Cleanbot 3000

Janine Dabu, Kevin Torres, Anthony Jerez, Inderjeet Gawra, Olumide Adewale, Cristian Robles, Mikel Mejia, Julien Mani, Jose Sanchez, Ronald Knight, and Orang Alitabar
Advisor: Professor James Flynn
California State University, Northridge

***Abstract*—This research paper discusses the development and testing process of the CleanBot 3000, an autonomous robot specifically designed to sanitize cleanrooms at NASA's Jet Propulsion Laboratory (JPL). The project's main objective is to develop an effective sanitization solution that adheres to JPL's requirements and reduces the risk of interplanetary bacterial contamination. For Spring 2023, the project was divided into three subgroups: Mapping & Navigation, Marker Localization/Systems, and Power & Mechanical Design. The Mapping & Navigation team has developed an autonomous navigation system using RPLidar to map the area that needs to be sanitized. On the other hand, the Marker Localization/Systems team has integrated sensors into the robot's new CPU (Raspberry Pi 4) to determine its location precisely. Finally, the Power & Mechanical Design team has designed and built a new power system, drive system, and robot chassis that meet JPL's requirements and address the issues of the previous prototype. CleanBot 3000 underwent rigorous testing to meet JPL's requirements. Its modular design makes it adaptable to various cleanroom environments, reducing contamination risks and providing a cost-effective maintenance solution, supporting JPL's mission of advancing space exploration by maintaining high cleanliness and safety standards.**

I. Introduction

The CleanBot 3000 project aims to develop an autonomous robot that can navigate JPL cleanrooms while sanitizing the floors with ultraviolet-C (UV-C) lights to eliminate harmful microorganisms. The risk of microbes originating on Earth's surface and being introduced into the cleanroom by foot traffic presents a significant threat to the ecology of Earth's upper atmosphere [1], especially since humans are the largest source of microbial contamination in the cleanroom [2]. By minimizing human contact and sterilizing the cleanroom floors, the CleanBot plays a crucial role in preventing the spread of harmful microbes.

In order to fulfill the main goal of the project. The CleanBot 3000 requires integrating several systems that meet JPL guidelines and requirements. These systems include control, navigation, power, drive, robot chassis, and sanitation systems. The robot must adhere to strict operational requirements, including maintaining a minimum distance of 1 meter from flight hardware. The control system is responsible for continuously monitoring various parameters, such as temperature, particle count, UV light, and battery levels while facilitating manual override shutdowns via wireless connection and providing diagnostic information to the operator. The navigation system must enable autonomous movement within the cleanroom while avoiding restricted areas and flight hardware. The power system should provide sufficient power to the CleanBot for travel and sanitation and includes a Lithium Ion battery equipped with short-circuit and overcurrent protection. The drive system must feature non-marking wheels, brushless, oil-free motors, and constraints to prevent tipping. The robot chassis must meet specific physical criteria, such as being under 30 inches in total height, 80% of the mass in the lower 20% of its height, and constructed out of stainless steel casing. Finally, the sanitation system must employ UVC LED modules that will effectively sanitize the floor while incorporating UV-C light shields to prevent the degradation of the robot's wheels over time.

To meet these requirements, the CleanBot 3000 team was organized into four subgroups: Mapping Navigation, Marker Localization/Systems, Power & Mechanical, and Sanitation. Each subgroup had a specific responsibility in designing and developing the robot: The Mapping & Navigation subgroup developed the mapping and navigation system to allow for autonomous movement, the Marker Localization/Systems subgroup worked on the control system, the Power & Mechanical Design subgroup developed the power, drive, and robot chassis systems, and the Sanitation subgroup was not active during the semester. By dividing the various systems among the subgroups, the team could streamline the development process and create a prototype that meets JPL requirements. The resulting robot can be further developed and improved to become a fully autonomous and effective cleanroom sanitizing robot eventually.

II. Mapping & Navigation

A. Previous Works

For the Fall 2022 semester, the mapping and navigation team decided to switch from the A-star (A*) algorithm to the Back-tracking spiral algorithm (BSA) to use as the CleanBot3000's coverage path planning (CPP) algorithm. This decision was mainly due to the strict requirements Jet Propulsion Laboratory (JPL) has given the CleanBot3000 project team. One requirement was that the robot must be able to traverse its environment autonomously. The other reasons for this change in the CPP algorithm are that the A* algorithm needs critical points to navigate an environment and an end goal. Since the CleanBot's environment changes daily, it makes the A* algorithm impractical because it does not traverse autonomously.

The BSA was chosen as a suitable replacement algorithm because of its ability to use sensor data to generate critical points depending on its environment. It has low computational costs, such as the A*, but the BSA has a back-tracing mechanism and finer grain group model. The back-tracing mechanism allows the robot to detect unvisited areas and store them as back-tracing cells (BS) to revisit after the spiral route is completed. The sensor data will be provided by the RPlidar running with the simultaneous localization and mapping (SLAM) algorithm. The SLAM algorithm is used to map an environment utilizing a Lidar-based tool. The RPlidar can rotate 360 degrees allowing for constant mapping of its environment.

Moreover, the BSA fell within the requirements given by JPL as it can traverse changing environments entirely autonomously. The mapping and navigation team achieved their goals for the semester by successfully simulating the BSA on ROS visualization (RVIZ) using open-source code. In addition, the team could map a teammate's room using SLAM on RVIZ.

B. *Plan of Action*

During the last semester, the mapping and navigation team used Robot Operating System (ROS) and RVIZ to simulate the BSA and SLAM algorithms. Completing this towards the end of the semester resulted in minimal knowledge of how to adjust or operate the code. Due to this, more research needed to be conducted before implementing the code onto the CleanBot prototype. For this semester, the team has decided to implement a Raspberry Pi to handle all sensor data before passing it on to Arduino. This allowed the Arduino to handle all motor controls and most of the computational cost of the Raspberry Pi. However, this meant ROS needed to be installed on the Raspberry Pi to run the BSA and SLAM algorithms.

The mapping and navigation team determined that for the Spring 2023 semester, it would work towards implementing the BSA and SLAM algorithms onto the Raspberry Pi and testing them on the prototype of the CleanBot. To achieve this, the team decided to research implementing ROS on the Raspberry Pi and the open-source code used by last semester. Parsing the open-source code will allow for modification of the code. If the BSA can be finely tuned and implemented with the SLAM onto the Pi, the prototype can be tested in different environments to see if the code will work.

C. *Research*

1) Full Coverage Path Planner: Last semester, the open source code utilized was a Full Coverage Path Planner (FCPP) implementing the BSA that Cesar Lopez and Tim Clephas developed [5]. The FCPP has two dependencies: ROS and the Move Base Flex (MBF). ROS provides device drivers, package management, visualizers, and other tools to create robot applications. The MBF package exposes action servers for recovering, planning, and controlling. In addition, it can produce detailed information about the plugin's feedback and the current state. Using the data, it can calculate navigation strategies that are intelligent and flexible [3].

In addition, simulating the FCPP on ROS requires tracking PID and mobile robot simulator packages. The tracking PID uses a tuneable PID control loop to follow a trajectory accurately with a tuneable velocity. The PID has three loops: angular, lateral, and longitudinal. Furthermore, it has two tracking options that depend on the path taken. One utilizes a carrot of arbitrary length in front of the robot to calculate velocity commands. This is based on the longitudinal and lateral error between the global and control points. Fig (1) shows how the Global point (GP), which is the projection of the radius of the robot including the wheel, follows the Control point (CP) or end of the carrot of length chosen. The other option requires a smooth path which allows the controller to track the path with the base link directly [6].

The mobile robot simulator is a robot simulator for ROS that includes two nodes. One used to simulate a mobile base and another used for a laser scanner. The first is called the mobile robot simulator node; it subscribes to the velocity commands received to update the robot's odometry. The other node is the laser scanner simulator node that performs raytracing on the occupancy grid map [7]. The FCPP must run simultaneously with the tracking PID and mobile robot simulator, but the MBF package must be installed to operate correctly.

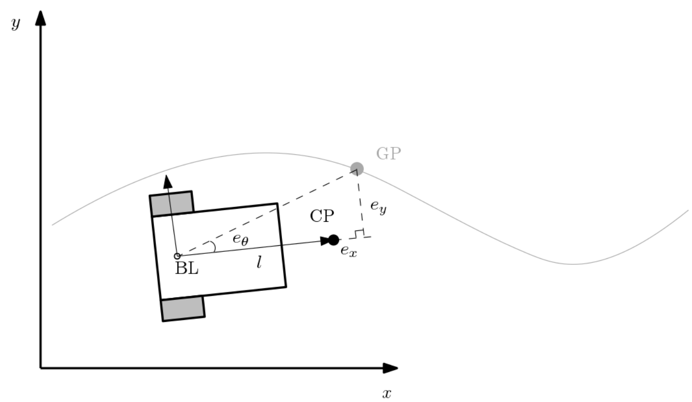


Fig. 1. The image illustrates the relationship between the tracking PID parameters such as Global point and the Control point [6].

2) Simultaneous Localization and Mapping: A crucial aspect of robot navigation is the ability to navigate autonomously within an unknown environment. To accomplish this, the team decided in the previous semester to use the Back-Tracking Spiral Algorithm (BSA). Implementing the BSA is done through an open-source package called Full Coverage Path Planner which acts as a plugin to the Move Base package in ROS meant to control a mobile robot's base [5]. The BSA implementation through the FCPP subscribes to a ROS topic named *'/tf'*, which supplies it with the robot's dynamic transformations that keep track of its movement in its environment. For a real-world implementation outside of the simulators, we need to provide the FCPP with a 2D map and the dynamic transformation data of the robot within that Map.

With this in mind, the team looked into SLAM algorithms. Amongst the many implementations of SLAM algorithms in ROS, the team determined that HectorSLAM was the most adequate in our situation due to its ability to be used without odometry and its ability to be used on platforms that perform roll and pitch motions [8]. Although it is unlikely for the robot to do any rolling maneuvers, one of the requirements set by JPL was for the robot not to tip while traveling over obstacles such as wire/cable covers, so it is likely the CleanBot will encounter pitch-changing obstacles. Furthermore, for creating the 2D Map and pose estimates within that Map HectorSLAM leverages the fast update rate of modern Light Detection and Ranging (LIDAR) systems like the RPLidarA1 provided by SLAMTEC [11]. The RPLidar A1 works on a laser triangulation ranging principle that uses high-speed vision acquisition and processing hardware created by SLAMTEC. The data coming from the RPLidar A1 is published on a topic that HectorSLAM uses to generate the 2D Map.

