt_systm_otsd_dcpwr .
9. Connect a 3A load to pwr_mngmnt_systm_sngl_brd_cmptr_hrdwr_dcpwr .
10. Turn on the power supply. Verify the specified peak loads are being fulfilled with an amp-meter.
11. Use the lab bench power supply's amp-meter to verify that otsd_pwr_mngmnt_systm_dcpwr does not exceed 1.0A.
12. Repeat steps 1 through 9 with otsd_pwr_mngmnt_systm_dcpwr set to 26V.

4.4.6 References and File Links

4.4.6.1 References

[1] R. Perez, "Batteries lead-acid battery state of charge vs. voltage," 1993. [Online]. Available: https://www.scubaengineer.com/documents/lead_acid_battery_charging_graphs.pdf. [Accessed: 16- Jan-2023].

4.4.6.2 File Links

- DeskMini Computer: https://store.minisforum.com/products/um350?variant=42279017611509

- Buck Converter: https://www.digikey.com/en/products/detail/alpha-omega-semiconductor-inc/AOZ1212AI/1855931
- Jetson Nano Datasheet: https://www.ximea.com/support/wiki/apis/Jetson_Nano_Benchmarks#:~:text=The%20critical%20point%20is%20that,the%20J28%20Micro%2DUSB%20connecto r.

4.4.7 Revision Table

Table 22: Revision table for section 4.4

Revision	Date	Author	Description
2.0	2/11/2023	Joseph Borisch	Final Draft: Updated document to include schematic. Updated values for interface definitions. Updated general information on document.
1.0	1/18/2023	Joseph Borisch	Initial Draft: Outlined document and filled in each section with relevant research content.

4.5 Enclosure Design

Block Champion: Junjie Sun

4.5.1 Description

In this validation, we will focus on designing an enclosure for the Nvidia Jetson Nano and power management board. The enclosure will be 3D printed, and it will provide protection to both devices while keeping them organized and easily accessible. Additionally, we will incorporate cord protection to ensure that the power cord is secure and protected from damage.

4.5.2 Design

The design process for the enclosure involved the following steps:

1. Initial Concept - We started by creating a rough concept of the enclosure design, considering the dimensions of both devices, and how they could be placed together to make a compact enclosure.
2. Sketching and CAD modeling - Using the concept, we created sketches and CAD models to visualize the design and ensure that the dimensions are accurate.
3. 3D Printing - We printed out the enclosure design using a 3D printer and made necessary adjustments.
4. Cord protection design - To incorporate cord protection, we added cable tie slots to the enclosure, which will ensure that the power cord is tightly secured and cannot be accidentally unplugged.

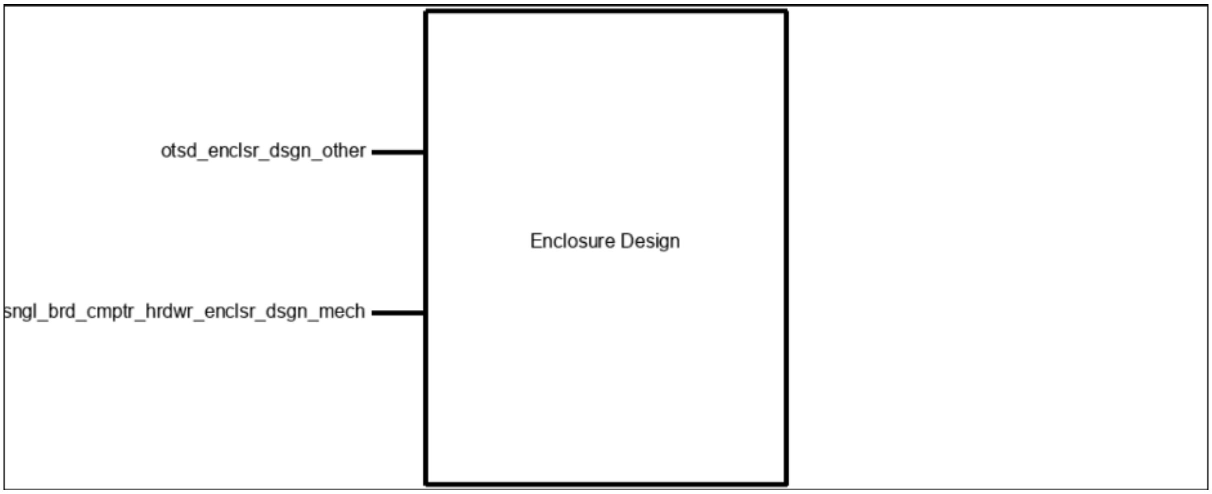

Figure 12: black block diagram

4.5.3 General Validation

To validate the design, we checked the following aspects:

1. Compatibility - We checked that the enclosure design is compatible with both devices, ensuring that they fit securely and accurately.
2. Access - We ensured that the design allows easy access to ports and buttons on both devices.
3. Cord protection - We verified that the cable tie slots provide adequate cord protection and prevent any accidental unplugging.

4.5.4 Interface Validation

otsd_enclsr_dsgn_other

sngl_brd_cmptr_hrdwr_enclsr_dsgn_mech

Table 23:Block Inputs

Other Input: Enclosure	Check if the enclosure fits Nvidia Jetson Nano and Power management board The enclosure must be able to house both Nvidia Jetson Nano and Power management board without any fitting issue	Validation Check

Temperature tolerance	Temperature Test The enclosure must maintain the temperature of Nvidia Jetson Nano and Power management board within the operating range	Validation Check
Vibration Test	enclosure must be able to protect Nvidia Jetson Nano and Power management board from external vibrations during operation.	Validation Check

4.5.5 Verification Plan

Fastening Process: The fastening process involved the following steps:

1. Determine the location - We determined the location where the enclosure protection needs to be fastened onto the Lego surface.
2. Drilling - We drilled through the enclosure using a drill bit that matched the size of the bolts we were using. We ensured that the drill went through both the enclosure and the Lego surface.
3. Bolting - We then inserted the bolts through the holes and secured them tightly with nuts.
4. Tightening - Finally, we tightened the bolts to ensure that the enclosure protection is firmly attached to the Lego surface.

Verification Plan: To verify the fastening process, we conducted the following tests:

1. Stability test - We tested the stability of the enclosure protection after fastening it onto the Lego surface to ensure that it was firmly attached and did not move or wobble.
2. Durability test - We tested the durability of the enclosure protection by applying pressure and stress to the fastening points to ensure that they could withstand the elements and hold up over time.
3. Compatibility test - We tested the compatibility of the enclosure protection with the Lego surface by ensuring that it did not interfere with the operation of other Lego components.

4.5.6 References and File Links

[1] IEEE Standard for Learning Technology--Learning Technology Systems Architecture (LTSA), IEEE Std 1484.1-2017.

[2] Abdi, J. et al, "Design for Additive Manufacturing: Trends, Opportunities, Considerations, and Constraints".

[3] Lydia Sloan Cline, "Fusion 360 for Makers: Design Your Own Digital Models for 3D Printing and CNC Fabrication".

4.5.7 Revision Table

Table 24: Revision table for section 4.5

Revision	**Date**	**Author**	**Description**
1.0	3/12/2023	Junjie Sun	Initial Draft: Outlined document and filled in each section with relevant research content.

4.6 Pathfinding Software

Block Champion: Junjie Sun

4.6.1 Description

This block is about the path-finding block that makes our vehicle move to a certain location. It contains several sub-blocks such as the navigation block, receiving and transferring data from the camera block, and the picture-handling block.

It allows the robot to make the decision in real time and predict certain events happening and avoid obstacles in advance. Overall, it enables the APDR (Autonomous Package Delivery Robot) to optimize the best routines to arrive in certain locations and it gives a feedback system that allows the robot to learn as training of this algorithm. The Pathfinding Block Validation block is the key component that enables autonomous driving ability. The design validation involves pre-processing, memory management, dynamic environment, machine learning, and optimization. The code development is expected to take place from week 1 to week 8 of the winter term, with two weeks left for improvements and adjustments. Alternatives include reducing the complexity of the algorithm to a simplified version with only an A* algorithm or a reduced version with limited details. The code should work alone for simulations and perform well when connected with the rest of the group blocks.

4.6.2 Design

There are several key components to how to design a pathfinding algorithm.

In summary: it contains obstacle avoidance, picture detection mechanism, Real-time planning, and replanning. It contains an optimization mechanism that ensures the upgrading of the efficiency of pathfinding. It needs to integrate with the vehicle to make sure the CPU of the vehicle understands the instructions of this algorithm. Overall, the design should make the vehicle to have the ability to find the best routine and make an intelligent decision about where to go and how to arrive at that destination.

Design Thinking:

1. Optimization: it has the corresponding algorithm to ensure each time, it can improve its feedback and make the calculation faster than the previous training. There are multiple routines to tackle the problem: (1). We can use Euclidean distance; it has the calculation function to speed up the pathfinding process. (2). Eliminating branches of the search tree, which through select the worst routines and eliminate them, then leave the best solutions for the vehicle. (3). Using multi-threading techniques to enhance the search process even faster, because of multi-cores in modern processors. But due to monetary issues, our project is lacking this functionality. We are thinking of replacing the current raspberry pi with an intel multi-core CPU, but the engineering of the power management is just too complex.
2. Machine learning: There are several machining algorithms that this project can benefit from. Like Q-learning, neural networks, decision trees, random forests, gradient boosting, and support vector machines. But those algorithm needs to make the corresponding changes based on our project. We need to try different algorithms to be better fitting for our projects. I’m inclined to use neural networks which can be a better fit for our camera sensors. I’m doing some research on Feedforward Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, and Generative Adversarial Networks.
3. The nature of the dynamic environment. For my block, I need a part of the code in charge of the dynamic environment and to understand the traffic, pedestrians, and other vehicles to give the pathfinding algorithm feedback in real-time and make the corresponding adjustments. Using the heuristic method, which lowers the accuracy of pathfinding but it is good enough to give a relatively good estimation of the environment. And it is a good balance between the cost of the path and the estimated cost of the goal.

