AutoServe - Automatic Room Service Robot

ECE 445 Design Document - Spring 2026

Project Team #84

Ethan Jiang, Johan Martinez, Nikhil Vishnoi

Professor: Joohyung Kim

TA: Ko Po-Jen

Contents

1. Introduction

1.1 Problem

In hotels and other service industries, guests or residents often request small amenities such as snacks, toiletries, chargers and more. Fulfilling these requests typically requires manual labor, such as a staff member traveling long distances across hallways and between floors which is time-consuming, inefficient, and tedious. Especially during busy peak hours, this inconvenience leads to increased labor costs for these facilities, and delayed deliveries to customers which lowers satisfaction. Organizations such the Hoover Institution at Stanford and media outlets like ETFTrends are reporting that the adoption of service robotics has been increasing worldwide to address similar issues such as labor shortages and to improve efficiency and service quality. According to the International Federation of Robotics, “With 102,900 units (+14%) sold in 2024, more than every other professional service robot was built for the application class transportation and logistics,” highlighting the rapid growth of demand for mobile robots designed for supplementing service objectives.

However, while some automated delivery robots exist, current commercial solutions are extremely expensive and often impractical for smaller deployments or retrofitting existing buildings. While similar industries such as restaurants have begun to implement more widespread technology with service robots, other common hospitality facilities have not adopted any change yet due to a lack of solutions that are useful yet justifiable in cost. There is a need for an affordable yet flexible indoor delivery system capable of autonomously transporting small items within multi floor buildings while operating within existing building infrastructure constraints. An affordable autonomous service robot can also have several societal implications such as reducing worker fatigue, tackling workforce staffing shortages, and minimizing response times for operational efficiency.

1.2 Solution

Our proposed solution in this project is building an autonomous indoor delivery service robot capable of transporting items between locations in a multi-floor building such as a hotel. The robot will navigate hallways autonomously via communicated predetermined paths, and deliver items from a central base location, similar to a hotel lobby desk or snack bar in real scenarios, to a specified room destination. The robot's movement will be entirely autonomous and be monitored wirelessly by staff. Elevator actuation is assumed to be externally triggered by the building staff as is currently most common in real service situations, while the robot will autonomously handle entering, riding, and exiting the elevator at the correct floor with sensor detection (explained below). This design choice reflects realistic constraints of existing building logistics while allowing the project to focus on autonomous navigation, system integration, and practicality.

An ESP32-based controller located on the robotic navigation unit will handle sensor integration and communication with the base server. We would also incorporate graphed routes that are optimized for avoiding obstacles, and implement an IR based proximity sensor that avoids any obstacles not included in the graphed routes. Two brushless motors with encoders will be used to move the robot while accurately assessing its position. This system would reduce staff workload, improve response time for guests, and demonstrate how embedded robotic platforms can be useful to automate common but repetitive manual logistics tasks.

1.3 Visual Aid

Figure 1: Example conceptual diagram showing high-level interaction between robot system and destination, with communication outlined by basic arrows.

1.3 High Level Requirements

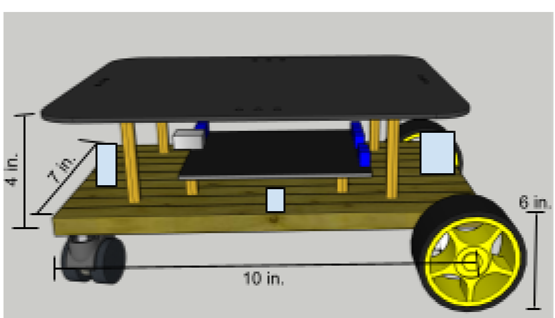
1. **Delivery:** Robot should be connected to computer via wifi sending back information about current task and taking commands to start deliveries. The robot itself should be capable of carrying 10 lbs of payload weight.
2. **Maneuverability:** Robot is able to detect and maneuver around simple obstacles within 10 centimeters (ex. person blocking a straight path).