3) Robot Operating System on Raspberry Pi: So far, the CleanBot 3000 has had all its navigation algorithms running on simulators provided by ROS, such as RVIZ and GAZEBO. With the semester's goal of running the navigation algorithms on the physical robot, we determined that using a Raspberry Pi solely was more practical than the previous semester's goal of using the Zybo Z7-10 due to the Pi's low cost and high performance in robotic applications similar to our own. The team must select an operating system to run ROS on the Raspberry Pi. Because of the JPL requirement that the CleanBot must have a wireless connection, it was determined that the best candidate for an operating system was an Ubuntu Server with fewer programs installed because it needed a typical GUI, saving computational resources and meeting the wireless requirement. To communicate with the Raspberry Pi with the Ubuntu Server operating system installed, there must be a wireless connection through Secure Shell (SSH), and all actions must be done through a command line interface.

D. Implementation

1) Full Coverage Path Planner: After parsing through the FCPP, MBF, tracking PID, and the mobile robot simulator documentation, the mapping and navigation team installed all packages and dependencies needed to run the FCPP on RVIZ. When attempting to run the FCPP in conjunction with all the other codes, an error message was shown in the terminal stating that neither the *'mbf_costmap_nav* node nor the *'move_base_legacy_relay.py'* could not be launched. In addition, RVIZ displayed the mobile robot simulator and the tracking PID outside the given environment with a warning message stating that 'no map received.' The team decided to ensure all the dependencies and packages needed were installed.

When reviewing each dependency and package separately, it was discovered that MBF needed all the packages. However, when attempting to use the catkin make function after cloning the MBF package onto the catkin workspace, another error message appeared stating that it could not find the package configuration file provided by *'tf2_sensor_msgs'*. Due to the inability to install the MBF package properly, the FCPP could not be simulated on RVIZ this semester. Currently, the mapping and navigation team is looking into how to resolve the error encountered when attempting to install the MBF package onto the Catkin workspace. After cross-referencing the file left by the previous semester, resolving this issue allows the team to run the FCPP successfully on RVIZ.

2) Simultaneous Localization and Mapping: To implement the SLAM algorithm that would provide 2D maps for the FCPP when the robot is autonomously navigating, the first step was to download the SDK provided by SLAMTEC for the RPLidar A1. This allowed for the installation of the drivers to connect the sensor to our computer. Once this was done, the ROS package from Github meant for the RPLidar had to downloaded. This was done by going into the ROS catkin workspace on the computer and cloning the GitHub repository. The package for RPLidar published data from the sensor to a topic named */scan*.

HectorSLAM next had to be installed by cloning its GitHub repository and then making and sourcing the environment. Once this was done, HectorSLAM was ready to be launched with the RPLidar concurrently. While both were running, the RPLidar package was published to the */scan* topic while HectorSLAM was subscribing and creating a map, all while locating the RPLidar in the environment. Below is a visual of the program running. Fig. 2 shows that the gray box is the environment map, and the green line is the path the RPLidar took within the map, Fig. 3 shows the program running during the mapping operation. The map generation was successful and ready to be implemented with the FCPP for the autonomous navigation of the CleanBot.

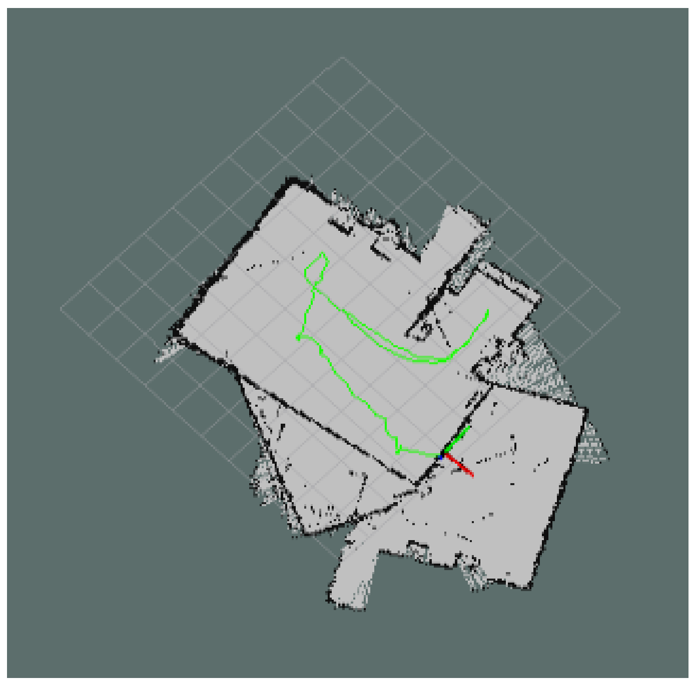


Fig. 2. Generated Map from Fig. 3

3) Robot Operating System on Raspberry Pi: Moreover, for the implementation of ROS on the Raspberry Pi, the first step was to use the Raspberry Pi Imager application. The Imager allows you to flash an operating system onto an SD card that will later be used as the hard drive on the Raspberry Pi. For this application, the team used a server version of Ubuntu which was the Ubuntu Server 20.04.2 LTS. After selecting the operating system in the Imager, it was flashed onto the SD card that would later serve as the hard drive on the Raspberry Pi.

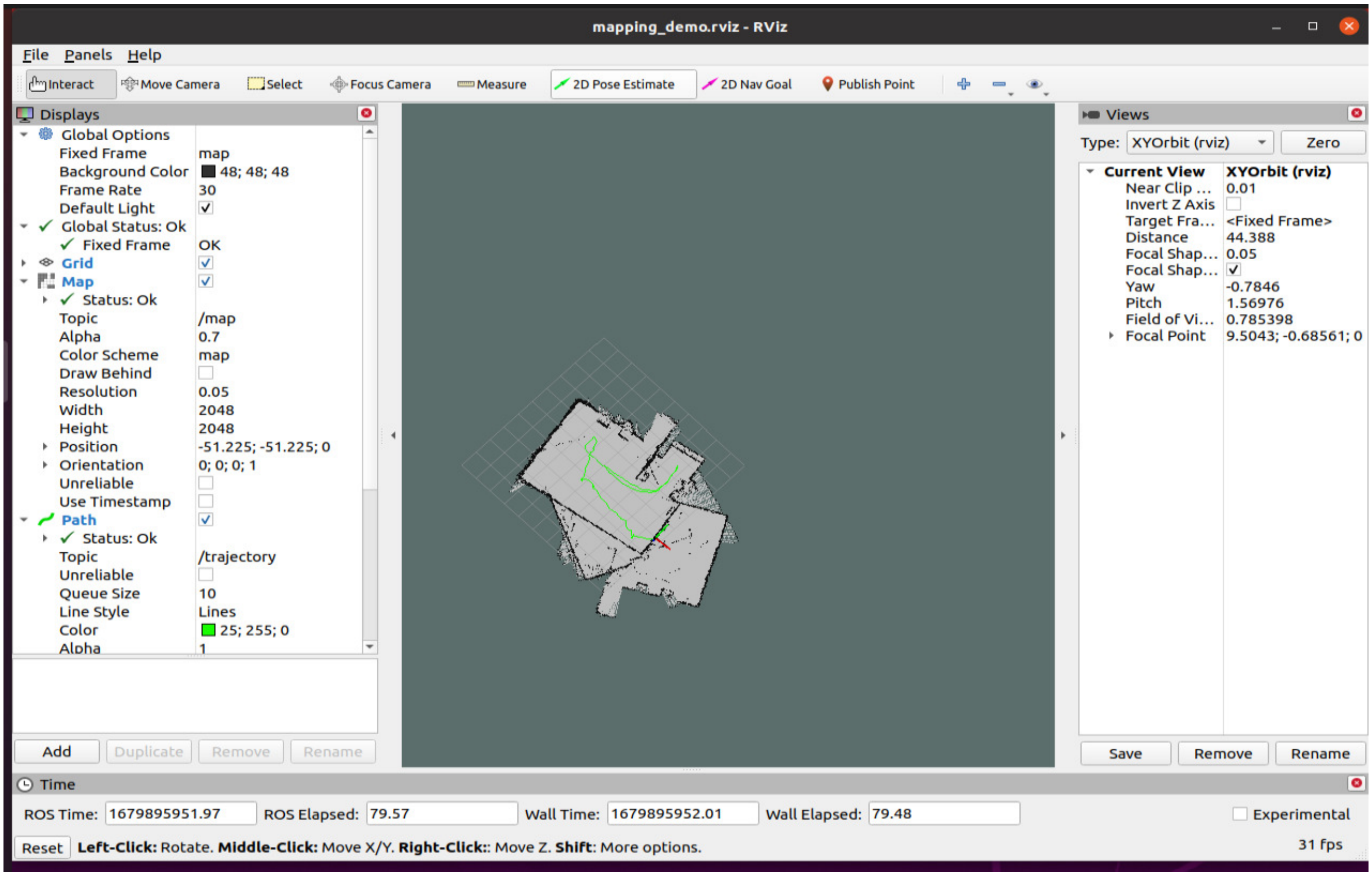


Fig. 3. The mapping of a teammate's room using the RPlidar and SLAM algorithm.

Before putting the newly erupted SD card into the Raspberry Pi, it was necessary to make changes to a 'network-config' file. The changes included the local WiFi name and password and an address for a static IP address for us to connect to the Raspberry Pi through SSH later on. Once the 'network-config' file was edited, it was time to plug the SD card into the Raspberry Pi.

After giving it time to boot up, the Raspberry Pi was connected through SSH using the information in the earlier network-config file. Once the connection was secure, it was time to install ROS. The version of ROS that is best with the Ubuntu 20.04.2 LTS is ROS Noetic which comes without all the GUI elements and takes less computational power [10]. To install this, the team followed all of the instructions provided by the ROS Noetic installation guide. Once the installation was complete, we created a workspace folder named Catkin, where all navigation algorithms will be held later. The command *'catkin_make.'* was used to build the workspace. Finally, the team successfully ran ROS after using the command *'roscore'*. The CleanBot is now ready for the navigation algorithms to be put onto the Raspberry Pi and begin testing on the physical robot.

III. Marker Localization / Systems

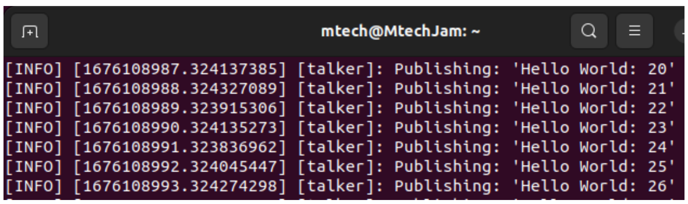
A. Previous Works

In the Fall 2022 semester, the Marker Localization and Systems team focused on progressing the implementation of Digilent's Zybo Z7-10 SoC [12] as the central processing unit (CPU) of the CleanBot. To concurrently operate the Pmod TMP3 temperature sensor, the team used Raspberry Pi Camera v2, SLAM/BSA navigation algorithms, HR-SR04 ultrasonic sensor, and RPLiDAR 360 A1 with the Zybo Z7-10, and the Vivado hardware development environment was used to configure the peripherals required to communicate to the different external devices. After the generation of the hardware file from Vivado, Vitis was used to develop the project's software further. Using this development flow, the team focused on implementing the Raspberry Pi Camera using Adam Taylor's project, "Building a Camera/Imager Test Platform [13]." Much research was done at this stage since multiple IPs were not modifiable or compatible with the newer Vitis IDE. Ultimately, the team implemented a final design that connected the Raspberry Pi Cam via MIPI to the Zybo Z7-10 while leaving a template that will guarantee the build will work on various operating systems.