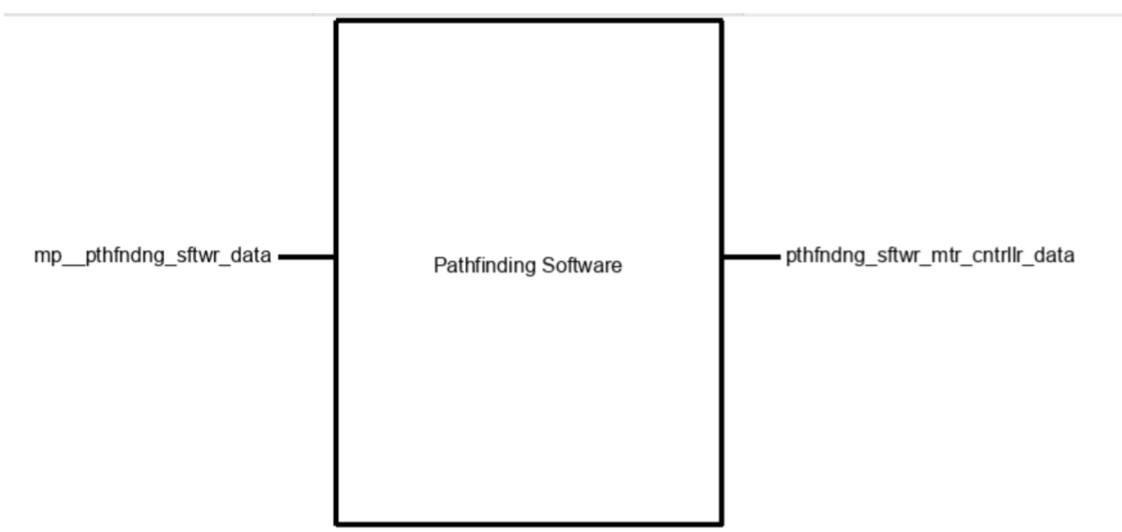

Figure 13: black block diagram

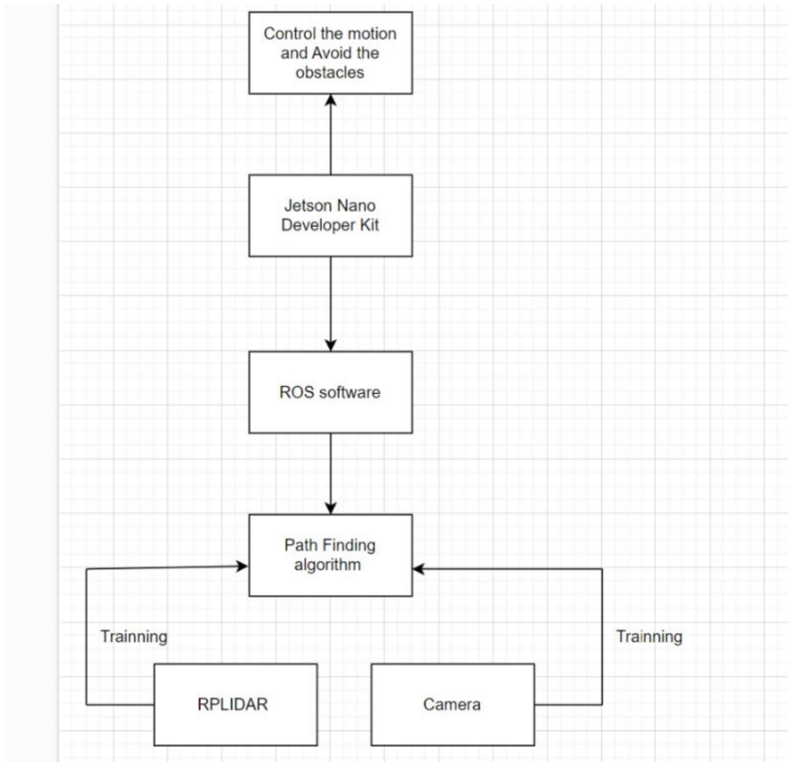

Figure 14: Overall Architecture

The Main Block Diagram for Pathfinding Algorithm:

- Starting location and destination location for the package
- Use a camera sensor to map the environment and identify any obstacles that the robot has to navigate around.
- Design specific pathfinding algorithms such as Dijkstra's algorithm or A* algorithm to find the shortest path from the starting location to the destination.
- Use the path to decide which route that robot will go to the destination. There are additional sensors or algorithms to decide on other environmental elements.
- Once the robot arrives at the destination, it opens the lids for the human to pick up the package. And repeat the overall process for the next round of the command.
  Block Diagram as the following:

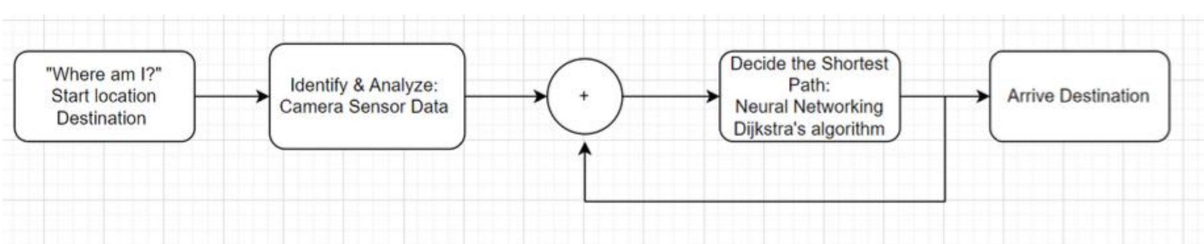

Figure 15: Block Diagram for Pathfinding Algorithm

Design Process:

There are multiple inner blocks and outer blocks that need to fulfill in order to test the functionality of the pathfinding algorithm. The most unexpected environmental problem is that the Nvidia Jetson TX2 seems does not support pygame (which I used to visualize the optimized path result from the algorithm). The solution is to use mini-Linux desktop with Ryzen5 processor to handle the pygame, ROS2 system and output the result to control the motor and receive the data from Astra S 3D Camera. The reason to use a mini desktop is that our group mentor Andrew think that we need a powerful chips to handle the computation and it is a good idea to set up a Linux desktop machine. But there is another problem with replacing the old raspberry pi with this Linux machine. It does not have the GIPO pins to connect with motor control chip. For this part, I need to work with Joe, who is in charge of motor control block to figure an acceptable interface.

Due to the time limitation, for my pathfinding algorithm, my block is focus on pin down the source and destination, and the algorithm runs and then send out the best routines command through a Arduino Uno to turn on four LEDs to indicate the signal is successfully sent and receive by Arduino board (This part is to simulate the motor control board)

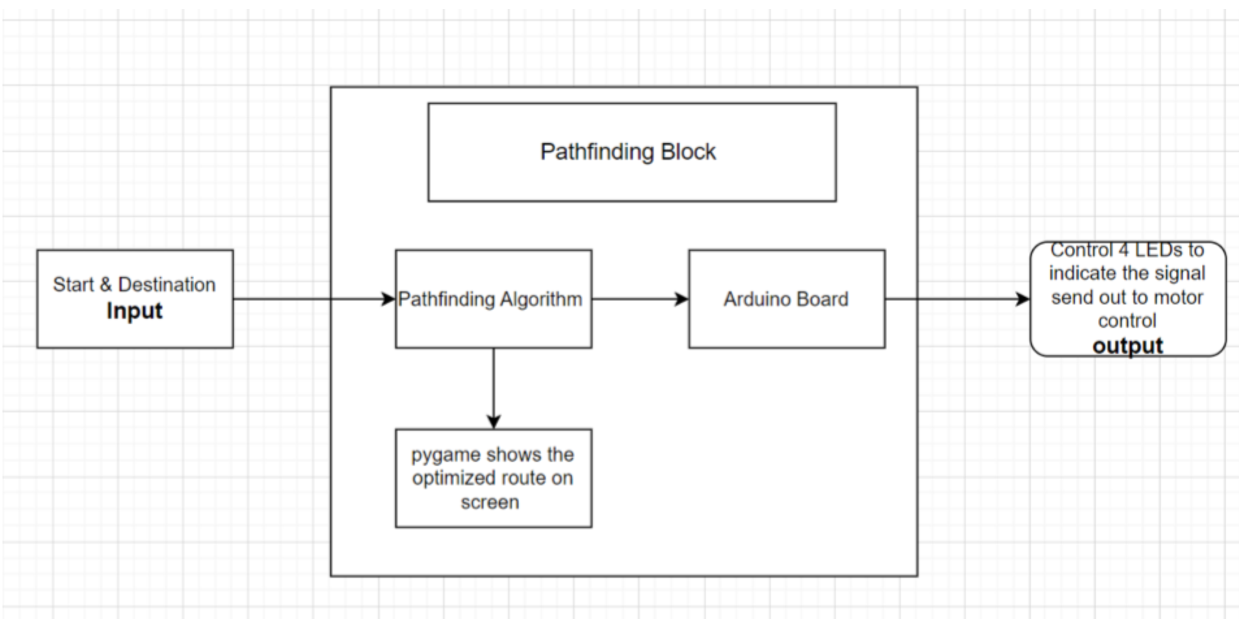

Figure 16: Design Validation

4.6.3 General Validation

My design details include the pre-processing, memory management, dynamic environment, machine learning, Optimization. The mainly cost this program is the time that I spending on researching on how to write each of code blocks. For example, how to write a simple machine learning code to implement that part of code. There is no financial cost directly related to this part. As for the understanding each of the sub-blocks that are comprised of the overall block, which is very important to implement the corresponding codes. For this block, we have to take into the account for engineering time. Because this block is relatively independent for the rest of the code blocks. It can extend for the whole period of the building the autonomous vehicle. As for building the code, it should last for the mostly winter term. From week1 to week8 mainly. The rest of the two weeks includes the improvements and adjustments of the code to the CPU portion which Boliva is in charge of and once that is done. The code has to be adjusted for sensing camera portion to test the real data into the code. Before that, we can use simulation to test the pathfinding code to match the group expectation.

One Alternate Solution:

It depends on the time schedule, it could reduce the whole algorithm into a simplified version, which only contains the neural networking portion or the corresponding mechanism. For this version, it has the basic capacity of sensing the environments and take as the feedback to adjust its decision actions to achieve a relatively calculation result.

Second Alternate Solution:

For the second version, it is just one level lower than the current block plan, which contains all of the blocks but with limited detail code for start location and destination and limit capacity to take the amount of the data the camera takes into. (This means the memory management part of this portion could be reduced or simplified from the original plan.)

Overall, general validation should work alone for simulations. Once this block is connected with the rest of the group blocks. It should perform the responsibility of receiving the camera sensors and give acute feedback to assist the main pathfinding algorithm to give the instructions to achieve the shortest path outcome.

4.6.4 Interface Validation

Interface Explanation:

The interface validation for the "imgng_snsrs_pthfndng_sftwr_envin" and "pthfndng_sftwr_mtr_cntrllr_data" must meet specific properties to ensure their proper functioning. The "imgng_snsrs_pthfndng_sftwr_envin" interface must demonstrate the presence of obstacles and available routes on the window, show the start and destination for the user choice in the terminal, and be able to find the best routes within 5 minutes. The "pthfndng_sftwr_mtr_cntrllr_data" interface must have a data rate that shows that under a baud rate of 9600, python is set up with the motor control board, show successfully sent messages when the command for the motor control board is finished, and be able to display the command signals through the 4 direction LEDs turning on.

imgng_snsrs_pthfndng_sftwr_envin: Input

Table 25: Data Input

Sensor Data Input	Data Output It needs to provide the information of the shortest path to the robot. It makes the robot able to reach its destination by following the path. Use the shortest path calculated by the algorithm as output for the robot the destination that it wants to go to deliver the package.	Implement Python and window display
Other: info: Window show the routes	Display on window It must display the shortest path on the window for the user to monitor the progress of the robot Implementation	Validation Check
Other: info: Show the outcome of the command that send to robot	Display on LEDs It must also display the information of the shortest path on the LEDs for the robot to follow Implementation	Validation Check

imgng_snsrs_pthfndng_sftwr_envin

Table 26:Algorithm Output

Identify & analyze the Datarate	It must show that under that baud rate 9600, python set up with motor control board	Validation Check
Transfer message show on terminal	When the command for the motor control board is finished, the successful sent message shows on python terminal	Validation Check
Four the signal LEDs show the correct order of the communication array	For the motor control board, it is able to show the command signals through 4 direction LEDs turn on Implementation	Validation Check

4.6.5 Verification Plan

Specified Output Interface:
It informs the robot where it is and give the destination coordination. It prepares for the algorithm to decide which one is the shortest path.