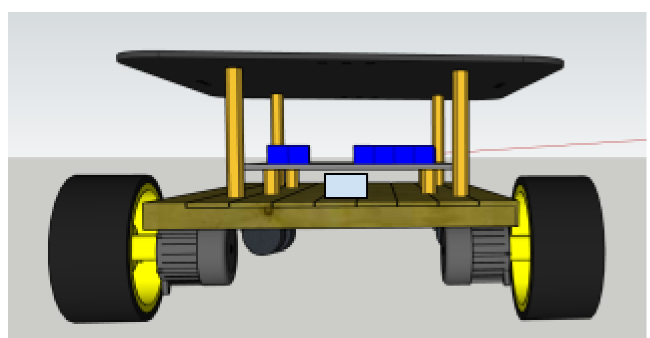
3. **Reliability**: The Robot should be able to use position sensing to accurately arrive at a location and return within 10 minutes and have a battery life that can last the duration of delivery.

2. Design

2.1 Physical Design

The picture below models a general sketch of what our physical navigational robot will aim to look like. The physical model includes two driver wheels, which will be 6 inches in diameter each connected to our hall encoder DC motors (B07GNGQ24C), and two caster wheels which will turn and follow the driver wheels. The motors are each 3.217 inches in width from end to end, and 2.7 inches without considering the axle that the wheel will go on because it sticks out of the board's width. The Base will contain our electrical board and other electrical components like our sensors, and attached will be our driver and caster wheels located in each respective corner. We will utilize the ECE Machine shop to construct the overall physical structure of our robot, and we plan to use wood for our platform material which will carry delivered items. The driver wheels should be built on a separate bridge, which will be connected to the rest of the structure by the bottom of our base board. Our platform dimensions of the bottom base that will hold our electronic components is 7x10 inches. The height of the raised platform which will carry our payloads should be around an additional 4 inches in height. Final dimensions may be adjusted upon determination by the machine shop.





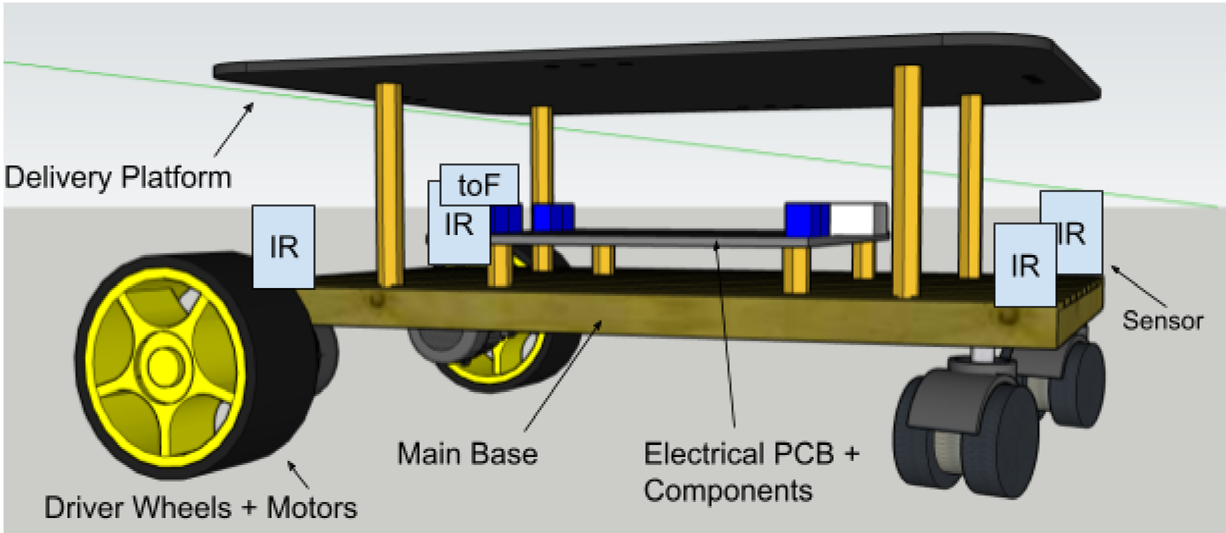


Figure 2: Different views of our general physical navigational robot design.