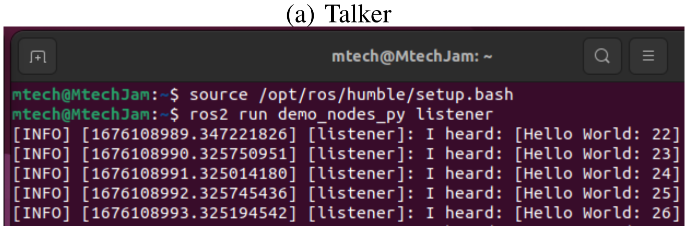
B. Plan of Action

During the Spring semester of 2023, the Systems team decided to switch to the Zyo Z7-10 for the Raspberry Pi 4 Model-B [14]. This decision does contradict last semester's work, as it was decided to move forward with the Zybo Zy-10 without the Raspberry Pi. Our decision to switch to the Raspberry Pi provided a more straightforward solution that better aligned with our technical expertise. In addition, the Raspberry Pi could efficiently do what the Zybo was meant to, and with previous semesters' research, it was determined that all necessary project features could be implemented on the Raspberry Pi without compromising functionality or performance.

We decided to reformat the Pi with Ubuntu [15], specifically the server side, because it is more compatible with ROS, as it's primarily developed on it. It also has an extensive repository of packages and hardware support that could be used if we encounter any errors.



(a) Talker



(b) Listener

Fig. 4. Communication between Machines

While setting up the Raspberry Pi, the team conducted further research into setting up a server that would enable external communication with the robot. To test this setup, the team established communication between two machines, one sending a "Hello World" message and the other receiving it, as shown in Fig 4(a) talker and (b)listener. This successful test paves the way for future communication with multiple CleanBots at once and the ability to receive real-time data from the robot.

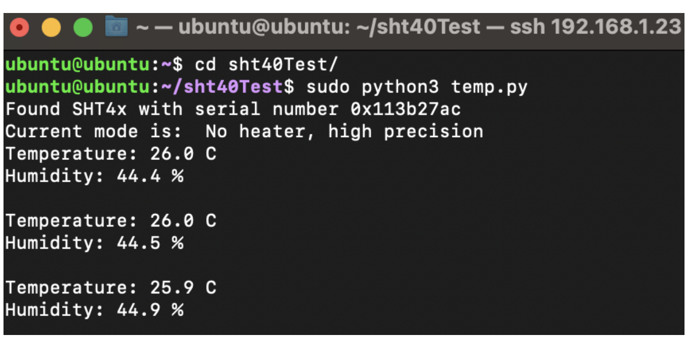


Fig. 5. Implementation of SHT40 Temperature Sensor [16]

Due to our decision to switch back to the Raspberry Pi, we also changed the temperature sensor to the SHT40 [16]. We successfully implemented the new temperature sensor onto the PI in Python using Adafruit's CircuitPython SHT4X libraries [17], as seen in Figures 5 and 6. In the future, we plan to reimplement the sensor in C/C++ using Adafruit's Arduino libraries and functions when we conduct further research. To ensure that the sensor was working, a unit test was conducted.

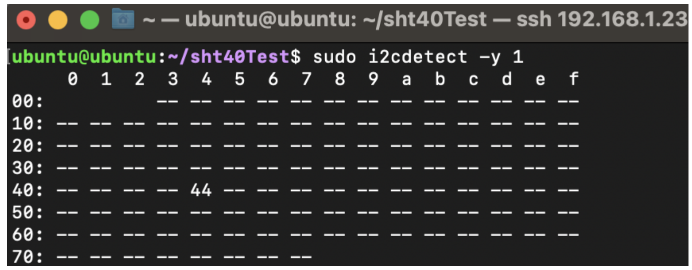


Fig. 6. I2C Bus Scan of SHT40 Temperature Sensor

With the temperature sensor working, the team plans to explore using ROS with multiple machines. This would allow us to take the load off the PI and, in the near future, communicate with multiple CleanBots at once. In the future, we also plan to work with the BSA algorithm to develop an autonomous driving capability based on the map generated by the navigation algorithm.

C. *Research*

This semester, the Systems team looked to use the Raspberry Pi with an external computer. With this setup, some of the load would be lifted off the Raspberry Pi since the computational stress from running the mapping algorithms can now be run on the external computer rather than the Pi, which still has other tasks. In addition, having an external computer that takes over some of the computations has other advantages. For example, in the future, if multiple CleanBot were created, they could be managed by this central computer which would handle analyses and display important information on the CleanBots.

The Systems team planned to connect the external computer and the Raspberry Pi via the Internet. After consulting with the team at JPL and receiving the green light for this idea, the team has decided to move forward. One of the first steps to moving forward with this idea was to figure out exactly how the Raspberry Pi and the external computer would connect and communicate. Since the connection would be via the Internet, the team considered using an Internet protocol called User Datagram Protocol (UDP).

UDP is a communication protocol used all over the Internet and helps send time-sensitive information [18-19]. UDP uses a client-server model, which means that the Raspberry Pi would be set up as a server while the external computer would act as the client and connect to the server through the UDP connection. UDP utilizes both the two devices' IP addresses and port numbers to send information.

To test if a connection could be established using UDP, two sets of code were written and run on both the external computer and the Pi using Python. The server side of the code was run on the Raspberry Pi, while the external computer ran the client side. After running the two sets of code, a connection was successfully established between the Pi and the external computer, and the team could send simple lines of text between the two devices. Considering that most information being transmitted will be information regarding the navigation algorithm, UDP should be more than capable of transmitting the amount of data produced. Also, since UDP is well used for scenarios requiring time-sensitive information, the team should be able to send and receive times, which makes UDP a better choice than other internet protocols.

D. Raspberry Pi Cam Implementation

During previous semesters, the systems team began to implement ArUco markers which would be used as a reference to find the location of a given object for the specific marker. Having moved away from the Zybo Z7-10, the systems team needed to implement a camera and chose the Raspberry Pi High-Quality HQ camera. The camera offers a 12.3-megapixel resolution and is simple to integrate with the Raspberry Pi [20]. The team also uses a 6mm 3MP wide-angle lens with the camera [21].

The Raspberry Pi camera chosen allows simple integration with Raspberry Pi 4. The camera connects directly to the Pi via the Pi's camera port. Once the camera is connected and the Pi is powered, the team can open a preview window using the terminal by running the command:

raspistill -t 0

The team can adjust the camera using the preview window to get the best image possible. Once the camera is configured, the team can take pictures using the following command:

raspistill -o image.jpg

IV. Power & Mechanical Design

The Power & Mechanical Design is a critical component of the CleanBot project that focuses on developing the key components that drive the autonomous sanitizing robot. These components include the Drive System, Power System, and Robot Chassis. In addition, the Power & Mechanical Design team is responsible for integrating these components with other essential systems, such as the Control, Navigation, and Sanitation systems. The team's work plays a crucial role in ensuring the seamless operation of the robot, allowing it to sanitize cleanrooms effectively. This section provides a detailed account of the design, development, and integration of the Power & Mechanical systems, highlighting the various challenges faced and the innovative solutions devised to overcome them.

A. Previous Works

During Fall 2022, the Power & Mechanical team extended their research and transitioned to the prototyping phase to finalize CleanBot's components. The team successfully built a working prototype within a six-day deadline and with no funding, utilizing RP-LIDAR for mapping, an ultrasonic distance sensor for collision mitigation, and a UV-A LED panel for simulating UVC sanitation. An RC remote and Arduino code controlled the robot's traversal to a halt within 45 cm of any obstacle [22].

Regarding the power system, the team opted for the Varta EasyBlade24 battery [23], providing a 1500Wh capacity, extended run time, and safety features like a short circuit, over-discharge, and overcharge protection. The battery and charging station was updated to a single battery, providing more fail-safe features from the previous design of eight batteries in parallel.

For the drive system, the team implemented rubber wheels, brushed DC motors, a dual H-bridge motor controller, and an Arduino Mega 2560 as the microcontroller sending PWM signals. The motor controller modulated the speed of the motors using PWM, and the Arduino was programmed to receive signals through the ia6b receiver, enabling remote control of the robot through the Flysky FS i6x remote control.

The robot chassis design evolved from the initial plan of a 3D-printed design to a modular aluminum chassis that satisfies JPL requirements for 80% mass in the lower 20% of its height and a total height under 30 inches, including wheels [22].

The SolidWorks team made minor design adjustments for various component changes, including motors and batteries, the Zybo board, RP-LiDAR, and Ultrasonic Distance Sensor. The team maintained the three-wheel design with two motorized wheels at the rear to enable sharp turns, clean more areas, and achieve more efficient traversal [22].

Finally, the team demonstrated the prototype at JPL facilities to gain insight into improving the project and overcoming any obstacles. During the demonstration, the robot lacked traction and slid down from an inclined platform.

B. Plan of Action

The Power & Mechanical team of the Spring 2023 semester successfully designed and developed a new CleanBot prototype that meets JPL requirements. The team initiated the project after identifying several issues with the previous model, such as non-compliance with JPL requirements, substandard traction, and a lack of reliable documentation for many components. The new prototype is modular and meets JPL requirements, serving as a foundation for future semesters to build upon.

To begin the project, the team performed rigorous testing and troubleshooting on the previous prototype and researched BLDC motors while focusing on closed-loop control techniques. The team identified the essential parameters, including torque and speed, required for selecting BLDC motors and chose a compatible driver matching the selected motors' voltage and current draw. The team opted for a driver with integrated hall sensors that could provide feedback for the closed-loop control. Additionally, the team procured all components for the robot that meet JPL requirements.

The team constructed the robot's enclosed chassis using aluminum supports and acrylic material to protect it. The team then integrated the BLDC motors, power distribution system, wheels, and drive system into the chassis, finalizing the Spring 2023 CleanBot prototype.

To meet JPL requirements, the team upgraded the power system with Li-Ion batteries [24] and a BMS (Battery Management System) and developed a new power distribution scheme incorporating multiple protection layers for safety and an efficient design.

The Power & Mechanical team integrated the microcontroller (Arduino Mega 2560) and motor drivers, tested the robot's movement, and made necessary adjustments. The team then subjected the robot to stress tests by loading it with a 20lbs payload and conducted a speed test to verify its average speed. Finally, the team added the final touches and improvements to the prototype and prepared the final report and presentation.

C. Research

As previously stated, the focus of the Power & Mechanical team of Spring 2023 lay around revising the overall drive system of the previous CleanBot version to adhere to JPL requirements and improve the comprehensive drive system's performance. To initiate this process, the team followed a systematic troubleshooting process to address the previous prototype's power and drive system issues.