1. Setup a window by demonstrating that which one is the optimized route.
2. Successfully print out the start and destination coordinate on the terminal.
3. Successfully print the command that the pathfinding algorithm gives to the motor control board.
4. Successfully setup the serial communication with Arduino control broad and sending data and receiving data.
5. Arduino motor control board is sending the correct signal to the four motors to respond with the optimized route.

4.6.6 References and File Links

4.6.6.1 References (IEEE)

[1] P. D. Wasserman and T. Schwartz, "Neural networks. II. What are they and why is everybody so interested in them now?" in IEEE Expert, vol. 3, no. 1, pp. 10-15, Spring 1988, doi: 10.1109/64.2091.

[2] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," in IEEE Transactions on Neural Networks, vol. 2, no. 2, pp. 252-262, March 1991, doi: 10.1109/72.80336.

[3] R. Lippmann, "An introduction to computing with neural nets," in IEEE ASSP Magazine, vol. 4, no. 2, pp. 4-22, Apr. 1987, doi: 10.1109/MASSP.1987.1165576.

[4] A. M. Luthfi, N. Karna and R. Mayasari, "Google Maps API Implementation On IOT Platform For Tracking an Object Using GPS," 2019 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bali, Indonesia, 2019, pp. 126-131, doi: 10.1109/APWiMob48441.2019.8964139.

[5] Y. Sazaki, A. Primanita and M. Syahroyni, "Pathfinding car racing game using dynamic pathfinding algorithm and algorithm A*," 2017 3rd International Conference on Wireless and Telematics (ICWT), Palembang, Indonesia, 2017, pp. 164-169, doi: 10.1109/ICWT.2017.8284160.

4.6.6.2 File Links

https://github.com/Thyix/astar-pathfinding

https://github.com/ShaheerSajid/OpenCV-Maze-Solving

https://github.com/sxaxmz/Advanced-Vision-Motion-and-Image-Analysis-

4.6.7 Revision Table

Table 27: Revision Table for section 4.6

1.0	1/14/2023	Junjie Sun	Initial draft: Setup the framework for this document and allocate each section with proper research content.
1.1	2/5/2023	Junjie Sun	Second Draft: Add Design challenge part Update verification plan
1.2	2/6/2023	Rachael Cate	Third Draft: Fix the indentation and numbering issues for the reference and the font& naming problem

4.7 Edge detection

Block Champion: Budsakol Tiangdah

4.7.1 Description

This edge detection block takes in data from the imaging sensor block as grayscale images at the capturing rate of 30 frames per second. It takes those images into the process of image processing then sends output as images (edge map) to the mapping block. The main goal of this block is to detect the edge of the pathway and obstacles. Then send the processed data to the mapping block for further usage in determining the best route.

Overall, this edge detection block plays a crucial role in allowing the robot to navigate and interact with its environment. This edge detection block includes the process of identifying and locating the boundaries between different objects or surfaces within an image allowing the robot to get the data about the distance away from the edges of the pathway, obstacles, or even upcoming traffic.

4.7.2 Design

The objective of this block is to process the data received from the imaging sensor block and provide the data output about the robot environment including the edges of the pathway, obstacles, and upcoming traffic to the mapping block as contour images and vector data set of the contours on each image at the same rate as the input data rate which is 30 frames per second, same rate as video streaming that will give the robot real time environmental data. It will save each contour mapping image as a grayscale .PNG file, and save the contouring data as coordinate as text file for each image, total of 30 files per second.

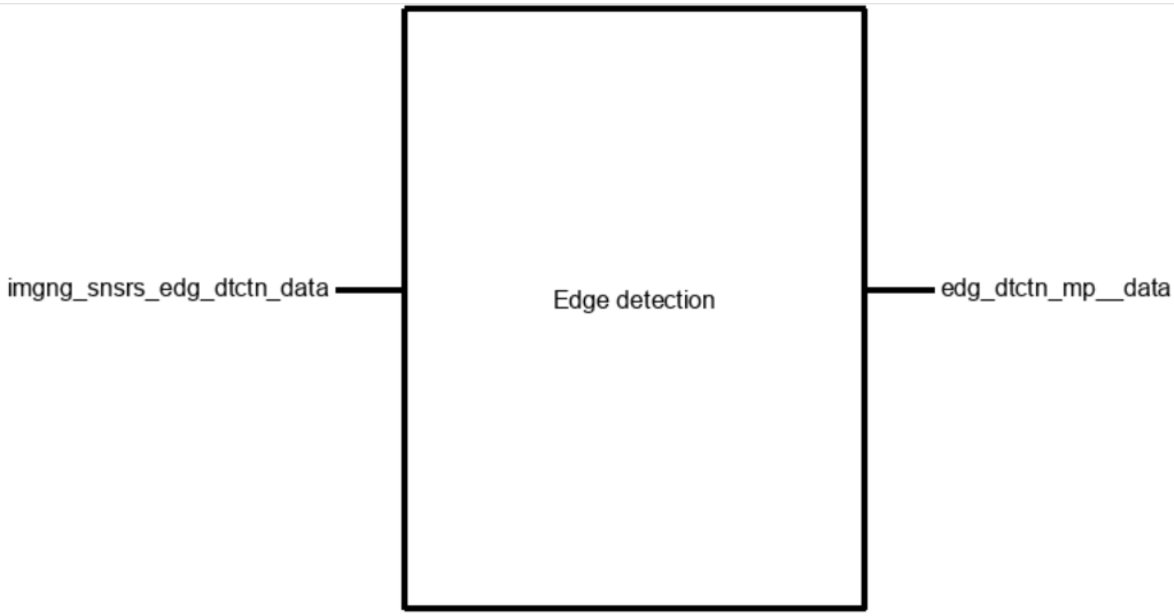

Figure 17: black block diagram

4.7.3 General Validation

The Edge detection technique that will be used in this project is Canny's edge detection which is a popular method for edge detection in computer vision and image processing, and is often used in autonomous robotics due to its effectiveness and efficiency. One of the main advantages of the Canny edge detection algorithm is its ability to accurately detect edges while minimizing false positives and false negatives which is achieved through a multi-stage process that includes smoothing the image, calculating gradient magnitude and direction, and performing non-maximum suppression and thresholding. Canny edge detection is also adaptable to different levels of noise and contrast in the image, which is important for autonomous robots that may encounter varying environments and lighting conditions. Therefore, Canny edge detection is computationally efficient, making it a suitable choice for real-time applications on our autonomous robotics, where fast processing times are necessary.

4.7.4 Interface Validation

imgng_snsrs_edg_dtctn_data : Input

Table 28: Block Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Data Rate: Minimum of 30 frames per second	This minimum data rate will allow the robot to sense the movement of its environment.	Data rate of 30 frame per second is the data rate of the video. Capturing images at this rate will gives the robot the same data as video without having it to do further video processing.
Other: RGB image	This will allow the edge detection process to make use of the original RGB image in further processing.	RGB images allow the better results in Canny's edge detection [1].
Other: .png file type	This will reduce the complexity of image processing steps (file conversion) needed in edge detection and path finding.	In Canny edge detection the common input type is .png.

edg_dtctn_mp__data : Output

Table 29: Block output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Datarate: Minimum of 30 frames per second	This minimum data rate will allow the robot to sense the movement of its environment.	Data rate of 30 frame per second is the data rate of the video. Capturing images at this rate will gives the robot the same data as video without having it to do further video processing.
Other: vector data of contour in coordinates formatting in .txt file for each frame	This vector data as coordinate .txt file is easy for further processing of determining the distance away from the obstacle and edges.	The location of each contour coordinate will inform the robot of its environment including the obstacle and edges.

Other: grayscale contour image	Contour images will allow further processings on mapping block to determine the distance away from the obstacle and edges.	

4.7.5 Verification Plan

The first verification plan will be performed off the robot to test if the edge detection is able to perform image processing and output data in the desired format.

1. Inputting images to the block by making it takes output from the camera module.
2. Run the python script to start image processing.
3. See if the block is outputting contour mapping images and saved on the working environment in a proper format which is grayscale images PNG file.
4. See if the block is outputting contour vectors as x,y coordinate and saved on the working environment in a text file.
5. To verify the frame rate, capture the timer and see if it is actually capturing images and outputting contour vector data at the rate of 30 files per second.
6. Turn off all programs and power off the camera.

4.7.6 References and File Links

4.8.6.1 References

[1] Comparative analysis of edge detection between grayscale and color image (May, 2016) Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA. Available at: https://www.caeaccess.org/archives/volume5/number2/malik-2016-cae-652230.pdf [Accessed: March 12, 2023].

4.7.7 Revision Table

Table 30: Revision table for section 4.7

Date	Author	Description
3/12/2023	Budsakol Tiangdah	Initial Draft: Outlined document and filled in each section with relevant research content.

4.8 Imaging Sensors

Block Champion: Budsakol Tiangdah

4.8.1 Description

The image sensor is the block that takes in the environmental data from outside of the system which tells the robot what the environment around it looks like including obstacles, edges, paths, signs, and free spaces. The robot will get the environmental data through both Lidar (Light Detection and Ranging) sensor that we currently have on board and the camera that will be added to the robot according to the recommendation from our project partner in order to incorporate image processing capabilities, reinforce object identification, improve the accuracy in edge detection, and enhance some safety features. Since the Lidar sensor was already implemented and used on board, this block will mainly focus on implementing and utilizing the camera capabilities. The camera will capture its environment as still images or video frames depending on the task that the robot is performing. Then sending those digital data to the onboard processor to perform image processing. The data from a camera will be used by other blocks including the edge detection block and pathfinding block allowing the ability to fulfill the requirement of doing edge detection, object identification and avoidance, object collision prevention, and path generation. The robot will use these data to control the motor controller block to perform desired autonomous behaviors and accomplish its tasks.

Overall, this image sensor block on an autonomous robot plays a crucial role in allowing the robot to perceive and understand its environment, make decisions, and perform tasks in a more autonomous manner.

4.8.2 Design

The objective of this block is to add the camera to the system which will provide the robot the valuable data and information, such as real-time images, that can be used to improve the robot's performance. This block needs to perform two tasks including capturing the images or video frame of the robot’s environment and communicating or transferring data to the central processing unit to perform image processing which is referred to as the mapping code block and edge detection code block. The camera has only one physical connection to the single board computer hardware through USB 2.0 which will do both powering the camera and transferring the data to the edge detection block and mapping block.

The camera will capture still images which give the robot high-resolution information about the scene to perform tasks such as object recognition or semantic segmentation. The camera will capture still images every 33.33 milliseconds, giving the data rate of 30 frames per second. The captured data will be stored on could as grayscale images in PNG file type, ready to be sent to the mapping and edge detection clock for further analysis and usage of determining the location of obstacles, and distance away from the edges of the pathway which will be useful for determining the path that the robot will take to reach its designated destination safely.