2.2 Block Diagram

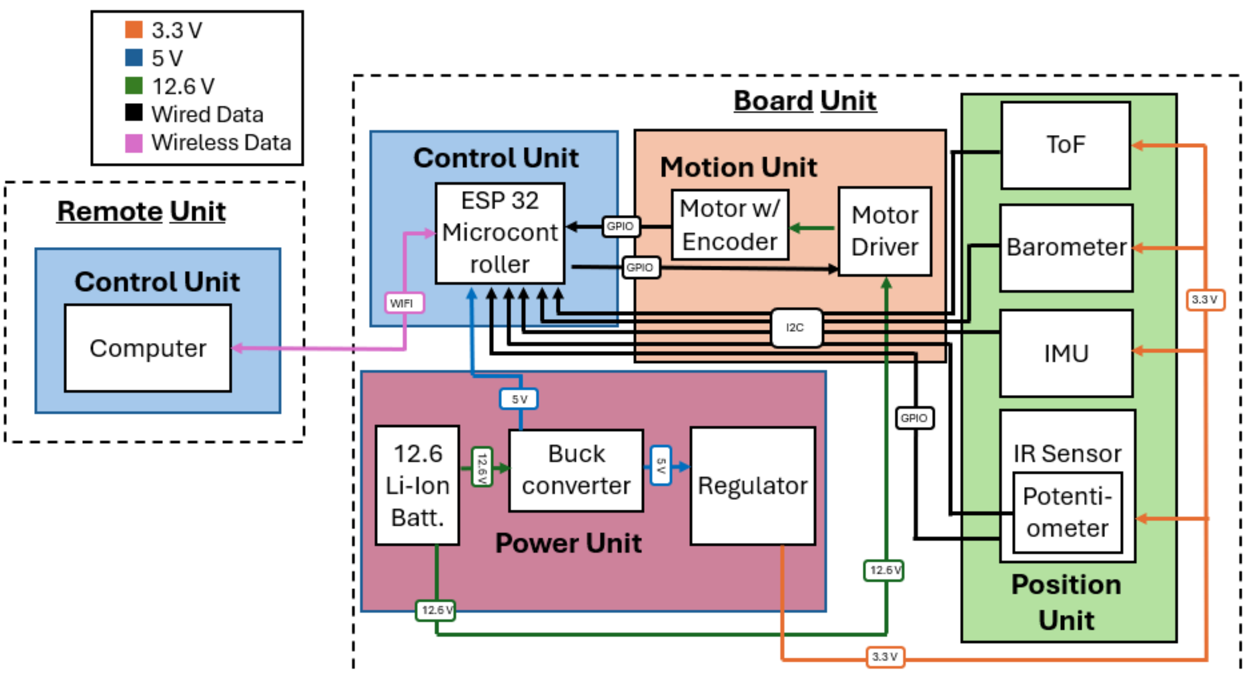


Figure 3: Block Design depicting our overall system integration.

The overall robotic ecosystem consists of a remote control unit, which we will use our computer for, and the navigation platform which will contain the board that connects to our necessary subsystems for the delivery robot. The motion unit will control the motors and our microcontroller will handle calculations for the path that we want our robot to take. The overall unit will be powered by our power distribution subsystem, consisting of a Li-ion powered battery that supplies 1A/12V to each motor. Positioning will be handled by the motor's encoders, and the IR sensor will be used to detect obstacles in a defined proximity.