First, the team reviewed the parts and components' documentation to verify their specifications and operational set-up parameters. Next, they checked the voltage and current source and measured each device's input to ensure the correct ratings were supplied.

The team then visually inspected the system to identify physical damages or defects on all parts and components. They also verified the installation, connection, and wiring to ensure everything was in place and not shorted.

To identify faulty parts, the team proceeded with a process of elimination by testing the drive system. First, the team conducted a torque and speed test by directly powering the motors with a variable power supply. The test showed that the motors had high speed but low torque, which verified that the prototype could not navigate and drive over an incline and had poor traction during the differential drive. In addition, further investigation of the motor hardware revealed they were Brushed DC motors, not the required Brushless DC (BLDC) motors. Therefore, the team concluded that the motors needed to be replaced.

Since the H-Bridge motor controller controlled and powered the motors of the drive system, the team measured the input and output parameters of the controller. First, they observed the input and output PWM signal and voltage into the motors and obtained two results: (1) the controller received 12V but output less than 1V, and (2) the input and output PWM signals had a frequency below 1kHz, while the controller manual stated a max frequency of 20kHz.

As the team needed new and compatible controllers/drivers for the different type of motors, the team decided not to pursue changing the code to fix the frequency parameter of the H-Bridge controller.

The team worked closely with Prof. Flynn through this testing process to troubleshoot and test the previous prototype. The tests concluded that the traction issue was due to the motor's performance. Therefore, the team designed a new drive system that complied with JPL's requirements and used compatible motors. The new drive system required the replacement of all parts and components to ensure compatibility with the motors and efficient and safe performance of the overall drive operation of the CleanBot. Extensive research was conducted on the power system design of the CleanBot project, covering several crucial aspects. First, the study focused on evaluating the robot's power requirements and identifying the most appropriate components for the power system, including the operation and protection of lithium-ion batteries. The team also researched the battery management system's functionality, effective implementation, and the significance of overcurrent and short circuit protection. This research played a crucial role in ensuring the overall effectiveness and safety of the power system, enabling CleanBot to power its electrical components with maximum efficiency and reliability.

TABLE I
MOTOR SIZING INPUT PARAMETERS

Parameters	Values
Total Mass	25 lbs
Number of drive motors	2
Radius of drive wheel	3 in
Robot Velocity	0.05 m/s
Maximum incline	30 deg
Supply voltage	24 V
Desired acceleration	0.2 m/s^2
Desired operating time	5 hrs
Total efficiency	100%

TABLE II
MOTOR SIZING OUTPUT PARAMETERS

Parameters	Values
Angular Velocity	6.2691 RPM 0.65617 rad/s
Full-load Torque	2.20 Nm 22.433 kgf-cm 311.54 ozf-in

After testing and troubleshooting the previous prototype, the team discovered that the traction issue was due to a hardware limitation. Specifically, the brushed motors used in the last prototype could not carry a 25 lbs payload at a slow speed. This meant the team needed to find a new solution to allow the robot to carry a heavy load while maintaining a slow speed.

The previous prototype's brushed motors were designed for high-speed and low-torque applications, which was unsuitable for CleanBot's requirements. On the other hand, the new BLDC motors needed low speed and high torque specifications to meet the robot's needs. This meant that the new motors could carry a heavy payload while maintaining a slower speed, which was necessary for the robot to complete its cleaning tasks effectively.

To determine the specific torque requirements for the new motors, the team used the motor sizing tool [25] to calculate the necessary torque for the robot to move up a 30-degree incline at an average speed of 5 cm/s while carrying a 25 lbs payload. The tool's calculations showed that the robot would require motors with low speed and high torque specifications to accomplish this task. The input and output parameters of the Motor sizing tool are in Tables 1 and 2.

By selecting motors with 5-10 RPM and 2-5 Nm = 283-708 ozf-in specifications, low speed, and high torque, the team can ensure that the robot will have the necessary power to accomplish its cleaning tasks effectively. The new BLDC motors will provide the required torque for the robot to move slower while carrying a heavy payload, enabling it to clean large areas thoroughly and efficiently.

In summary, conducting preliminary research on the desired specifications of the new components is crucial to ensure the team procures suitable components for the new prototype. Understanding the importance of these specifications and functions will ensure that the parts meet the requirements, perform efficiently under load conditions, and avoid wasting limited resources. Furthermore, by procuring the right components, future prototype iterations can proceed smoothly without the need for replacements or new research.

D. Power Systems Design

The research and development of the power system design of the CleanBot project are critical to its performance and effectiveness in sanitizing cleanrooms. The new prototype power system, designed and developed by the Power & Mechanical team of the Spring 2023 semester, adheres to the stringent requirements set by JPL.

The Lithium Ion battery installed in CleanBot 3000 has safety features such as overcurrent and short circuit protection, guaranteeing optimal safety and dependability [24]. The power system design for the prototype comprises crucial elements such as a battery and charger system, a battery management system, and a power distribution arrangement. These elements operate collaboratively to supply a sturdy and reliable power source for CleanBot's diverse functions, such as sanitization and mobility. The meticulous design and execution of these components are vital to the robot's performance, and this section will furnish an elaborate summary of their design and evolution.

1) Batteries/Charging: The team has upgraded the power system with Li-Ion batteries (Fig. 7) [24] and a BMS to meet JPL requirements. Despite this upgrade, the Varta Easy Blade 24V (Fig. 8) remains the preferred battery choice due to its adherence to safety standards and reliability requirements. [23] The Varta battery is widely used in the industry for AGVs and features an integrated cooling system for active temperature regulation, ensuring a longer lifetime by preventing hot spots. Additionally, the battery has integrated BMS and charger control with a dynamic Master-Slave-BMS concept via CAN communication [23]. With a capacity of 25.9V and 58Ah, the battery provides 1502Wh, enabling the robot to run longer than the previous battery configuration. The battery has short circuit protection, over-discharge current protection, overcharge current protection, as well as IEC62133-2:2017 and UN28.3 certifications. Furthermore, it has an active cooling system and does not require external battery management [23] .

However, it should be noted that the battery is still pending approval by JPL.

Fig. 7. Lithium Ion Battery with 2Ah Charger [24]

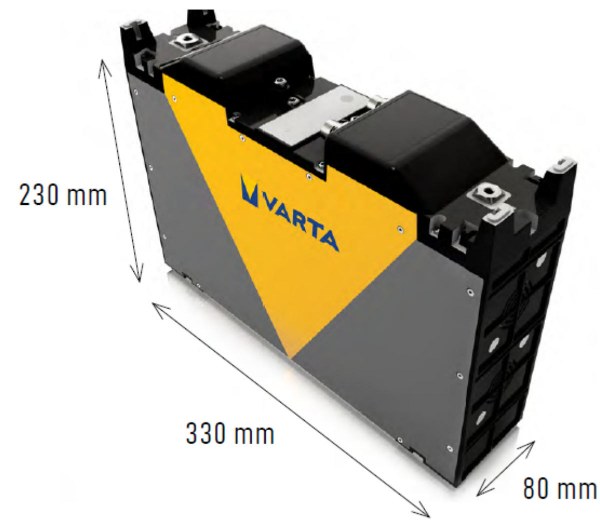


Fig. 8. Varta Easy Blade 24V [23]

2) BMS - Protection: The Battery Management System (BMS) is crucial to CleanBot's power system. It ensures that the lithium-ion battery, the robot's primary power source, works safely and efficiently. The BMS acts like the battery pack's brain, guaranteeing the battery charges and discharges correctly and doesn't overcharge or over-discharge. It also protects against overcurrent and short-circuits, keeping the power system reliable and safe [26-27]. With a well-designed and adequately implemented BMS, the power system could succeed or get damaged, causing safety hazards or performance issues.

An essential function of the BMS is cell protection, which prevents overcharging, over-discharging, over-temperature, short-circuits, and other dangerous situations that can damage the battery or compromise its safety [28]. Overcharging can cause overheating, and even explosion or flame, while over-discharging will permanently reduce the battery capacity or accelerate aging [26]. Furthermore, the BMS ensures that

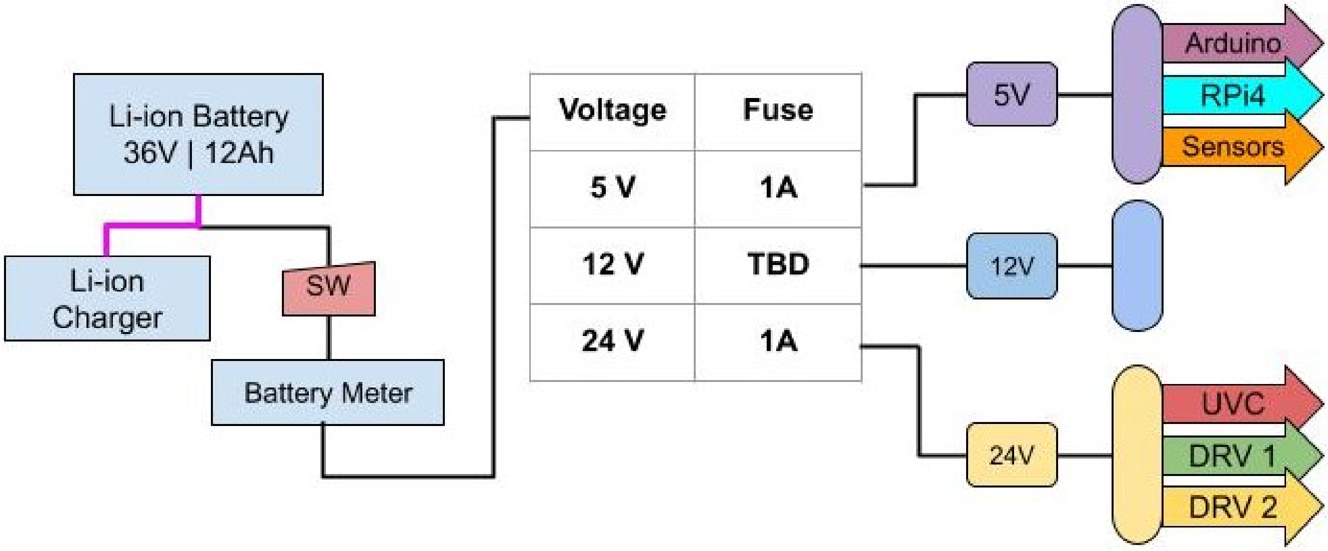


Fig. 9. SP23 Power Distribution Scheme.

the battery works within the defined specification limits and delivers the maximum admissible current, voltage, or power from or to the battery, depending on the working conditions [27]. In addition, the BMS provides an accurate battery state of charge gauge during the entire operating range of the battery, which is a crucial function for energy management [28].

The BMS is critical for the safe and efficient operation of CleanBot's power system. It monitors and controls the battery's charging and discharging, provides overcurrent and short circuit protection, and ensures the battery works within defined limits. As a result, the final build battery, VARTA Easyblade 24 [23], and the current prototype lithium-ion battery [24] are equipped with BMS guaranteeing security, increasing efficiency, and enhancing operational performance.