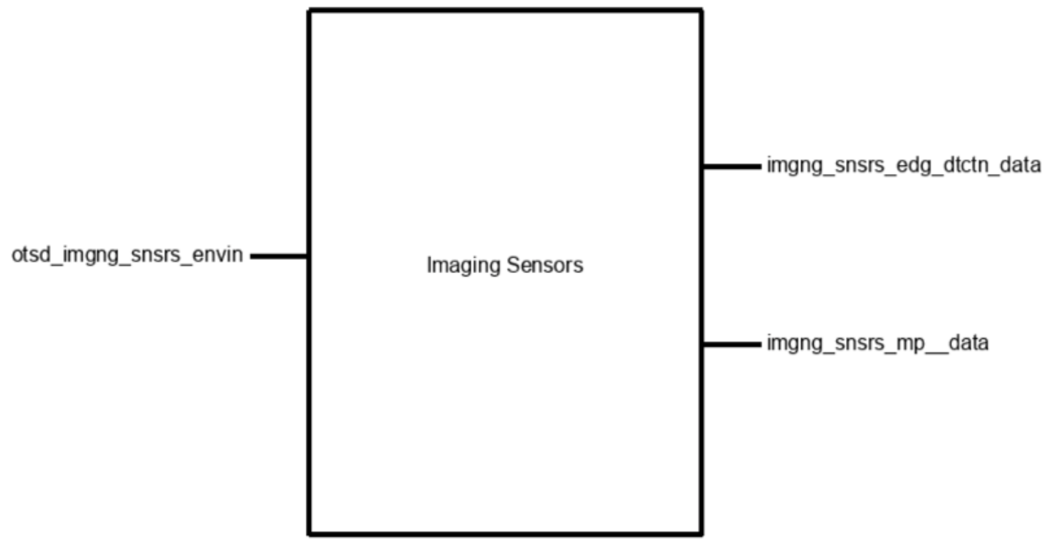

Figure 18: black block diagram

To capture the picture as desired file type at a specific rate, the camera will run a python script which will use opencv to capture and do the image processing steps.

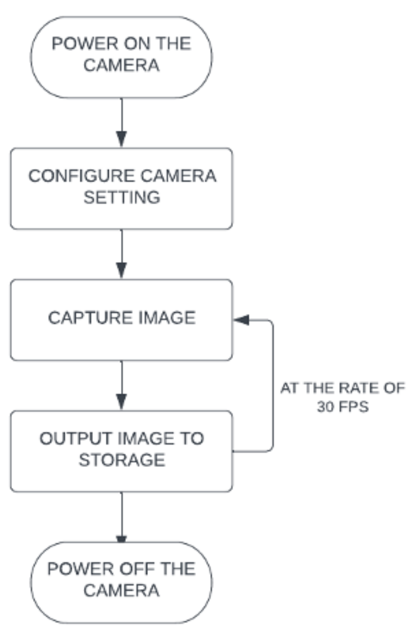

Figure 19: flow charge of the code for capturing and storing images

The first step is to connect to the camera then configure some camera setting including frame rate which is 30 frames per second, image resolution (640*480). Then the camera will start capturing images and store them on cloud storage in the desired format so that the image processing blocks (mapping and edge detection) can pull these data out from the storage for further usage. It will keep capturing the image and send them to storage at the rate of 30 frames per second until the camera is turned off or the robot turns to sleep mode that no longer needs the environmental data to navigate.

4.8.3 General Validation

The camera that will be installed on board is Astra S which is manufactured by Orbbec. The reasons that making this specific module of camera well-suited for use in our autonomous package delivery robots are:

- Compact Size: It has a compact and lightweight design that makes it easy to integrate into a wide range of robotic systems which makes it an ideal choice for use in smaller robots and drones including this autonomous package delivery robot.

- Stereo Processing: It features two high-resolution CMOS sensors that provide stereo image data that allows the camera to perform real-time depth calculation, which is crucial for obstacle avoidance, mapping, and navigation.

- Onboard Processing: It has an onboard processor that performs real-time stereo processing and depth calculation which helps reduce the computational load on the main processor of the robot which allows the robot to perform these tasks in real-time and more efficiently.

- High-Quality Images: It produces high-resolution images with good color and contrast, which is important for object recognition and monitoring the traffic of its environment.

There are several ways of capturing environmental data which are always streaming video, capturing only images, or dynamically capturing both video and images depending on the movement that the robot detects, for example, using still image capturing for initial object detection, then using video frames for tracking the traffic of the detected objects over time.

For our design choice, we will only capture still images over time and store them as grayscale PNG files because storing them as grayscale images is easier for further usage. The grayscale images are less complex and easier for processing because they only have one component of information which is the intensity that represents the brightness of that specific pixel, making it easier to compute the gradient magnitude of the image for edge detection, while RGB or color images, each pixel has three color components including red, green, and blue. Thus, in color image processing, vector-valued image functions need to be used instead of scalar image functions [1]. Using grayscale images also reduces the resource of the onboard computer needed for the processing process.

The images will be captured at the rate of thirty frames per second which is equal to the data rate of the video. The data at this rate will give the robot temporal information about the scene allowing object tracking and motion estimation which make the robot able to sense the movement of objects around it and behave accordingly to its environment which is necessary when it needs to deal with the flow of its environmental traffic. In another word, this data rate will give the robot the same set of data as capturing video while using fewer resources to store and process because when doing the edge detection technique, the robot will not need to convert the data fom video file type into images making the processing step go faster.

4.8.4 Interface Validation

otsd_imgng_snsrs_data : Input

Table 31: Environmental Data Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Other: Field of view 60°H x 49.5°V (73°D)	The camera is capable of capturing the specified field of view without turning, so it can be stationarily attached to the robot.	Orbbec-Astra-Wiki Documentation page 3[2]: Field of view 60°H x 49.5°V x 73°D
Other: minimum range of 0.4 meters	The minimum range of 0.4 meters allows the robot to get minimum data of its environment that sufficient for the necessary processing	Astra S, Technical Specs [3]: range 0.4 – 2 meters
Other: maximum range of 2 meters	The data generated beyond 2 meters from the robot will become unreliable.	Astra S, Technical Specs [3]: range 0.4 – 2 meters

imgng_snsrs_edg_dtctn_data : Output

Table 32: Image Data Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Data rate: Minimum of 30 frames per second	This minimum data rate will allow the robot to sense the movement of its environment.	Data rate of 30 frame per second is the data rate of the video. Capturing images at this rate will gives the robot the same data as video without having it to do further video processing.
Other: Output as .png file type	This will reduce the complexity of image processing steps (file conversion) needed in edge detection and path finding.	In Canny edge detection the common input type is .png.
Other: Output as RGB image	This will allow the edge detection process to make use of the original RGB image in further processing.	RGB images allow the better results in Canny's edge detection [1].

imgng_snsrs_mp_data: Output

Table 33: Image Data Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Data rate: Minimum of 30 frames per second	This minimum data rate will allow the robot to sense the movement of its environment.	Data rate of 30 frame per second is the data rate of the video. Capturing images at this rate will gives the robot the same data as video without having it to do further video processing.
Other: Output as .png file type	This will reduce the complexity of image processing steps (file conversion) needed in edge detection and path finding.	In Canny edge detection the common input type is .png
Other: Output as grayscale image	This will reduce the complexity of image processing steps (file conversion) needed in edge detection and path finding.	In Canny edge detection before detecting any edge images need to go through noise reduction process which convert them to grayscale images.

4.8.5 Verification Plan

The first verification plan will be performed off the robot to test if the camera is able to perform image capturing and storing data in the desired format.

1. Power on the camera module by connecting it to the computer.

2. Wait for startup configurations to process.

3. Run the python script to start capturing frames.

4. See if the image is captured and saved on the working environment in a proper format which is grayscale images PNG file.

5. To verify the range of the images, place some objects at the minimum and maximum range. In this case, for Orbbec Astra S, the minimum range is 0.4 meters, and the

maximum range is 2 meters. Then capture the environmental data and see if those objects can be seen clearly in the picture.

6. To verify the field of view of the images, which indicates the area that the camera can capture in images, place the object at a known distance d then capture a picture. The field of view can be calculated using the following formula, where d is the distance between the camera and the subject, h is the height of the object from the ground, and f is the focal length of the lens.

7. To verify the frame rate, capture the timer and see if it is actually capturing images at the rate of 30 frames per second.

8. Turn off all programs and power off the camera.

4.8.6 References and File Links

4.8.6.1 References (IEEE)

[1] Comparative analysis of edge detection between grayscale and color image (May, 2016) Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA. Available at: https://www.caeaccess.org/archives/volume5/number2/malik-2016-cae-652230.pdf [Accessed: February 10, 2023].

[2] Orbbec-Astra-Wiki Documentation (2018) orbbec3d. Orbbec. Available at: https://readthedocs.org/projects/astra-wiki/downloads/pdf/latest/ [Accessed: February 10, 2023].

[3] Astra S, Technical Specs *Orbbec*. [Online]. Available: https://shop.orbbec3d.com/Astra-S?quantity=1&custcol_ava_item=Astra%2520S&custcol_ava_incomeaccount=Sales%2520of%2520Product%2520Income&custcol_ava_pickup=F. [Accessed: February 10, 2023].

[4] Comparative analysis of edge detection between gray scale and color image (no date) Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA. Available at: https://www.caeaccess.org/archives/volume5/number2/malik-2016-cae-652230.pdf

[Accessed: February 10, 2023].