2.3 Functional Overview & Block Diagram Requirements

2.3 Subsystem Requirements

2.3.1 Control Subsystem

The control subsystem is responsible for connecting the autonomous robot to a control computer to get the motions it needs to take on a preloaded map. The subsystem additionally reads data from the position subsystem to measure location on map. Using the position data the subsystem commands the motion subsystem to move the robot forward x feet at a constant speed of 3 ft/s. During movement the control subsystem constantly reads from the position subsystem to verify the front is clear; if it is not, it will command the motion subsystem to stop moving and wait until the obstacle is cleared. The control subsystem is made up with an ESP32-WROOM-32E-N4 Microcontroller Unit which interfaces with the position subsystem through GPIO pins to the IR sensor output and a I2C connection to IMU, barometric height sensor and digital potentiometer to change the IR sensor range; interfaces with the motion subsystem with GPIO pins to the motor driver and motor encoder; interfaces with the control computer through its built in Wifi module. The control subsystem will be powered by the power subsystem via a 3.3 Volt LDO regulator capable of supplying 1 Amp of current. For more information on the software design of the control subsystem refer to Section 2.5.2. To ensure the control subsystem is fulfilling its responsibilities for receiving controls from the control computer, reading sensor data and appropriately commanding the motor motion, a requirements and verification table can be found below.

Table 1 Board Subsystem - Requirements and Verifications

Requirements	Verification
• When Control Subsystem given list of motion commands (move 1 foot x, 2 feet y, 1 feet x,...) it is able to measure how far it is moving and accurately follow each command with a margin of +- 6 inches	• Control subsystem is able to quickly within 500 ms give a stop and start command with speed and runtime to the motion subsystem • System capable of moving motors backwards at slower speeds to account for any unplanned overstep • Control subsystem can read xy motion through combining imu and motor encoder outputs accurate to +- 6 inches
• When Control Subsystem detects obstacle within 1 foot blocking its motion it will instantly stop moving and alert the computer controller that it is stuck	• When moving, the control subsystem reads from IR detectors within 200 ms to assure that there is no detected obstacle • If there is an obstacle detected must stop the motion subsystem movement within 200 ms • Subsystem must command motion subsystem to move back until it is 2 feet away and inform computer control system of a unknown obstacle • Subsystem is always able to take new master commands from computer control system with abandoning previously given paths
• When Control Subsystem given information to go to next floor it uses the elevator with human/operator assistance	• Subsystem enters elevator it must poll the barometric height sensor until it reaches the level it wants • Subsystem should remember its entrance steps and appropriate exit steps to leave elevator • If elevator disconnects subsystem from computer control system should reconnect after leaving elevator within 20 seconds

2.3.2 Motion Subsystem

The motion subsystem is responsible for taking motion commands and converting them into appropriate motor controller inputs. The subsystem reads data from the motor encoders as a feedback that the motors are rotating at their desired speed. The motion subsystem is made up of the motor driver (DRV8871) and hall encoder DC motors (B07GNGQ24C) connected to the control subsystem via GPIO. For hardware design of the motor driver refer to Section 2.4.1.The motion subsystem will be directly powered by a 12.6 Volt Li ion battery from the power subsystem. For more information on the software design of the motion subsystem refer to Section 2.5.1. To ensure the motion subsystem is fulfilling its responsibilities for properly executing commands from the control subsystem with encoder feedback, a requirements and verification table can be found below.

Table 2 Motion Subsystem - Requirements and Verifications

Requirements	Verification
• When Motion subsystem given command to move <x,y> coordinates at a given speed and time it individually controls the motor speeds to achieve desired motion within a speed margin of 0.5 ft/s and a angle margin of +- 5 degrees	• System is able to control both motors and have them rotate forward at the same speed using encoder data as feedback to achieve a maximum motion of 3ft/s • System can accurately do non straight motions accurate to +- 5 degrees by rotating the motors at different premeasured speeds.
• Motion subsystem is capable to drive the motors at stall current with a minimum of 1 Amp delivery to each motor	• Motor driver is able to transfer enough current at the motors 1 Amp to each • Output from motor drivers does not randomly drop from unstable voltage supply through use of bypass capacitors