As part of future work, the team plans to implement capacity and battery monitors to display real-time values of the battery's health and charge. Additionally, telemetry will be added to remotely send real-time data to the robot's external or operator computer [29]. The battery monitor will monitor the battery voltage during operation and trigger an alarm when the voltage drops to a dangerous level [30], indicating the need for the CleanBot to return to the docking station for battery charging. These additions will further enhance the performance, efficiency, and safety of CleanBot's power system design in the future.

3) Power Distribution Scheme: The power distribution system is critical to a successful and reliable robot design. It aims to ensure that each electronic component receives the necessary power to function correctly while preventing voltage drops and reducing the risk of component failure due to insufficient power.

For the CleanBot 3000, a new power distribution scheme is currently being developed, which features multiple layers of protection and an efficient design, see Fig. 9 The power will come from the battery, pass through a meter and a fuse box, and then be distributed to various components through three buck/boost converters, each corresponding to a different voltage level (5V, 12V, and 24V).

The 5V level will power the Arduino, Raspberry Pi, sensors, and RPLidar, while the 24V level will be required to power the BLDC motors through the BLDC motor drivers. Additionally, the current prototype of the CleanBot 3000 includes an UltraViolet-C LED module (UVC) [31], which needs a supply voltage of 24V. Adding a dedicated 12V voltage level also allows for the future addition or replacement of components that require this voltage level.

As determined through testing, all drive system components will require a 1-amp fuse to ensure overcurrent protection. The buck/boost converters will switch between buck and boost modes depending on how low or high the input voltage is compared to the target output voltage [32]. They will be connected to a terminal block that will evenly distribute power to the rest of the components.

The team is designing the scheme to ensure effective and safe power distribution. The current design uses blade fuses, commonly known as Automobile-style fuses, and are easily replaceable when blown [32]. The team explores other fuses and relays to improve a power system's overcurrent, short circuit, and surge protection. Different types of fuses are available, such as fast-blow, ultrafast, and slow-blow fuses. Fast-blow fuses open quickly when the rated current is reached, while ultrafast fuses protect semiconductor devices that can only tolerate short-lived overcurrents. Conversely, slow-flow fuses can tolerate a transient overcurrent condition and are often used to protect electro-mechanical devices like motors[32]. The team will consider these options for future improvements to CleanBot 3000's power system.

4) Prototype Development: The current prototype of CleanBot 3000 used a variable power supply as the primary power source, providing 24V to the power distribution system. This was chosen due to the supply's short circuit and overcurrent protection, essential during testing. Even though the prototype battery has a built-in BMS with similar protection, the power supply is more suitable for safely establishing the optimal setup parameters and overcurrent protection (determined as 1 Amp). The primary testing objectives are to ensure that the motors move and to establish safe setup parameters for the power system to supply power to various components in the robot. Once these parameters are identified and verified, the battery can be the primary power source. This will be important for the upcoming tests, which will require stress testing in different environments, such as outdoors or indoors, with no electrical outlet. In addition, the prototype will be portable once the battery is integrated into the system. The battery's overcurrent and short-circuit protection will also be verified. However, if the BMS fails to provide sufficient overcurrent protection, the multiple layers of protection employed in the system will help prevent and isolate faults and damages.

E. Mechanical Design

The mechanical design of the CleanBot involves developing its drive system and robot chassis. However, during the Fall 2022 JPL demonstration, the robot exhibited poor traction and slid down an inclined ramp. Moreover, the robot's chassis did not meet JPL's prerequisites, which include an enclosed casing and a minimum ground clearance of 2 inches, as indicated in Fig. 10 [22]. To tackle these problems, the mechanical design team centered on three primary goals: the team aimed to troubleshoot the previous prototype, research and design a new drive system that could perform under specified and peak conditions, and construct and manufacture an enclosed casing for the new prototype that met JPL's requirements.

The new drive system underwent rigorous testing to ensure its functionality and performance, and the enclosed casing was manufactured to meet JPL's specifications.

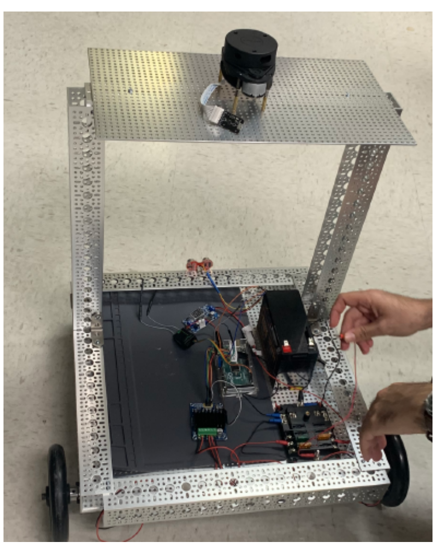


Fig. 10. Fall 2022 Prototype [22]

1) Drive System:

a) Wheels: To minimize the accumulation of dirt, bacteria, and dust during CleanBot's movement, the team opted for polyurethane wheels [33-34] with numerous benefits, such as reduced floor marking and minimal bacteria transportation [35]. In addition, the team confirmed using polyurethane wheels as they can maintain cleanliness while CleanBot navigates through the JPL clean rooms, which requires wheels that do not leave marks or pollutants on the floor [22].

The fixed wheels chosen for CleanBot have two degrees of freedom and can move in both forward and backward directions, making them ideal for drive wheels in rear-wheel robots. Furthermore, with two independently controlled motors, the robot can effortlessly move in a 360-degree motion, making it easy to navigate through tight spaces [36].

To enhance stability in narrow and curved environments, the team implemented the differential drive method that includes two wheels on a common axis driven by separate motors and one undriven wheel, usually caster wheels, for stability [37]. As a result, the drive system prototype features a 3-wheeled configuration consisting of one caster wheel and two polyurethane wheels, as shown in Fig. 11 [33-34]. The back two wheels are motorized, and the front wheel serves as a caster idler for steering; this configuration provides efficient movement on smooth terrain [37]. Overall, CleanBot's wheels research and development team thoughtfully considered various materials and configurations for the wheels to ensure optimal performance in clean environments.

Fig. 11. 2in Caster & 5in Polyurethane Wheels [33-34]

b) Motors and Motor Drivers: The CleanBot team's main goal was to address the poor traction and sliding issues experienced during the Fall 2022 JPL demonstration by redesigning the drive system. The team decided to research and develop a new drive system prototype using Brushless DC (BLDC) motors.

The team began the development of the new drive system by researching and calculating the required torque and specifications for BLDC motors. BLDC motors are a type of motor that converts electrical energy into rotational energy using electromagnetic fields. Unlike brushed motors, they do not contain brushes that contact the rotor, resulting in higher efficiency, lower maintenance, better thermal performance, and longer lifetime [38].

The team considered several specifications and functions to choose suitable BLDC motors for their system. The motors need low speed, and high torque, have integrated hall sensors for feedback control, have good documentation and customer support, a reasonable price, and a good life expectancy. After careful consideration, the team chose the BLWRPG092S-24V-4600-R264 Brushless Planetary Gear Motor from Anaheim Automation, as shown in Fig. 12. These motors were coupled with planetary gears, which reduced the RPM from 4600 to 17.4 RPM and increased the peak torque from 3.39 to 416.3 oz-in [39].

Fig. 12. BLWRPG092S-24V-4600-R264 Brushless Planetary Gear Motor [39]

BLDC motors are typically controlled using a Pulse-Width Modulation (PWM) signal and a feedback sensor such as a hall effect sensor. The PWM signal is generated from a source such as an Arduino and sent to a driver circuit to tell the motor to move. The sensor returns to the driver circuit and Arduino to indicate how much the motor has moved. The prototype used BLDC motor drivers/controllers to control the speed and direction of the motor.

For precise movement in restricted areas, closed-loop control is crucial in motor control. It allows the system to accurately track the motor's position within its rotational range, aiding navigation through obstacles and constrained spaces. For the CleanBot project, the team recognized the importance of choosing the proper motor drivers to accommodate inputs for the motors' hall sensors and enable closed-loop control. The hall sensors provide feedback to the motor driver, allowing it to adjust the motor's speed and position as necessary. This level of control enables even finer-tuned motor controls, making the CleanBot more efficient in navigating through JPL clean rooms.

After selecting the appropriate motors, the team searched for compatible motor controllers or drivers matching the motors' voltage and current ratings while accommodating inputs for the motors' hall sensors, which will be used for closed-loop control in the following semester. In the future, the CleanBot will use closed-loop control with hall sensors to precisely monitor the position of each wheel within the arch of rotation, enabling even finer-tuned motor controls that will aid in navigating through JPL clean rooms while avoiding obstacles and restricted areas.

Due to its various features and specifications, the team chose the BLD-120A Motor Driver [40-42] (as shown in Fig. 13) as the BLDC motor controller. The BLD-120A is a 30VDC 120W BLDC Motor Driver that can be applied to a host computer (PLC or SCM) for PWM or analog speed control. It also offers manual speed control mode with its potential or external potentiometer, PID speed loop and current loop control, start-stop control (EN), reversible control (F/R), and braking fast stop (BRK). Additionally, it offers overload protection settings with different power motor protection parameters through the linear potentiometer, high-speed torque output stability, speed and stability, low-speed high torque output, and various protections such as overcurrent, overvoltage, brown, stall, hall signal illegal, and temperature protection [40-42].

Fig. 13. BLD-120A Motor Driver [40]

The BLD-120A motor driver's various protections prevent motor damage caused by inadequate or excessive current or voltage, lock rotor detection, and incorrect hall sensor connections [40-42]. While the prototype did not utilize the hall sensor integration due to time and funding constraints, CleanBot's future iteration will benefit significantly from this feature, allowing for even more precise motor control. In addition, by taking advantage of the hall sensor integration, the CleanBot will be able to fine-tune its motor controls, making it easier to navigate JPL cleanrooms while avoiding obstacles and restricted areas. Despite the constraints, the team successfully designed a CleanBot that meets its goals with the necessary features and protections to accomplish its tasks.

The procurement and background research for the drive system selection took longer than expected due to the team's meticulous approach to understanding each component's function before proceeding to procure them. This was important for the team, as the project had no funding and needed to utilize its limited resources efficiently. Furthermore, the team ensured that the chosen parts would be used in the final iteration of the CleanBot, so future semesters would not have to change or replace them, allowing them to focus on fine-tuning the system.

Due to this reason, the team focused on testing and evaluating the new drive system for the rest of the semester. The team focused on conducting stress and performance test to verify the robot's ability to operate and perform. The observations and results can then be analyzed, and the team can identify weaknesses and make necessary adjustments to the prototype.

2) Robot Chassis:

a) Solidworks: The robot chassis design created in Solidworks during the previous semester remains the final design. The team used the dimensions from the 3D model to construct a prototype chassis; only the enclosed portion of the model was manufactured in Fig. 14 [22].