4.8.6.2 File Links

https://readthedocs.org/projects/astra-wiki/downloads/pdf/latest/

https://github.com/MyNameIsCosmo/astra-wiki/blob/master/docs/source/examples.rst

4.8.7 Revision Table

Table 34: Revision table for section 4.8

Dates	Author	Revision
1/15/2023	Budsakol Tiangdah	Initial draft: setting up template, description, design draft, revision table
1/19/2023	Budsakol Tiangdah	General validation, references, interfaces validation, verification plan
2/9/2023	Budsakol Tiangdah	Update interfaces validation
2/10/2023	Budsakol Tiangdah	Update the design, references
2/11/2023	Budsakol Tiangdah	Verification plan update
2/20/2023	Budsakol Tiangdah	Update interfaces to match with the online validation form, black blog diagram update
2/25/2023	Budsakol Tiangdah	Update the verification plan to match the updated interfaces

5. System Verification Evidence

5.1 Universal Constraints

5.1.1 The system may not include a breadboard

System does not contain any breadboards for any of the blocks. All blocks of the system are either purchased modules, or designed by the team. There is a teammate designed PCB for the robot. For this design, the universal constraints were kept in mind to ensure that all connections are secure and will be reliable over time

Below is a picture of all of the hardware components for this project:

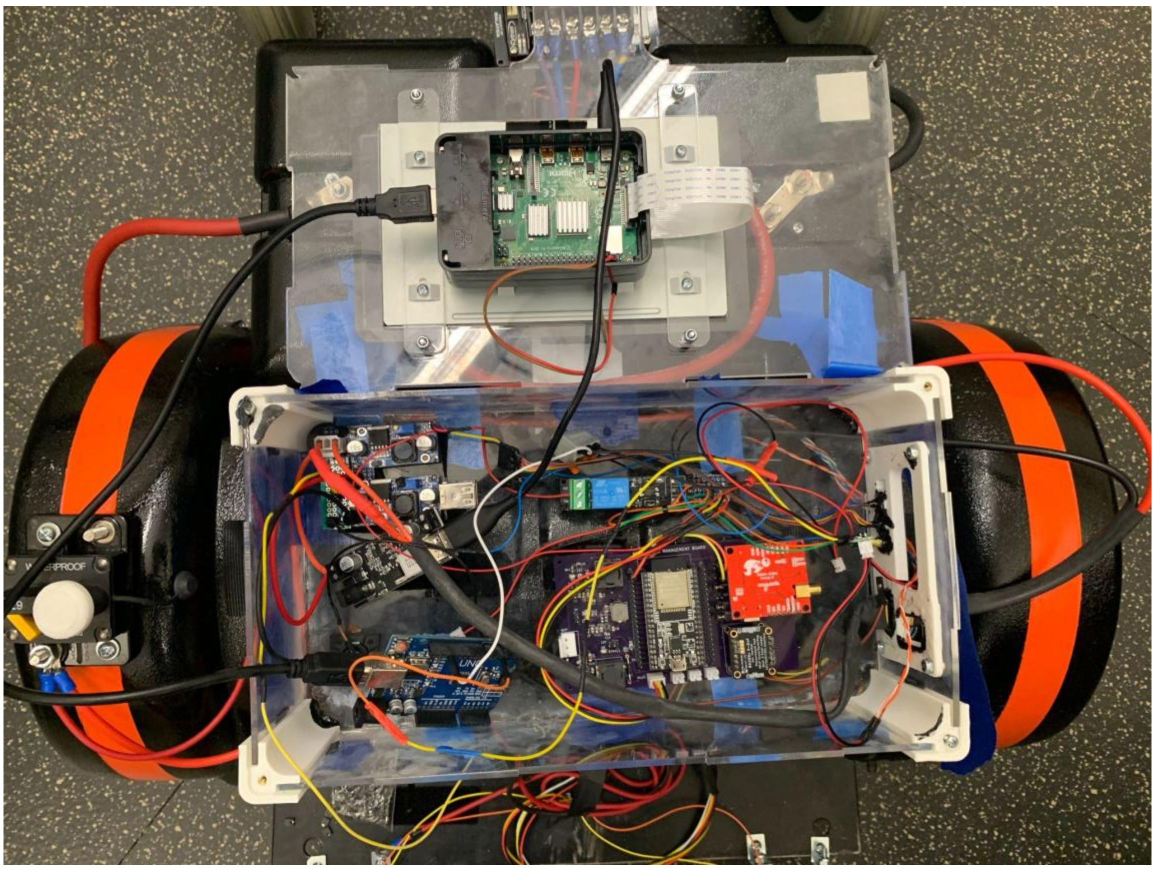

Figure 20. Hardware Components

5.1.2 The final system must contain a student-designed PCB

Figures X and Y contain the schematic and footprint of the student designed PCB for the project. This PCB comes from the power management block, and provides most of the power needs for the various components of the robot. As described in section 4 of this document, the power management block takes power from the onboard batteries and sends it to several different voltage converters. These converters act to power the various components and sensors of the robot. There are approximately 106 SMT connections on the PCB.