2.3.2 Position Subsystem

The position subsystem is responsible for getting environment data and motion data to the control subsystem enabling the robot to make the proper decisions and competently verify the motion made by the motion subsystem. The motion subsystem is made up of the IMU (LSM6DSO32TR), a barometric sensor (BMP585), a time of flight sensor (LSM6DS3TR) and 4 IR sensors with range setting through a digital potentiometer (DS3502U+T&R). For more information on the hardware design of the IR sensor refer to Section 2.4.2. The position subsystem's barometric sensor, IMU and potentiometers are IC modules connected to the control subsystem via a I2C connection. The digital output of the IR sensors are also connected to the control subsystem via GPIO pins. Additionally the hall motor encoders on the B07GNGQ24C motors GPIO input into the control subsystem are utilized by the position subsystem to verify the IMU data. The position subsystem will be powered by the power subsystem via a 3.3 Volt LDO regulator capable of supplying 1 Amp of current. For more information on the software design of the control subsystem refer to Section 2.5. To ensure the position subsystem is fulfilling its responsibilities for sending IMU, barometric and IR sensor data to the control subsystem, a requirements and verification table can be found below.

Table 3 Position Subsystem - Requirements and Verifications

Requirements	Verification
● Collision sensors can detect objects within 1 foot where the robot might want to move	● Combined placement of ToF and 4 IR sensors covers confidently covers the center forward motion along with all four edges of the robot
● When the sensor detects obstacles within 1 foot of the robot it is able to find and report its distance to the robot within a margin of +- 6 inches	● Every 20 ms the system checks the toF sensor for any object directly in front of the PCB ● During motion every 20 ms the position subsystem varies the potentiometer through 100-10k until a obstacle is detected ● On obstacle detection the position subsystem sends a kill motion command to the control subsystem ● Once still the position subsystem finely varies the potentiometers on all IR sensors to detect the distance from the robot and the obstacle and reports to the control subsystem ● IR detector is setup and different potentiometer values are measured for their detectability ● For safety IR detectors are calibrated to detect dark obstacles as they reflect the least amount of light allowing a margin of overestimating how close lighter objects appear by ~15 cm
● When moving combination of IMU, motor encoder and barometric gauge used to get xyz within a margin of +- 5 inches	● Barometric gauge is able to be calibrated and used to accurately measure z motions between floors accurate to the floor ● IMU is double integrated to get position and compared with the motor encoder data to verify the motor encoder motion accurate to +- 0.5 feet ● Gyro data of IMU is able to get the turning angle for change in direction motions accurate to +- 5 degrees

2.3.2 Power Subsystem

The power subsystem is responsible for delivering high current power to the motors and power to the control and position subsystem. The power subsystem is made up of a buck converter (12->5V) and a 3.3V/1A LDO regulator (LM3940IT), and a 12.6V/3000mAh rechargeable Lithium Ion Battery (XZ02). For more information on the hardware design of the buck converter using a PWM controller refer to Section 2.4.3. The voltage from the Lithium Ion battery is directly connected to the motion subsystem and the voltage from the LDO is connected to the position and control subsystem. To ensure the power subsystem is fulfilling its responsibilities of powering the entire system, a requirements and verification table can be found below.

Table 4 Power Subsystem - Requirements and Verifications

Requirements	**Verification**
● Power subsystem is able to deliver 2 Amps at 12 Volts to the Motion Subsystem	● Lithium Ion battery is able to output its expected voltage and current can be a little higher than 12 also just must be above 12 volts as a minimum
● Power subsystem is able to deliver 800mA at 3.3 Volts to the control and position subsystem	● Buck converter must be efficient to reduce excess heat and loss of current to enable the most of the remaining 1000mAh from the Lithium ion battery to make it through LDO ● LDO must have proper circuit setup and verified that it can output at least 800 mA to take care of upper extreme estimations of current draw from subcomponents of these systems

2.4 Hardware Design

2.4.1 Motor Driver

To control the motors an electronically controllable H-bridge motor driver (DRV8871) will be used. This IC is capable of driving 3.5 Amps at 12 Volts which is more than enough to drive the motors. The motor direction is controlled by two digital pins into the IC. The different operating modes and a typical application circuit for this driver is shown below.