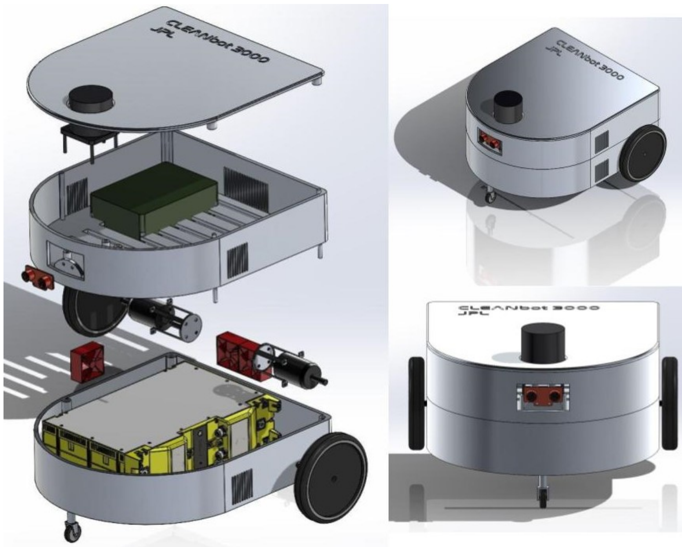


Fig. 14. CleanBot Solidworks Design [22]

Additionally, the Solidworks software was also employed to redesign the goBilda channel mount [43] as a motor mount, which was then manufactured using 3D printing, Fig. 15. Moving forward, the prototype chassis will be further developed to include additional features, such as vents across the chassis and a mounting hole for the RPLidar on top, as depicted in Fig. 14.

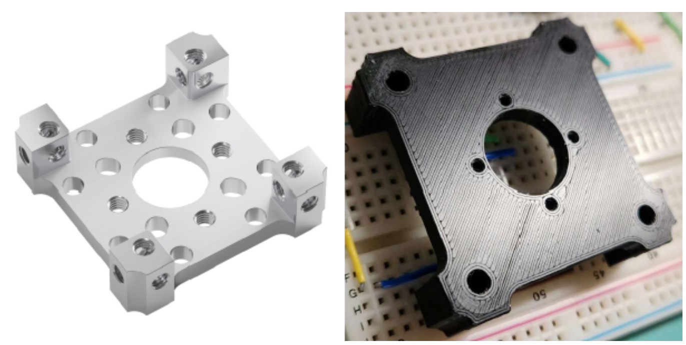


Fig. 15. goBilda Channel Mount [43] & 3D Printed Mount

b) Chassis Manufacturing: The mechanical design team's primary objective is to manufacture a robot chassis that addresses the shortcomings of the previous prototype, which did not meet JPL's requirements due to its open frame design, holes, and insufficient 2-inch ground clearance. To avoid impeding the drive system testing's progress, the Actobotics channels-based chassis [44], as shown in Fig. 16, which weighs 15 lbs, was used in the initial round of tests. This chassis is lightweight and narrower than the previous GoBilda channels-based prototype (Fig. 10) [45-46], which is broader and heavier.

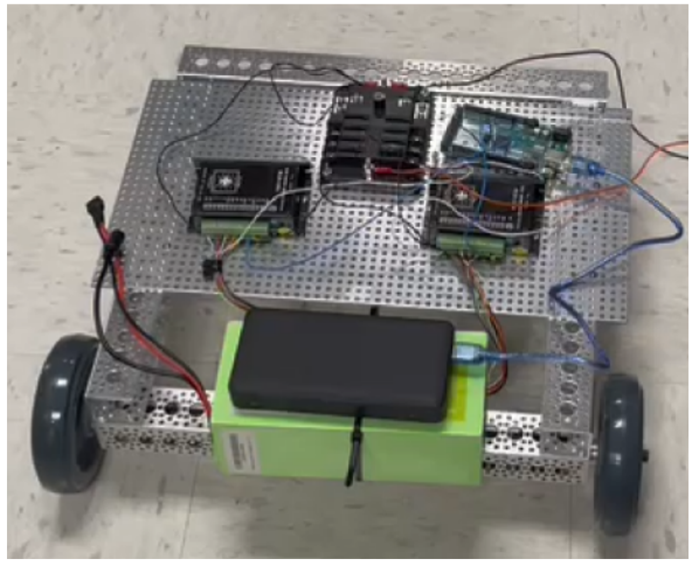


Fig. 16. Actobotics Chassis 15lbs

The manufacturing process involved creating a blueprint on paper to determine the precise dimensions and number of 0.25-inch thick acrylic sheets. The acrylic pieces, two sheets of 24 W * 48 L * 1/4 D inches, were cut from the sheets, heated with a torch for 15-20 minutes for bending, and then joined together using acrylic glue. Two identical pieces were made, with six acrylic tubes mounted inside the wall, each with an internal thread glued inside. Two platforms were made, one for the second floor and another in the center, with seven acrylic pieces attached using glue for the second-floor platform. The chassis was painted using a paint sprayer after sanding, left to dry for several hours, and then the wheels were attached and assembled. Fig. 17 shows the chassis sanded and Fig. 18 shows the finished and painted chassis.

Fig. 17. Acrylic Chassis - Side View

The new acrylic chassis is complete and weighs 20 lbs with all the mounted parts and components within the 25 lbs weight limit. This design choice allowed for a weight reduction and closer adherence to the final design, making it ready for further testing.

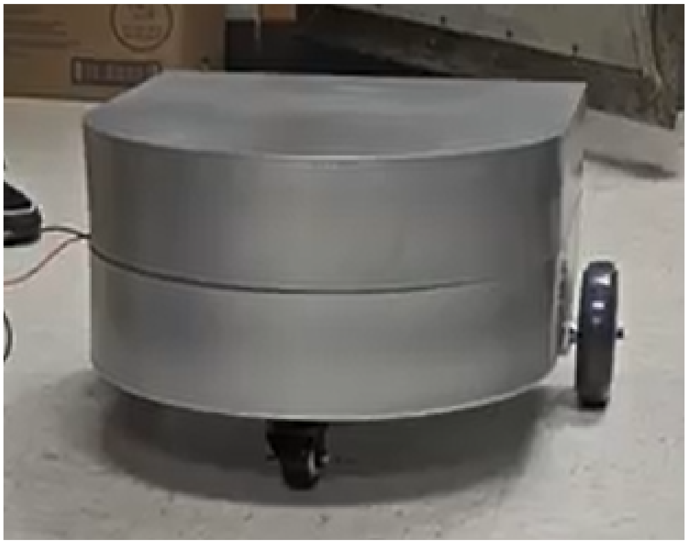


Fig. 18. Finished Acrylic Chassis

3) Prototype Development and Testing: The Fall 2022 prototype used a heavier and larger frame, 20 by 20 inches, Fig. 10. While the new chassis was being constructed, the team opted to construct a light 16 by 15-inch frame using Actobotics Channels [44] , which are lightweight and narrow, Fig. 16.

Since the new prototype used BLDC motors, the team replaced the motor controllers compatible with the motors. When comparing past motor controllers to the one used in this iteration of the CleanBot, the BLD-120A motor controller is significantly better due to its various functions. Besides the previously mentioned potentiometer, other noteworthy and welcomed capabilities of the BLD-120A are its current and voltage protection, error indication light, and ability to incorporate hall sensors.

The error indication light on the BLD-120A, the color, and the flashing sequence denotes current and voltage problems within the system. For example, two red flashes for an over-voltage warning, six red flashes for hall sensor errors, etc. [40-42]. This feature proved helpful within the drive system tests by helping the team diagnose a connection issue we encountered. For example, shortly after completing the wiring and beginning the motor acceleration within the first stage of testing, one motor stopped responding while the error indicator light repeated a sequence of three green flashes. Although absent from the documentation, research and experimentation showed that the error indicated was due to an improper connection between the motors and motor controller, a loose wire connection in one of the controller terminals.

The overarching goal of the drive system tests performed upon the spring 2023 iteration of the CleanBot 3000 was to confirm whether or not the new BLDC gear motors chosen produce enough torque to move the CleanBot weighing less than 25 lbs around. In addition, it is important to verify that the robot can operate at a slow speed to enable the UVC lights to sanitize the JPL cleanroom floors well. Similarly, it is also important to verify its ability to move up a 30-degree incline because of the possible variance in the angle of surfaces it will be required to traverse and clean. The second and equally important goal was to assess the overall integrity of the communication between the Arduino Mega and the BLDC motors; this second goal would confirm the differential drive capabilities of the drive system by operating the motors at asynchronous speed and alternate directions. Furthermore, it effectively verifies that the robot will adhere to the instructions stipulated within the Arduino. In addition, the rated current, stall (max) current, voltage requirement, and other parameters within the specification sheet were verified during the testing process.

The tests were split into two stages of operating the motors: direct driver motor control and Arduino motor control. The distinction between the two stages is reinforced by the built-in potentiometer of the BLD-120A motor controller, one of the more important features absent from the previous motor controllers.

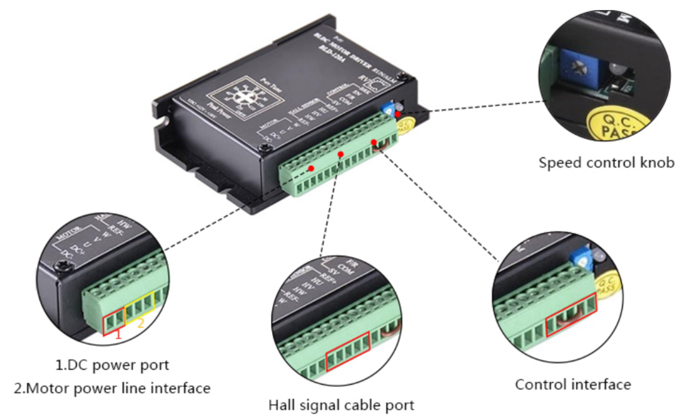


Fig. 19. BLD-120A Driver Ports. [41]

Stage One: Direct Driver Control

The potentiometer on the BLD-120A motor controller functions as a variable presence of resistance capable of obstructing the current flow from the Arduino's output pins to the motors'. When the potentiometer on the BLD-120A controller is set to its maximum resistance, the motor's speed is solely dependent and responsive to the power supply voltage level; all programming and commands within the Arduino code are ineffective. A lower voltage level equals slower motor speed. When the potentiometer is set to its minimum resistance, the motors adhere to the instructions within the Arduino code. Refer to Fig. 19 and 20 for the BLD-120A driver ports [41], interface, and wiring diagram [42].