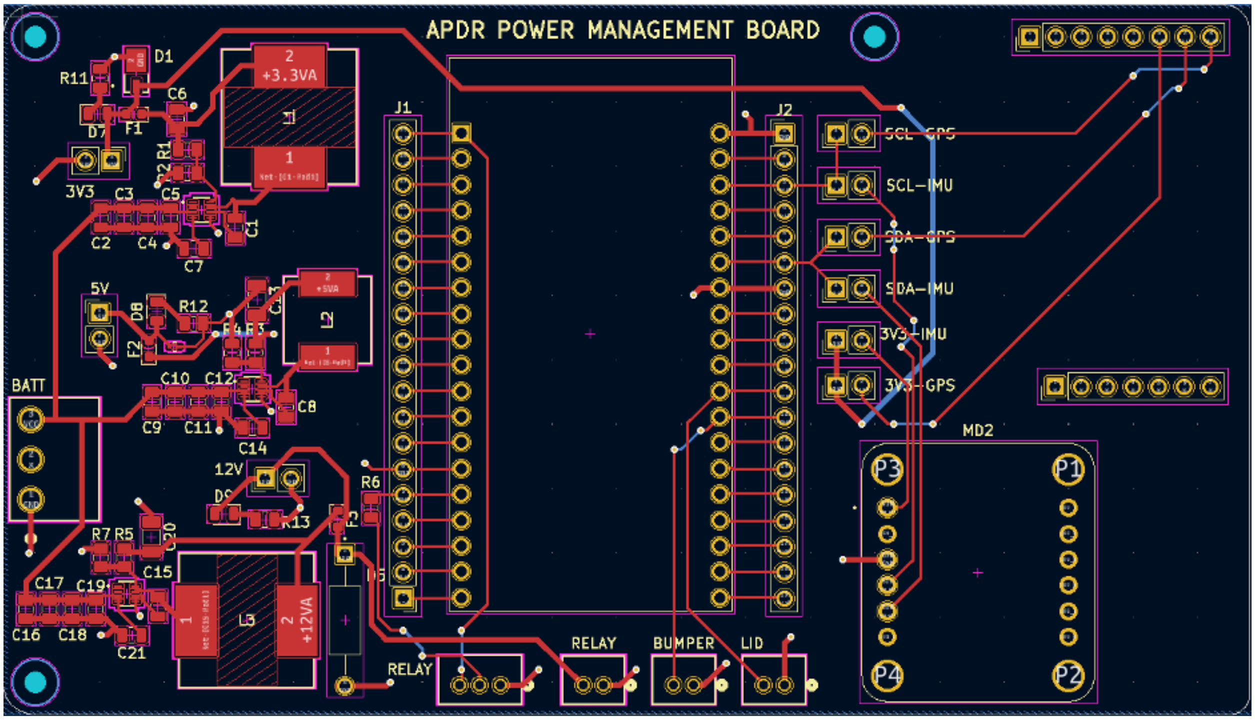

Figure 21. Student designed PCB

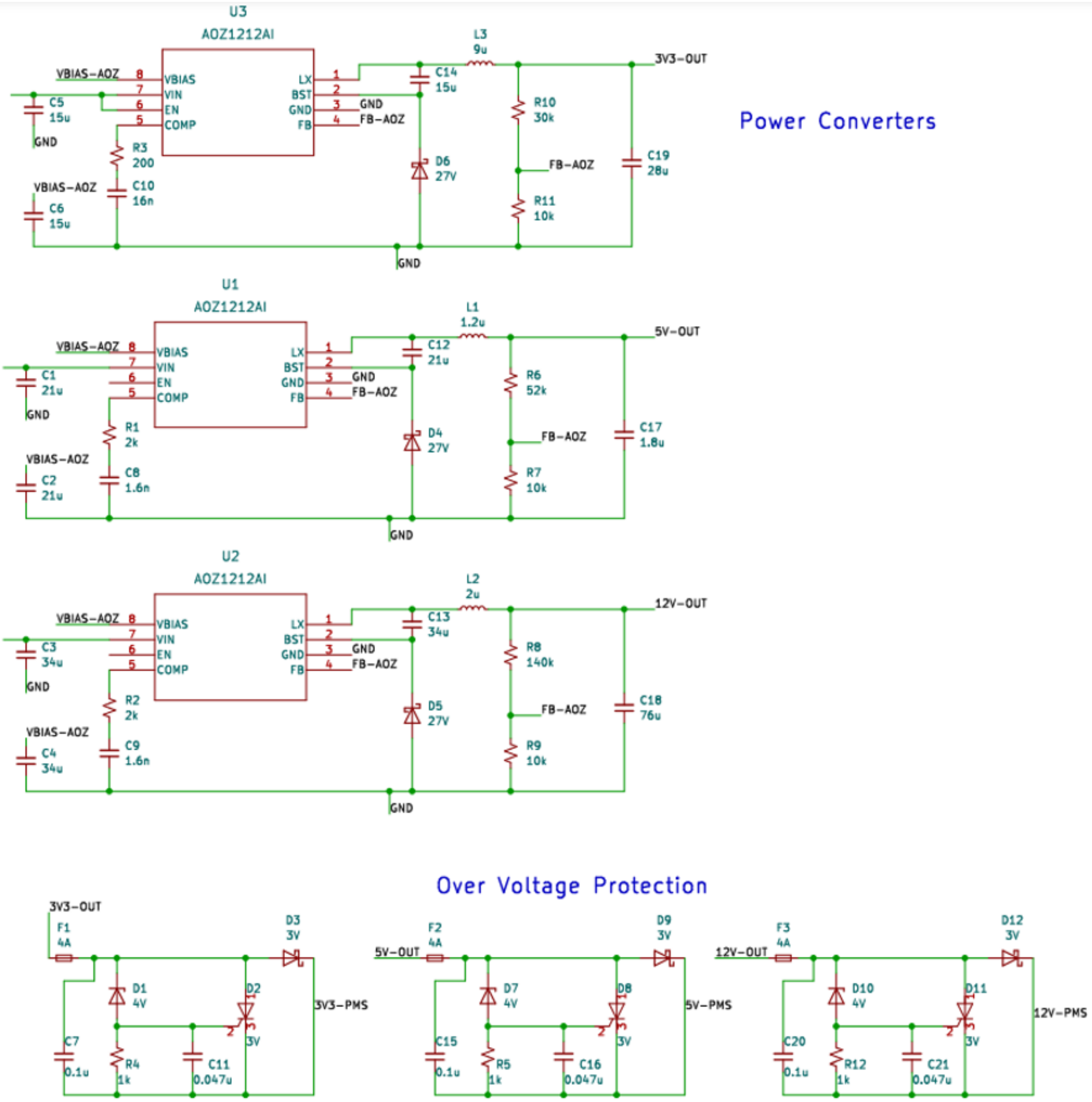

Figure 22: Student designed PCB footprint

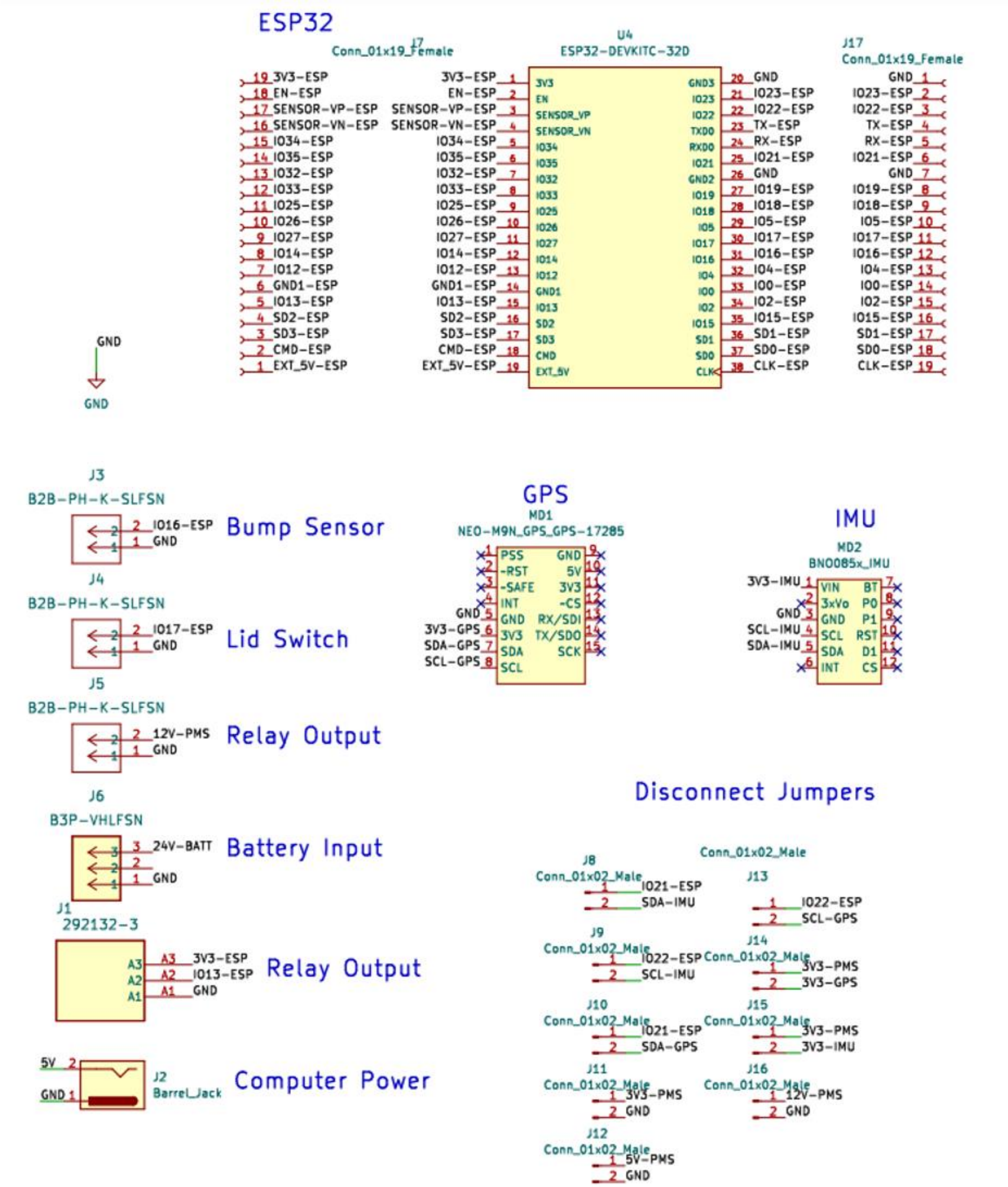

Figure 23: Student designed PCB footprint

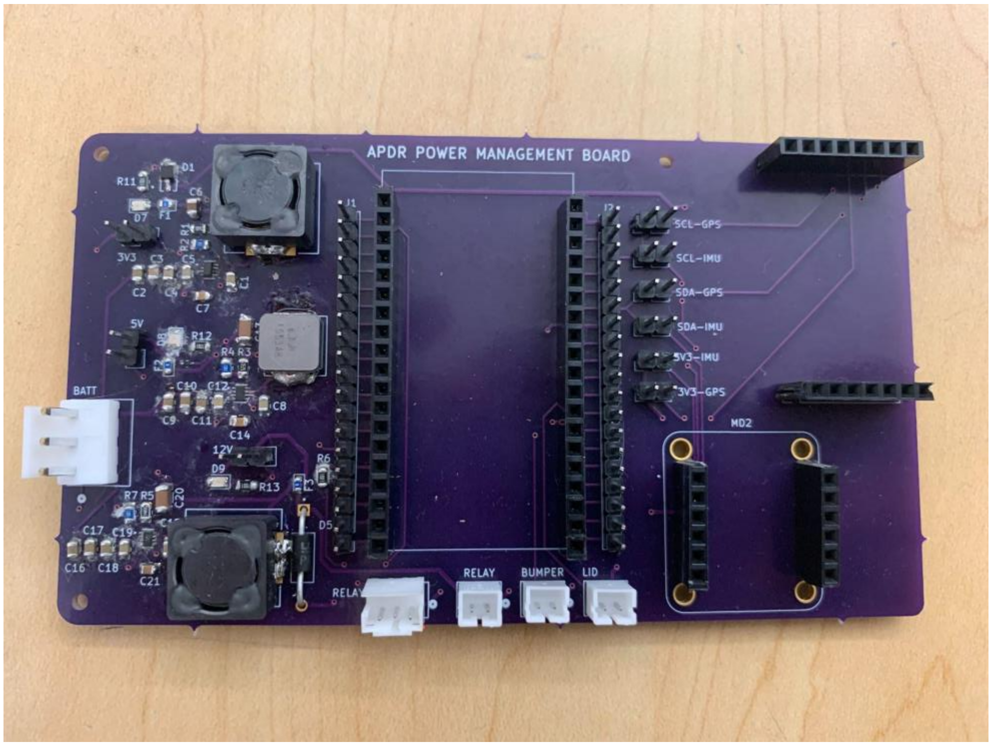

Figure 24. Student designed PCB (Assembled)

5.1.3 All connections to PCBs must use connectors

Figures 20, 21 and 22 above show the footprint and schematic, respectively, for the student designed PCB. As seen from the figures, all connections to the PCB have a dedicated connector. These connections are done either through JST, jumper, or USB cable. All other connections from purchased modules will be through USB or barrel jack connection.

5.1.4 All power supplies in the system must be at least 65% efficient

The power management block in section 4 for this project contains all of the information regarding power usage for the various modules. The power supplies for the power management block are all designed with the same switching buck converter controller, using different circuit layouts to obtain different output voltages. Switching buck converters typically show high efficiency when tested. The specific buck converter used for this project has a rated efficiency greater than 90%, but actual efficiencies for this project vary. According to the datasheet for the used buck converter and the various components used, the efficiency for each converter exceeds 65%. However, these values have not been physically tested.

5.1.5 The system may be no more than 50% built from purchased 'modules'

Differentiating between what was feasible to construct from scratch in the given time frame and what needed to be sourced externally took some deliberation at the start of this project. The Autonomous Package Delivery Robot system is an extremely complex project, the justification for the authorship of each block will be described in the table below.

Table 35: Team Authorship Verification

Project Block	Justification of Team Authorship
Single Board Computer Hardware	A single board computer contains the purchased module which is Nvidia Jetson Nano. However, to make it compatible with other modules we have on the robot, it requires additional configuration by the team. Therefore, this block is about 50% purchased modules.
Single Board Computer Software	This block is mainly a software block which is developed by the team and contains no purchased modules.
Motor Controller	This block consists of motors, wheel encoders, and a microcontroller to interpret the input from the wheel encoders and output a control signal to the motors. The protocol for communicating with the motors from the ESP32 needed to be created from scratch. The wheel encoders also required a custom mounting setup. The code to interface the input received from the encoders with the output being sent to the motors in order to prevent drift required custom development. The motors are an inherent component to the recycled electric wheelchair base that is a staple to this project, so they shall be discounted. Given the software interfacing with both modules and the custom mounting of the wheel encoders, this block can be said to contain greater than 50% custom developed modules
Power Management System	All circuitry was developed internally and placed on the custom PCB. This block was 100% developed internally and contains no purchased modules.
Enclosure Design	The enclosure is for the Nvidia Jetson Nano and power management board, incorporating cord protection to ensure that the power cord is secure and protected from damage. The enclosure was 3D printed. It will provide protection to both devices while keeping them organized and easily accessible. Additionally, we will incorporate cord protection to ensure that the power cord is secure and protected from damage.This block was 100% developed internally and contains no purchased modules.

Pathfinding Software	This software-exclusive block was built to interface the team's custom mixture of sensors and design goals. This includes the ability to create waypoints from predefined destinations and using data from camera and LiDAR to accurately localize the APDR within its environment and bring it to the desired destination. This block contains no purchased modules
Imaging Sensors	This block is a combination of hardware and software block. The hardware part is a camera which is a purchased module. The software to capture and process the images is fully developed by the team. This block contains about 50% purchased modules.
Edge Detection	This block is a software block. All software to convert the camera data to a useful coordinate system, filter images, isolate edges, and output data was developed internally. This block contains no purchased modules

The tables show that blocks vary greatly in the amount of purchased modules that were implemented in their design. It can be safely inferred that the project in its entirety contains no more than 40% purchased modules.

5.2 Requirements

5.2.1 Ease of Use

5.2.1.1 Project partner requirement: The system needs to be easy to use.

5.2.1.2 Engineering requirement: The system will have 9 out of 10 users set a destination and load a package in under 2 minutes.

5.2.1.3 Verification Process

To verify that the system is easy to use we will use the interactive user interface to let 10 users enter the desired destination and load the package. Each user will be timed and it should not take more than 2 minutes per person to do so. The test should be successful for at least 9 users.

5.2.1.4 Testing Evidence

https://youtube.com/shorts/NsNMj3wP4k4?feature=share

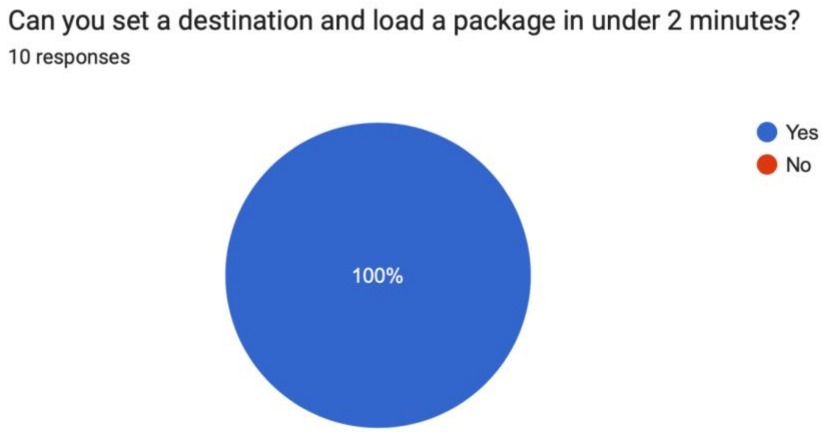

Figure 25: Survey result for ease of use verification

Name	Email	Can you set a destination and load a package in under 2 minutes?	Can you identify the emergency stop button on the robot?	Do you think the emergency stop button is easy to access?
Thatchapan Pianpucktr	pianpuct@oregonstate.edu	Yes	Yes	Yes
Parm kangsathein	Kangsatp@oregonstate.edu	Yes	Yes	Yes
Nichaphat Laotaweerungrueng	laotawen@oregonstate.edu	Yes	Yes	Yes
Jia Wei Cheng	chengjia@oregonstate.edu	Yes	Yes	Yes
Hoang Ngoc (Jenny)	lehoan@oregonstate.edu	Yes	Yes	Yes
Miguel Santiago	santiami@oregonstate.com	Yes	Yes	Yes
Chelsey Chiu	gi25ni@gmail.com	Yes	Yes	Yes
Jazmin cartagena	cartagej@oregonstate.edu	Yes	Yes	Yes
Lyon kee	Keel@oregonstate.edu	Yes	Yes	Yes
Ice Chamrasromran	icepcham@gmail.com	Yes	Yes	Yes

Figure 26: Survey results details

5.2.2 Edge Detection

5.2.2.1 Project partner requirement: The robot should stay on the sidewalk.

5.2.2.2 Engineering requirement: The system will determine the bounds of pathways and maintain a minimum distance of 15cm from the edge of said pathway.