IN1	IN2	OUT1	OUT2	DESCRIPTION
0	0	High-Z	High-Z	Coast; H-bridge disabled to High-Z (sleep entered after 1 ms)
0	1	L	H	Reverse (Current OUT2 → OUT1)
1	0	H	L	Forward (Current OUT1 → OUT2)
1	1	L	L	Brake; low-side slow decay

Figure 4: H-Bridge Control

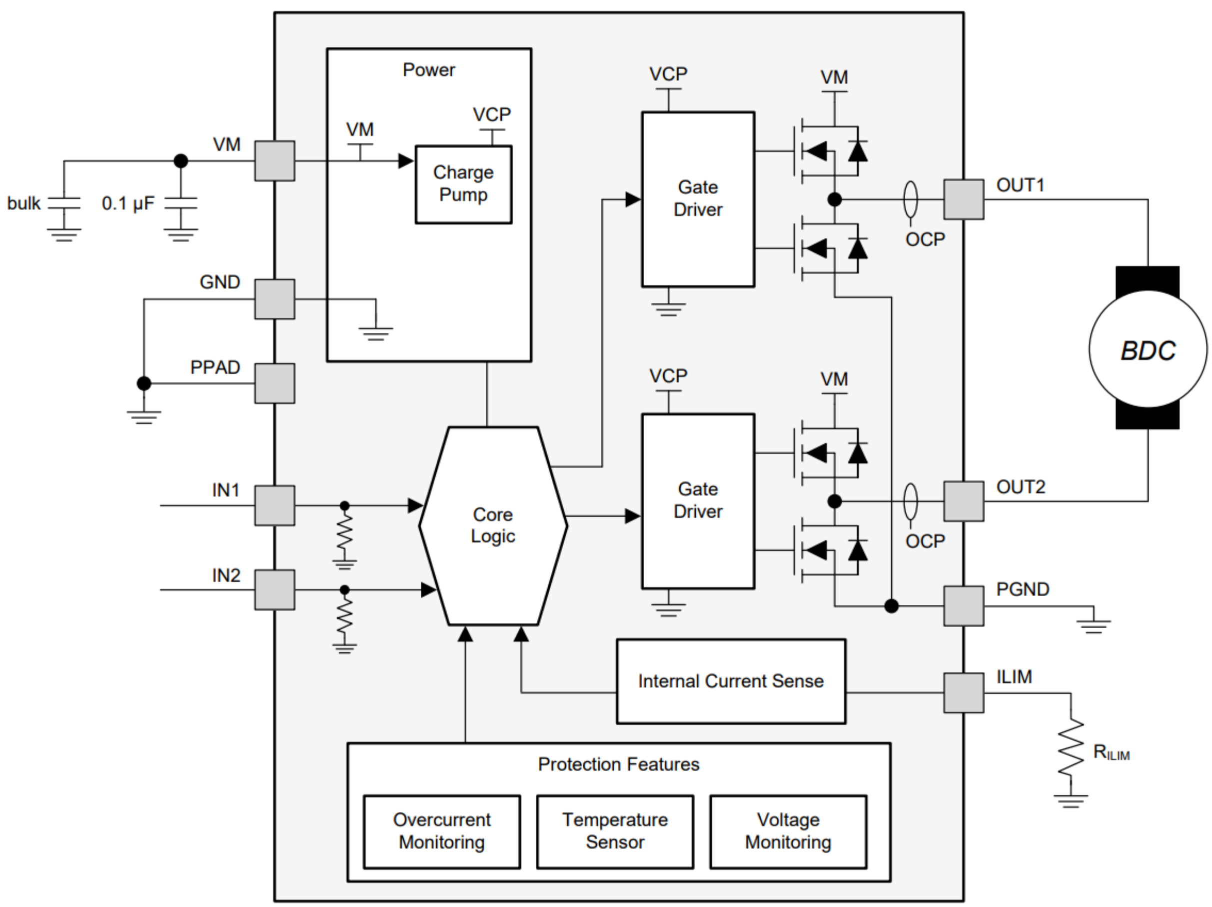


Figure 5: Typical Application of a DRV8871 Motor Driver

2.4.2 IR Sensor

The IR sensing circuit uses a IR LED & photodiode pair, an amplifier, a 10k ohm potentiometer and 2 220 ohm and 1 10k ohm resistor. The idea behind the circuit is that the reflections of infrared off objects will decrease the resistance of the photodiode which the amplifier can pick up and output. A simple example of this circuit is shown below. By changing the value of the potentiometer the sensitivity of the IR sensor can be changed thus resulting in a different detector range. For our circuit we will be replacing the 10k Ohm potentiometer with a digital potentiometer to software control the sensitivity. Each IR sensor will share the same Potentiometer allowing the position subsystem to concurrently check through the scan ranges of each sensor.