The new BLDC planetary gear motors were connected to individual motor drivers with the potentiometer on the driver set to its maximum resistance. Two (6A, 250V) toggle switches were included as cut-off switches; this strategy was a way of

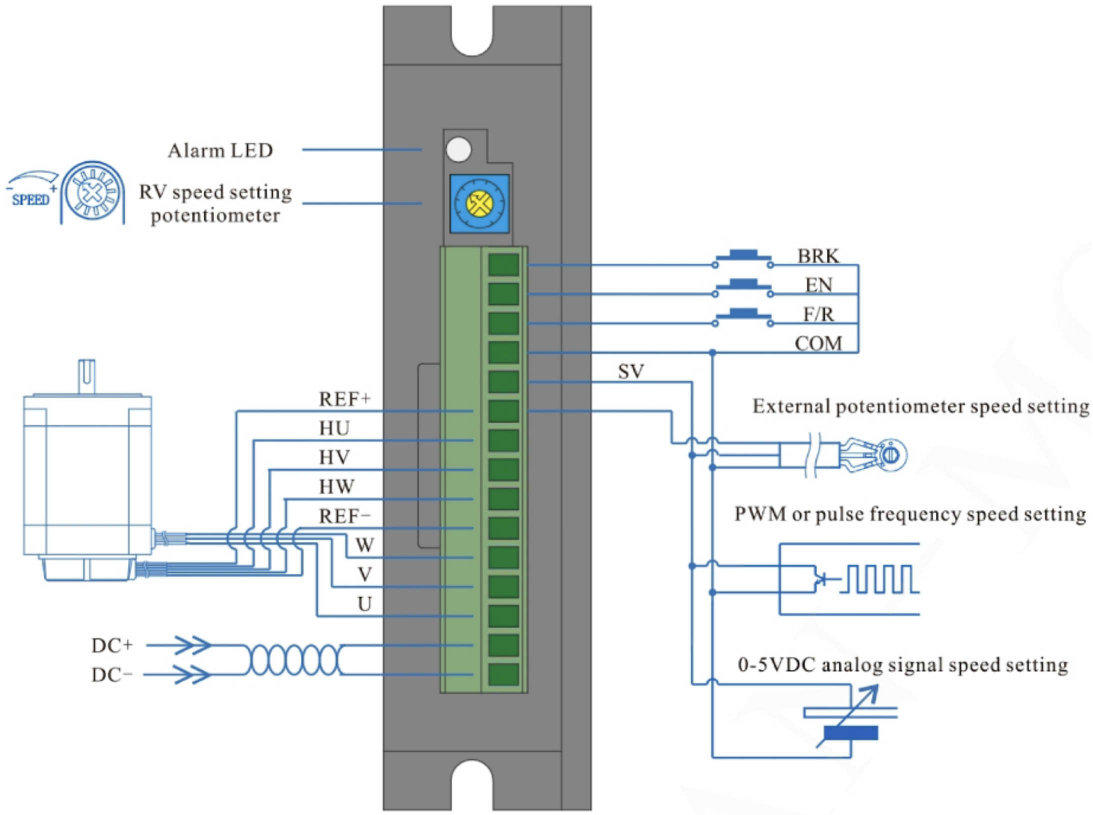


Fig. 20. BLD-120A Driver interface and wiring diagram. [42]

adhering to the safety protocols while giving us partial control of the motors (in the absence of an Arduino).

One toggle switch was connected in parallel with the forward-reverse terminal (F/R) and ground (on the controller). Whenever this terminal is shorted (connected to the ground), the motors rotate counterclockwise; vice versa when it isn't. The second toggle switch was connected from the brake terminal (BRK) to the ground. However, functionally an emergency brake, this immediate removal and supply of input power allowed us to monitor the continuous and stall current drawn. This limited functionality, of being able to dictate rotation direction and start-stop the motors, were the only two aspects of controlling the motor available in the first stage. The second stage offered more precise motor control due to the modulation of the width of the input signal to the motor controller.

These tests assessed each motor's speed, direction, and brake control and function successfully. Also, based on these two tests, the setup parameters were determined: a supply voltage of 24V and a 1A current rating for the fuse. These tests verified the motors' specifications and functionalities. These current and voltage ratings will be used to set up the power system to power the drive system efficiently and safely. The motors' function controls will be integrated into the Arduino code for the next testing stage.

A speed test was also conducted in this testing stage, and the motors were connected to the controllers without the Arduino. This setup only uses the direct driver control from the BLD-120A controller. The speed potentiometer of the controllers was set to full speed, and the directions were set to move forward. The test starts when the motor turns on and drives forward on a straight 1-meter line. The time it took to cover 1 meter was determined as 5.5 seconds; therefore, the resulting speed was 0.18 m/s. Consequently, we can conclude that the motors are more than capable of carrying 20 lbs, and to reduce speed, the PWM settings from the Arduino code can be modified, starting with 50% duty cycle settings.

Stage Two: Arduino Control with PWM Signal

The second stage of the testing includes the Arduino Mega 2560, which sends a PWM signal through the driver to control the motors. This setup was done for the 15 and 20 lbs chassis, Fig. 16 and 18. The code sends instructions that alter the speed and direction of each motor individually. More specifically, the code running on the Arduino code is a set of simple instructions used to demonstrate the capability of the robot's drive system: forward and backward motion, slow turning in both directions, and stationary rotation in both directions. To make identifying each set of executed commands easier, the code also increases delay time and speeds up between each command. The switches from the first testing stage were replaced by the control pin connection to the Arduino, and the code specifies what each pin intends to do, similar to the switches' functionality.

The team tested the new prototype's ability to perform differential drives with a weighted load using Arduino code. This test aimed to ensure that the motor's torque and performance were sufficient to enable the drive system to operate smoothly on flat and inclined terrain. The test was carried out with both 15 and 20 lbs chassis, and the robot successfully executed the differential drive without slipping or stalling during all three Arduino code commands. The robot's ability to move at low speed while maintaining high torque is vital to its effectiveness in the sanitization process it was designed for.

During these tests, the team observed that the amount of input voltage supplied to the motor drivers dictates the speed at which the motors rotate. Controlling this aspect of the motors requires defining a variable within the code to operate a PWM-capable output pin on the Arduino while specifying the voltage level. The 5V output signal of the Arduino pins is divided into a range of 0 to 255. Similarly to the first testing stage, the direction of rotation is governed by the presence of a high or low-voltage signal at the forward-reverse terminal of the motor controller. Controlling this aspect of the motor's behavior in this stage was defining a variable within the code which pertains to any output pin of the Arduino (not exclusively a PWM-capable pin). Manipulation of this variable and pin can then be effectively governed by instructions specified within the Arduino code.

Notably, the distinct value in accomplishing this testing stage is verifying the proposed revision and improving the overall drive system of the previous CleanBot prototype. The idea of controlling the two BLDC motors is meant to move the robot around with a PWM input signal from an Arduino.

Additionally, the Arduino code was modified to conduct the speed test under load conditions (20 lbs). The test aimed to verify that the robot can be driven slowly for adequate sanitation using UVC sanitizing modules currently on the market. Research and analysis of the dosimetry tests from the previous semester show that a required average rate of 0.05 m/s is necessary for adequate sanitation. An Arduino code was written to assess the robot's speed by having the robot move one meter in a straight path, and code was set to 50% PWM duty cycle.It took an average of 13 seconds to move one meter; this represents a peak speed of 0.077 m/s with 50% PWM duty cycle settings.

4) Presentation and Analysis of Results: The prototype development and testing phase results were presented and analyzed to assess the functionality and performance of the motor drive system. Regarding the motor driver functionality, the tests confirmed that it performs direction, speed, and brake control with precision. These results will be considered when creating the code for the Arduino control. In addition, the brake function will also be integrated into the system's control to improve the robot's drive and halt when the collision mitigation, through Sonar, is integrated back into the system.

The drive system's capability to perform differential drive using Arduino code control was successfully confirmed, with the 15 lbs and 20 lbs prototypes demonstrating the ability to carry a load of up to 25 lbs efficiently. This is a positive sign that the drive system meets the weight requirements of the robot. In addition, the new BLDC planetary gear motors generated enough linear force to move the robot with a curb weight of 20 pounds, 5 pounds below JPL's stipulated weight limit. Moreover, the reliability of the differential drive was validated by the robot's smooth movement in opposite directions at asynchronous speeds.

The speed test using direct driver control and Arduino control yielded results of 0.18m/s and 0.077m/s, respectively. The peak speed of the robot with a 20 lbs load was achieved using direct driver control set to full speed settings. This speed is at least triple the ideal average speed of 0.05m/s. The test concludes that the motor's capacity to carry 20 lbs is well within its capability, and adjusting the PWM signal settings from the Arduino can decrease the speed if necessary.

On the other hand, the second speed test with the Arduino control measured a speed of 0.077m/s, which is closer to the ideal average rate of 0.05m/s. The Arduino code was set to 50% PWM duty cycle. The duty cycle can be reduced accordingly to achieve the ideal average speed.

Additionally, the team could not confirm the robot's ability to move up a 30-degree incline in time, the team is looking into conducting this test during the summer break. Still, the output torque of the motors instilled confidence that the robot could operate on sloped surfaces. Therefore, the tests conclude that the new motors now have suitable torque output to propel the robot smoothly and slowly around under loaded conditions.

Nonetheless, it should be emphasized that the team is still awaiting the dosimetry test results to validate the optimum speed required for the robot to perform effective sanitation of the JPL cleanroom floors. In addition, a sturdy drive system is essential for CleanBot to navigate and disinfect the JPL cleanroom floors efficiently.

Once the dosimetry test results determine the optimal speed for effective sanitation, the team will conduct further tests to ensure the robot's performance under specific conditions. These tests will include driving the robot up a 30-degree incline and over floor cable covers to assess its ability to navigate and sanitize different surfaces and obstacles.

V. Sanitation

A. Dosimetry Tests

The JPL has completed dosimetry testing, which measures UVC radiation intensity and spectral distribution from the sanitizing modules. The JPL team is analyzing the data and will provide a report to aid CleanBot's development by the semester's end. The data will also provide more accurate information on the required velocity and LED exposure dosage for effective sanitation. Dosimetry testing ensures that UVC radiation safely sanitizes cleanroom floors. The test uses specialized equipment to measure UV radiation intensity and spectral distribution and verify UVC exposure of surfaces or instruments [47]. Results determine the appropriate exposure time and ensure compliance with regulations.

The results of the dosimetry test will be interpreted to calculate the optimal speed of the CleanBot to sanitize the cleanroom floors efficiently. The current prototype's speed of 0.05m/s was based on research and calculations conducted in previous semesters and referenced in [48], which investigated pre-existing UVC sanitizing AGVs in the market.

VI. Conclusion

During the Spring 2023 semester, the mapping and navigation team successfully implemented ROS onto the Raspberry Pi and ran SLAM using the RPlidar to map an environment on RVIZ. The team failed to get the FCPP simulated on RVIZ, but the research conducted on the FCPP wil help the following semester better understand how the code works. Before implementing the FCPP onto the Raspberry Pi the team must resolve the issues encountered when attempting to run the FCPP on RVIZ. This would allow the team to test how changing the code's parameters affects the robot's navigation route. The next semester will be able to use the research conducted this semester as a stepping stone to successfully implementing the FCPP onto the Cleanbot prototype.