5.2.2.3 Verification Process

Before starting the verification process, a circuit which has at least one sidewalk representation will be set up. A 15 cm threshold will be marked for visualization. To verify that the system is able to stay on the sidewalk we will power the system and wait for the booting process to finalize. Then, we will set up and initialize the software necessary to get the robot to a neutral initial state. Finally, a destination will be set. The robot will reach the destination without going over the defined 15 cm threshold from the sidewalk representation.

5.2.2.4 Testing Evidence

Code Explanation:

https://drive.google.com/file/d/1PCFUl3_bBWPHGYcOvZ4v-8buXrPwtJsG/view

Video Demonstration

https://drive.google.com/file/d/1PRBAI5-CjEEC2gFKo3D8RE_pV6J1Euqq/view

5.2.3 Emergency Shutdown

5.2.3.1 Project partner requirement: The robot should have an accessible and convenient on-board method to stop the robot in the case of an emergency.

5.2.3.2 Engineering requirement: The system will have an easy to access emergency stop button as reported by 9 out of 10 users and the robot will stop if it collides with an object of 10lbs or more.

5.2.3.3 Verification Process

Before starting the verification process, the delivery robot will be held in suspension to avoid collisions and have a controlled environment. The robot will be set to forward motion. A team member will trigger the emergency shutdown button. The robot will lose power. This will be repeated for 10 different people confirmed via signature.

5.2.3.4 Testing Evidence

https://www.youtube.com/watch?v=SlkSXwmcsJs

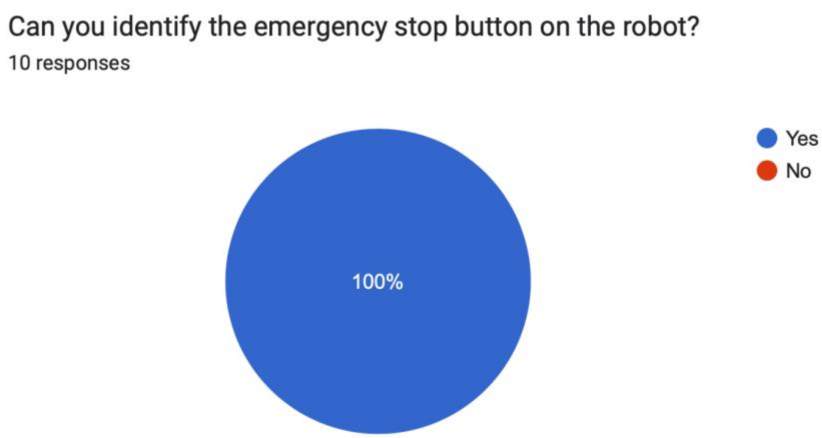

Figure 27: Survey result for emergency stop verification

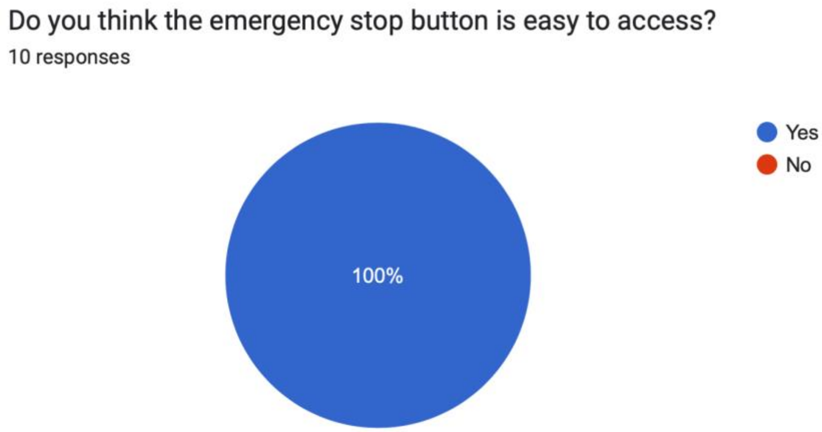

Figure 27: Survey result for emergency stop verification

Name	Email	Can you set a destination and load a package in under 2 minutes?	Can you identify the emergency stop button on the robot?	Do you think the emergency stop button is easy to access?
Thatchapan Pianpucktr	pianpuct@oregonstate.edu	Yes	Yes	Yes
Parm kangsathein	Kangsatp@oregonstate.edu	Yes	Yes	Yes
Nichaphat Laotaweerungrueng	laotawen@oregonstate.edu	Yes	Yes	Yes
Jia Wei Cheng	chengjia@oregonstate.edu	Yes	Yes	Yes
Hoang Ngoc (Jenny)	lehoan@oregonstate.edu	Yes	Yes	Yes
Miguel Santiago	santiami@oregonstate.com	Yes	Yes	Yes
Chelsey Chiu	gi25ni@gmail.com	Yes	Yes	Yes
Jazmin cartagena	cartagej@oregonstate.edu	Yes	Yes	Yes
Lyon kee	Keel@oregonstate.edu	Yes	Yes	Yes
Ice Chamrasromran	icepcham@gmail.com	Yes	Yes	Yes

Figure 28: Survey results details

5.2.4 Lockbox User Safety

5.2.4.1 Project partner requirement: The system needs to be safe for users.

5.2.4.2 Engineering requirement: The system will not allow for motion when the lockbox sub-system is open.

5.2.4.3 Verification Process

For this verification, the LED light will be used to indicate if there is a power to the system. The LED will indicate the following status.

1. When the lid is opened, meaning the user must be interacting with the robot, so it is not safe for the robot to move, the LED light will turn on indicating that the power of the robot was cut, so it won't be able to move. A move forward signal will be inputted to prove the robot will not respond.
2. When the lid is closed, meaning it is safe for the robot to move, the LED light will turn off indicating that the power of the robot resumes. A move forward signal will be inputted, the robot is expected to move forward uninhibited.

5.2.4.4 Testing Evidence

https://drive.google.com/file/d/1Szu1SQpeWBvFieh42DizURZyFVjvrWTH/view

5.2.5 Object Avoidance

5.2.5.1 Project partner requirement: The robot should be able to go around stationary objects in its path.

5.2.5.2 Engineering requirement: The system will traverse around stationary objects in its path and not get closer than 15 cm to said object.

5.2.5.3 Verification Process

Before starting the verification process, a circuit which has at least one stationary object representation set up. A 15cm boundary will be set up around the stationary object to demonstrate that the robot does not cross the 15 cm threshold. To verify that the system is able to transverse the circuit without collision, we will power the system and wait for the booting process to finalize. Then, we will set up and initialize the software necessary to get the robot to a neutral initial state. Finally, a destination will be set. The robot will reach the destination without colliding with the stationary object representation considering its 15cm threshold.

5.2.5.4 Testing Evidence

https://www.youtube.com/watch?v=HPu2KiqEki0

5.2.6 Object Collision Prevention

5.2.6.1 Project partner requirement: Must handle objects in its path.

5.2.6.2 Engineering requirement: The system will not impact an unavoidable object of 10cm x 10cm that is 100cm away when traveling at 3 kmph or more.

5.2.6.3 Verification Process

Before starting the verification process, a circuit which has at least one stationary object representation set up. To verify that the system is able to avoid collision, we will power the system and wait for the booting process to finalize. Then, we will set up and initialize the software necessary to get the robot to a neutral initial state. Finally, a destination will be set. The robot will reach the destination without colliding with the stationary object representation.

5.2.6.4 Testing Evidence

Code Explanation:

https://drive.google.com/file/d/1PBj7SXB5_fYH1K9mO48b_XGp7pJ8O1uN/view

Video Demonstration:

https://drive.google.com/file/d/1PVkKmCRIDszMXWhWjrZ2UeV6cU6Isdw4/view

Robot Demonstration:

https://www.youtube.com/watch?v=PvuUcVHoefk

5.2.7 Pathfinding Software Implementation

5.2.7.1 Project partner requirement: The system should be able to transverse from a predefined point to another.

5.2.7.2 Engineering requirement: The system will follow a predefined path of at least 20M to its destination and deviate from that path by no more than 10 meters.

5.2.7.3 Verification Process

Before starting the verification process, a circuit with a clearly defined start and finish is set up. To verify that the system is able to transverse the circuit, we will power the system and wait for the booting process to finalize. Then, we will set up and initialize the software necessary to get the robot to a neutral initial state. Finally, a destination will be set. The robot will reach the destination.

5.2.7.4 Testing Evidence

Code Explanation:

https://drive.google.com/file/d/1P49TFRp3_LuEMcq-18d4zlrmE2Aq-h3G/view

Algorithm:

https://drive.google.com/file/d/1PWW-Yk-KzTpbGYz2NSCkLU5M899IV9oO/view

Video Demonstration:

https://drive.google.com/file/d/1Q1S35--PcbDRhADhqgBAA69_ueA8B3v_/view

Robot Demonstration:

https://www.youtube.com/watch?v=zgyjR60dlr4

5.2.8 System Motion Control

5.2.8.1 Project partner requirement: Robot must be able to correctly turn and go forwards.

5.2.8.2 Engineering requirement: The system must move forward, reverse, turn/spin left and turn/spin right.

5.2.8.3 Verification Process

To verify that the system is able to turn and go forward, we will power the system and wait for the booting process to finalize. Then, we will set up and initialize the software necessary to get the robot to a neutral initial state. Finally, the robot will be directed to turn right. After the first action has been completed the process will be repeated individually for the other two commands; turn left and go forward.

5.2.8.4 Testing Evidence

https://www.youtube.com/watch?v=CeiJA9zzJW8

5.4 Revision Table

Table 36: Revision table for section 5.4

Date	Author	Description
5/12/2023	Budsakol Tiangdah	Add proof of verification (survey results)
4/14/2023	Bolivar Beleno	Modified and validated verification process description
3/13/2023	Bolivar Beleno	Added Verification Process, must be discussed by group for approval.
3/13/2023	Josep Borisch	Update universal constraints
3/12/2023	Budsakol Tiangdah	Initial Draft: Outlined document and fill in draft of each section.
3/12/2023	Sun Junjie	Lock box user safety verification process and testing evidence update

6 Project Closing

6.1 Future Recommendations

6.1.1 Technical recommendations

Replacement of ROS System with Linux Computer and Python-based A* Algorithm

This year, the team replaced the ROS system with a Linux computer for computation tasks and implemented an A* algorithm in Python to facilitate the robot's movement. The use of a Linux computer provided a more robust and stable platform, while the A* algorithm allowed for more efficient path planning. For future teams interested in further refining this aspect, we recommend reviewing the following resources:

A Formal Basis for the Heuristic Determination of Minimum Cost Paths.[1]

PythonRobotics (2021). PythonRobotics: A* Algorithm. [2]

Integration of Depth Camera for Enhanced Navigation

The team successfully integrated a depth camera into the robot, enabling it to better perform tasks and navigate its environment. The depth camera allowed for more accurate obstacle detection, improved edge detection, and contributed significantly to the robot's traversal capabilities. For future teams interested in further improving or integrating different depth camera models, we recommend reviewing the following resources:

Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. [3]

Intel RealSense Depth Cameras. [4]

Utilization of LIDAR and Depth Camera for Map Construction

This year, the team successfully utilized a combination of LIDAR and depth camera to construct a detailed map of the robot's environment, which was then used by the A* algorithm for efficient path planning. The fusion of LIDAR and depth camera data provided a more comprehensive representation of the environment, enabling the robot to navigate more accurately and avoid obstacles effectively.

For future teams interested in further refining this approach or exploring alternative sensor fusion techniques, we recommend reviewing the following resources:

Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph. IEEE International Conference on Robotics and Automation [5]

Comparing ICP Variants on Real-World Data Sets: Open-Source Library and Experimental Protocol. [6]

RPLIDAR: A High-Performance 360-degree LIDAR Sensor. [7]

ROS Point Cloud Library (PCL). [8]

Addressing PCB Design and Power Supply Challenges

The team encountered challenges in designing a functional printed circuit board (PCB) and implementing an efficient buck converter to supply power to the onboard computer. Proper PCB design and a reliable power supply are crucial for the robot's stable operation and performance. To address these challenges, we recommend the following strategies:

Invest time in learning PCB design principles and using PCB design software to ensure a well-designed and reliable board. Resources such as textbooks, online tutorials, forums, and articles can be helpful in developing a strong foundation in PCB design. We recommend the following resources for a better understanding of PCB design:

Printed Circuit Board Designer's Reference: Basics. Prentice Hall. [9]

Fedevel Academy. (n.d.). Online PCB Design Courses. [10]

Altium Designer. (n.d.). PCB Design Tutorials. [11]

6.1.2 Global impact recommendations

One global impact recommendation is to invest in the development of more sophisticated and aesthetically pleasing user interfaces of the robot to enhance the ability to interact effectively with users and maximize the economic impact of the robot by making it appealing and engaging to customers. This could include features such as personalized greetings, interactive displays, and intuitive touch screens that allow customers to easily customize their delivery preferences or provide feedback. Also, enhancing the ability of the user interface so that there will no longer need humans to operate can create an impact on the economy and environment as well. For developing user interfaces, look into the reference [12].

Another global impact recommendation is to try and use environmentally friendly materials moving forward. A sort of baseline has been developed at the conclusion of our part of the project, and now it’s time for a new rendition and improvements. Some of the parts used – including the plastics, electrical components, and other materials – could be substituted with eco-friendly materials very easily. As such, to reduce global and environmental impacts of this project even more, we highly recommend sourcing eco-friendly materials if anything new is to be added to the project, or replacing parts with environmentally-friendly materials.

6.1.3 Teamwork recommendations

In our Autonomous Package Delivery Robot project, we have identified that communication and synchronization issues have arisen due to cultural differences and a lack of team-building activities. To address these challenges and improve our teamwork, we propose the following recommendations:
Cultural Awareness and Sensitivity Training
To enhance cross-cultural communication within the team, we recommend participating in cultural awareness and sensitivity training sessions. These sessions will help team members understand and appreciate each other's cultural backgrounds, values, and communication styles. By developing this understanding, team members will be more empathetic and able to communicate more effectively.

Resources:
Cultures and Organizations: Software of the Mind. McGraw-Hill. [13]

Team Building Activities
To enhance team synchronization and cohesion, we suggest organizing team-building activities that encourage collaboration and trust. These activities can range from problem-solving exercises to social gatherings outside the work environment. By participating in these activities, team members will develop stronger interpersonal relationships and improve their ability to work together effectively.

Resources:

On Becoming a Team Player. Team Performance Management: An International Journal- This article provides guidance on how to become a better team player and highlights the importance of team-building activities.[14]

Structured Communication Channels and Regular Meetings
To ensure clear and effective communication within the team, it is crucial to establish structured communication channels and hold regular meetings. We recommend using collaboration tools such as Slack or Microsoft Teams for daily communication and project management. Additionally, schedule weekly or bi-weekly meetings to discuss progress, address any issues, and share updates. These meetings will help the team stay synchronized and facilitate better communication.

Resources:
Collaboration tools:
Slack: https://slack.com/
Microsoft Teams: https://www.microsoft.com/en-us/microsoft-teams/group-chat-software

By implementing these recommendations, our team will be better equipped to address the communication and synchronization issues we have encountered. This will ultimately lead to a more successful Autonomous Package Delivery Robot project and a stronger, more cohesive team.

6.2 Project Artifact Summary with Links

2020-2021 MPDR GitHub

2021-2022 APDR GitHub

2022-2023 APDR GitHub

6.3 Presentation Materials

Project Showcase site link: https://eecs.engineering.oregonstate.edu/project-showcase/projects/?id=gqLh2ZB1mvPlecxF

https://eecs.engineering.oregonstate.edu/capstone/ece/student/projectinfo.php

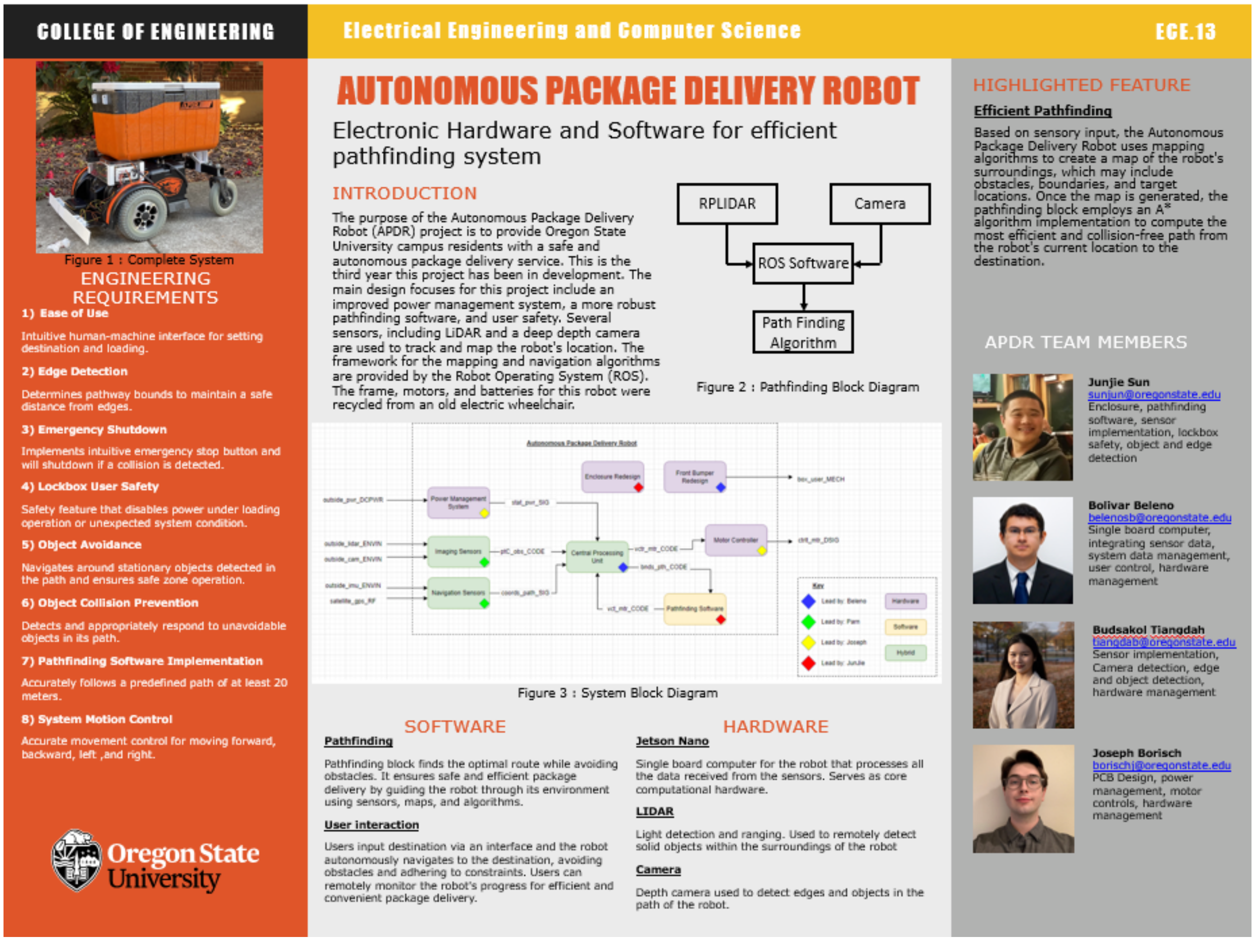

Figure 28: Project Poster

6.4 References and File Links

6.4.1 References

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, 1968.

[2] PythonRobotics, "PythonRobotics: A* Algorithm," 2021. [Online]. Available: https://pythonrobotics.readthedocs.io/en/latest/modules/path_planning.html#a-star-algorithm.

[3] T. Foote, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," International Journal of Robotics Research, vol. 32, no. 2, pp. 162-186, 2013.

[4] Intel RealSense, "Intel RealSense Depth Cameras," [Online]. Available: https://www.intelrealsense.com/depth-camera/.

[5] P. Beeson, N. K. Jong, and B. Kuipers, "Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph," in IEEE International Conference on Robotics and Automation (ICRA), pp. 4373-4379, 2005.

[6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets: Open-Source Library and Experimental Protocol," Autonomous Robots, vol. 34, no. 3, pp. 133-148, 2015.

[7] Slamtec, "RPLIDAR: A High-Performance 360-degree LIDAR Sensor," [Online]. Available: https://www.slamtec.com/en/Lidar.

[8] ROS Point Cloud Library (PCL), "ROS Point Cloud Library (PCL) Documentation," [Online]. Available: http://pointclouds.org/documentation/tutorials/.

[9] K. Smith, Printed Circuit Board Designer's Reference: Basics, Prentice Hall, 2019.

[10] Fedevel Academy, "Online PCB Design Courses," [Online]. Available: https://academy.fedevel.com/.

[11] Altium Designer, "PCB Design Tutorials," [Online]. Available: https://www.altium.com/solution/pcb-design-tutorials.

[12] QuinnRadich. (n.d.). *Guidelines for designing instructional Ui - Windows Apps*. Guidelines for designing instructional UI - Windows apps | Microsoft Learn. https://learn.microsoft.com/en-us/windows/apps/design/in-app-help/instructional-ui

[13] G. Hofstede, G. J. Hofstede, and M. Minkov, Cultures and Organizations: Software of the Mind, McGraw-Hill, 2010.

[14] J. T. Scarnati, "On Becoming a Team Player," Team Performance Management: An International Journal, vol. 7, no. 1/2, pp. 5-10, 2001.