During the Spring 2023 semester, the Power & Mechanical Design team successfully developed a new Cleanbot prototype that met JPL's standards and addressed issues with the previous prototype. The team conducted comprehensive testing of the new CleanBot prototype, which includes speed and performance tests with a 20 lb load, to ensure functionality. The speed test was performed to determine the capacity of the motors to perform under peak conditions. The speed measured at peak speed was 0.18 m/s, which is at least three times faster than the ideal average speed of 0.05m/s. These specific tests conclude that the speed can be reduced further by adjusting the PWM signal from the Arduino.

The tests also concluded that the new prototype could perform differential drive while subject to specific and peak load conditions. The team can further adjust, optimize, and fine-tune the drive system settings to reach their ideal speed. The successful testing of the new prototype and its components is relevant to the team's goal of developing a modular and functional Cleanbot robot that satisfies JPL's requirements. The new prototype is a foundation for future semesters to build upon and improve CleanBot's capabilities.

References

[1] D. J. Smith, "A Balloon-Based Payload for Exposing Microorganisms in the Stratosphere (E-MIST)," Gravitational and Space Research, vol. 2, no. 2, Dec., pp. 70-80, 2014.

[2] "NIH Human Microbiome Project defines normal bacterial makeup of the body," National Institutes of Health, 31-Aug-2015. [Online]. Available: https://www.nih.gov/news-events/news-releases/ni h-human-microbiome-project-defines-normal-bact erial-makeup-body.

[3] Santos, J. (Ed.). (n.d.). move_base_flex. ros.org. http://wiki.ros.org/move_base_flex

[4] Documentation. ros.org. (n.d.). http://wiki.ros.org/

[5] T. Clephas and C. López, Eds., "nobleo / full_coverage_path_planner," GitHub.com. https://github.com/nobleo/full_coverage_path_planner

[6] T. Clephas and C. López, Eds., "nobleo / tracking_pid," GitHub.com. https://github.com/nobleo/tracking_pid/

[7] M. Hansen, Ed., "mrath / mobile_robot_simulator," GitHub.com. https://github.com/mrath/mobile_robot_simulator*

[8] Wiki. ros.org. (n.d.). http://wiki.ros.org/hector_mapping

[9] Shahrizal Saat et al 2020 J. Phys.: Conf. Ser. 1529 042032

[10] Ed. (2021, September 28). Using ROS on Raspberry Pi: Best practices - the robotics back. https://roboticsbackend.com/using-ros-on-raspberry-pi-best-practices/

[11] Huang, T. (n.d.). RPLIDAR-A1 360°Laser Range Scanner domestic laser range scanner. SLAMTEC. https://www.slamtec.com/en/Lidar/A1

[12] "Zybo Z7 Reference Manual - Digilent Reference," digilent.com. https://digilent.com/reference/programmable-logic/zybo-z7/reference-manual?redirect=1

[13] Taylor, Adam. "Building a Camera / Imager Test Platform." Digilent Projects, 3 Dec. 2018, https://projects.digilentinc.com/adam-taylor/building-a-camera-imager-test-platform-9d9 A5d.

[14] "DATASHEET Raspberry Pi 4 Model B Release 1," 2019. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

[15] ed, "Install Ubuntu 20.04 on Raspberry Pi 4 (without monitor)," The Robotics Back-End, Jul. 06, 2020. https://roboticsbackend.com/install-ubuntu-on-raspberry-pi-without-monitor/ (accessed Apr. 18, 2023).

[16] A. Industries, "Adafruit Sensirion SHT40 Temperature Humidity Sensor," www.adafruit.com. https://www.adafruit.com/product/4885 (accessed Apr. 18, 2023).

[17] "adafruit/Adafruit_CircuitPython_SHT4x," GitHub, Oct. 03, 2022. https://github.com/adafruit/Adafruit_CircuitPython_SHT4x (accessed Apr. 18, 2023).

[18] CloudFlare, "What is UDP? — Cloudflare," Cloudflare, 2022. Available: https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/

[19] P. Fox, "User Datagram Protocol (UDP) (article)," Khan Academy, 2020. https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/user-datagram-protocol-udp

[20] A. Industries, "Raspberry Pi High Quality HQ Camera," www.adafruit.com. https://www.adafruit.com/product/4561

[21] A. Industries, "6mm 3MP Wide Angle Lens for Raspberry Pi HQ Camera," www.adafruit.com. https://www.adafruit.com/product/4563 (accessed Apr. 18, 2023).

[22] J. Dizon, O. Dabach, S. Blazic et al, "CleanBot 3000" California State University Northridge, Northridge, CA, tech., 2022.

[23] "Easy Blade - CAN-Bus enabled modular battery with over 1500Wh per module," www.varta-ag.com. https://www.varta-ag.com/en/industry/product-solutions/lithium-ion-battery-packs/asb/easy-blade (accessed Apr. 14, 2023).

[24] "Amazon.com : H HAILONG Ebike Battery Pack 36V 12Ah Ebike Battery/36V 8Ah 20Ah 48V 12Ah 20Ah Electric Bike Lithium Ion Battery with 2Ah Charger for 350W 450W 750W 500W 800W Electric Bicycle Motor(36V 12Ah) : Sports & Outdoors," www.amazon.com. https://www.amazon.com/dp/B09LQ93D3S?psc=1&ref=ppx_yo2ov_dt _b_product_details (accessed Apr. 14, 2023).

[25] "Drive Motor Sizing Tool," RobotShop Community, Mar. 07, 2013. https://community.robotshop.com/blog/show/drive-motor-sizing-tool (accessed Apr. 12, 2023).

[26] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," Renewable and Sustainable Energy Reviews, vol. 78, pp. 834–854, Oct. 2017, doi: https://doi.org/10.1016/j.rser.2017.05.001.

[27] X. Hu, S. E. Li, Z. Jia, and B. Egardt, "Enhanced sample entropy-based health management of Li-ion battery for electrified vehicles," Energy, vol. 64, pp. 953–960, Jan. 2014, doi: https://doi.org/10.1016/j.energy.2013.11.061.

[28] H. Ren, Y. Zhao, S. Chen, and T. Wang, "Design and implementation of a battery management system with active charge balance based on the SOC and SOH online estimation," Energy, vol. 166, pp. 908–917, Jan. 2019, doi: https://doi.org/10.1016/j.energy.2018.10.133.

[29] "What is telemetry and how does it work?," WhatIs.com. https://www.techtarget.com/whatis/definition/telemetry

[30] "LiPo Battery Monitor," The Fancy Voyager! https://thefancyvoyager.com/best-lipo-battery-review/monitor/

[31] Klaran, "Klaran LE," 120220, 2020

[32] "How to Build a Robot Tutorials - Society of Robots," www.societyofrobots.com. https://www.societyofrobots.com/electronics_fuse_tutorial.shtml (accessed Apr. 14, 2023).

[33] "QNCZ 2" Caster Wheels, Casters Set of 4 Heavy Duty but Silent, Excellent Locking Casters with Polyurethane (PU) Wheels, Swivel Plate Castor Wheels for Cart, Furniture, Workbench.: Amazon.com: Industrial & Scientific," www.amazon.com. https://www.amazon.com/gp/product/B0BPHF6317/ref=ppx _yo_dt_b_search_asin_image?ie=UTF8&psc=1 (accessed Apr. 11, 2023).

[34] "ATPEAM Shopping Cart Wheel Replacement Kit — 5" Diameter Polyurethane Stepped and Full Tread Face w/Double Ball Bearing Shopping Cart Wheel 1000 lbs Total Capacity, Pack of 4 (Gray): Amazon.com: Industrial & Scientific," www.amazon.com. https://www.amazon.com/gp/product/B09MKZ9V2Y/ref=ppx _yo_dt_b_search_asin_title?ie=UTF8&th=1 (accessed Apr. 11, 2023).

[35] O. Dabach, M. Valdez, J. Reyes, R. Aanan, J. Flores et al, ”CleanBot 3000” California State University Northridge, Northridge, CA, tech., 2022.

[36] Robert, ”Types of Wheels Used in Robotics,” Robotics Shop, Jun. 09, 2020. https://roboticsshop.net/types-of-wheels-used-in-robotics/#: :text=Types%20of%20Robot%20Wheels %201%20Standard%20Fixed%20Wheel

[37] “An Overview of Wheeled Mobile Platform Systems • Smashing Robotics,” Smashing Robotics, Jul. 11, 2012. https://www.smashingrobotics.com/an-overview-of-wheeled-mobile-platform-systems/

[38] J. Zhao and Y. Yu, “Brushless DC Motor Fundamentals Brushless DC Motor Fundamentals Application Note,” 2011. Accessed: Apr. 15, 2023. [Online]. Available: https://media.monolithicpower.cn/document/Brushless_DC_Motor _Fundamentals.pdf

[39] “BLWRPG09 - Brushless Motors with Planetary Gearboxes,” www.anaheimautomation.com.

[40] “BLD120A 24V 120W BLDC Motor Driver - RobotDigg,” www.robotdigg.com. https://www.robotdigg.com/product/1234/BLD120A-24V-120W-BLDC-Motor-Driver (accessed Apr. 15, 2023).

[41] “Brushless DC Motor Controller, 12V 30V,” ATO.com. https://www.ato.com/brushless-dc-motor-controller-12v-30v

[42] “SYS-BLD-120A DC Brushless Motor Driver Overview: SYS-BLD-120A Suitable for speed regulation of three-phase DC brushless motors with power below 125W.” Accessed: Apr. 15, 2023. [Online]. Available: https://www.sys-motor.com/Account/Plug-ins/kindeditor/attached/file/20180507/20180507153333_7585.pdf

[43] “1201 Series Quad Block Pattern Mount (43-2),” ServoCity. https://www.servocity.com/1201-series-quad-block-pattern-mount-43-2/ (accessed Apr. 15, 2023).

[44] “U-Channel (19 Hole, 15.00” Length),” ServoCity. https://www.servocity.com/15-00-aluminum-channel/ (accessed Apr. 15, 2023).

[45] “1116 Series Grid Plate (29 x 53 Hole, 232 x 424mm),” ServoCity. https://www.servocity.com/1116-series-grid-plate-29-x-53-hole-232-x-424mm/

[46] “1120 Series U-Channel (17 Hole, 432mm Length),” ServoCity. https://www.servocity.com/1120-series-u-channel-17-hole-432mm-length/

[47] “UVC Dosimeter American Ultraviolet,” https://www.americanultraviolet.com/germicidal-healthcare-solutions/uvcdosimeter.html#: :text=The%20UVC%20dosimeter%E2%84 %A2%20is%20a%20safe%20and%20efficient (accessed Apr. 15, 2023).

[48] “What You Need To Know About UVC Lights and Autonomous Floor Scrubbing Robots-Avidbots,” Avidbots. https://avidbots.com/resources/what-you-need-to-know-about-uvc-lights-and-autonomous-floor-scrubbing-robots/ (accessed Feb. 08, 2023).