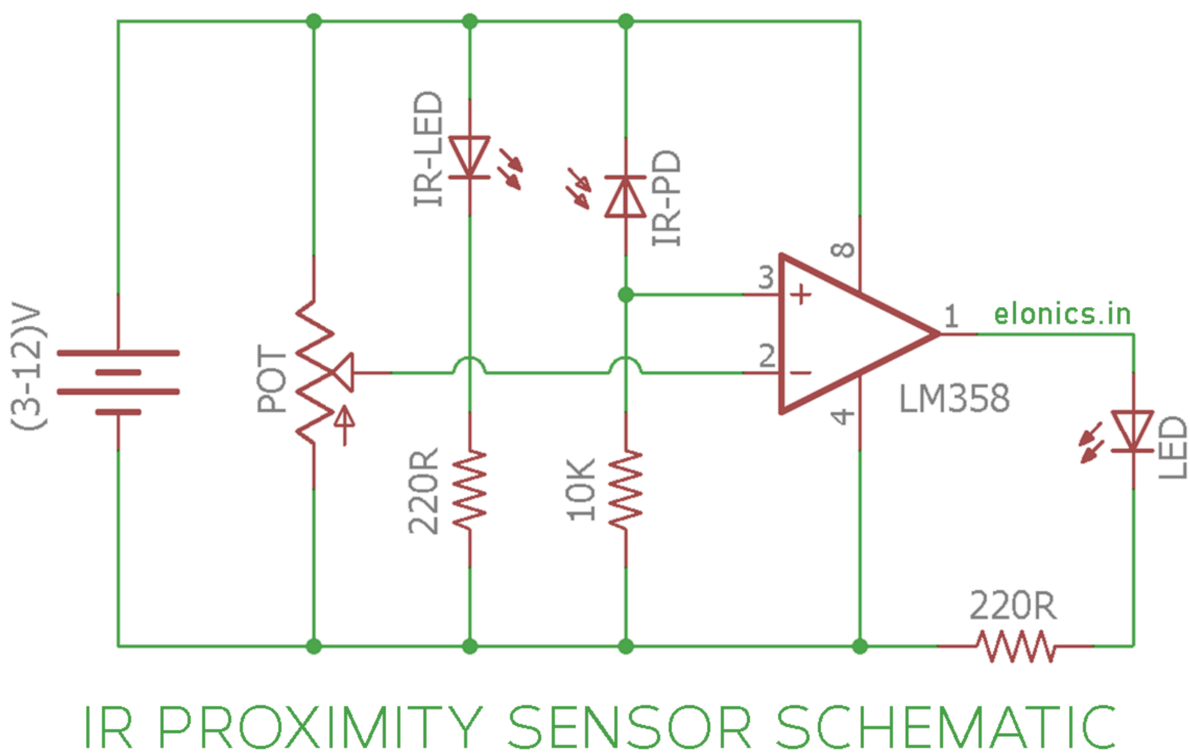


Figure 6: Example IR sensor circuit

2.4.3 Buck Converter

To step down the 12.6 Volts from the power supply to an acceptable value for the LDO a buck converter is used for its efficiency and higher current outputs. The buck converter circuit will be based on a PWM controller (TL494). The basis of a Buck converter is switching a switch on and off between the power supply and a RLC circuit. Based on the switching speed the RLC circuit settles on a lower voltage than the input power supply. To do this switching a PWM controller is used to turn a BJT transistor. An example circuit capable of converting 32 Volts into a 5V/10A power supply is shown below.

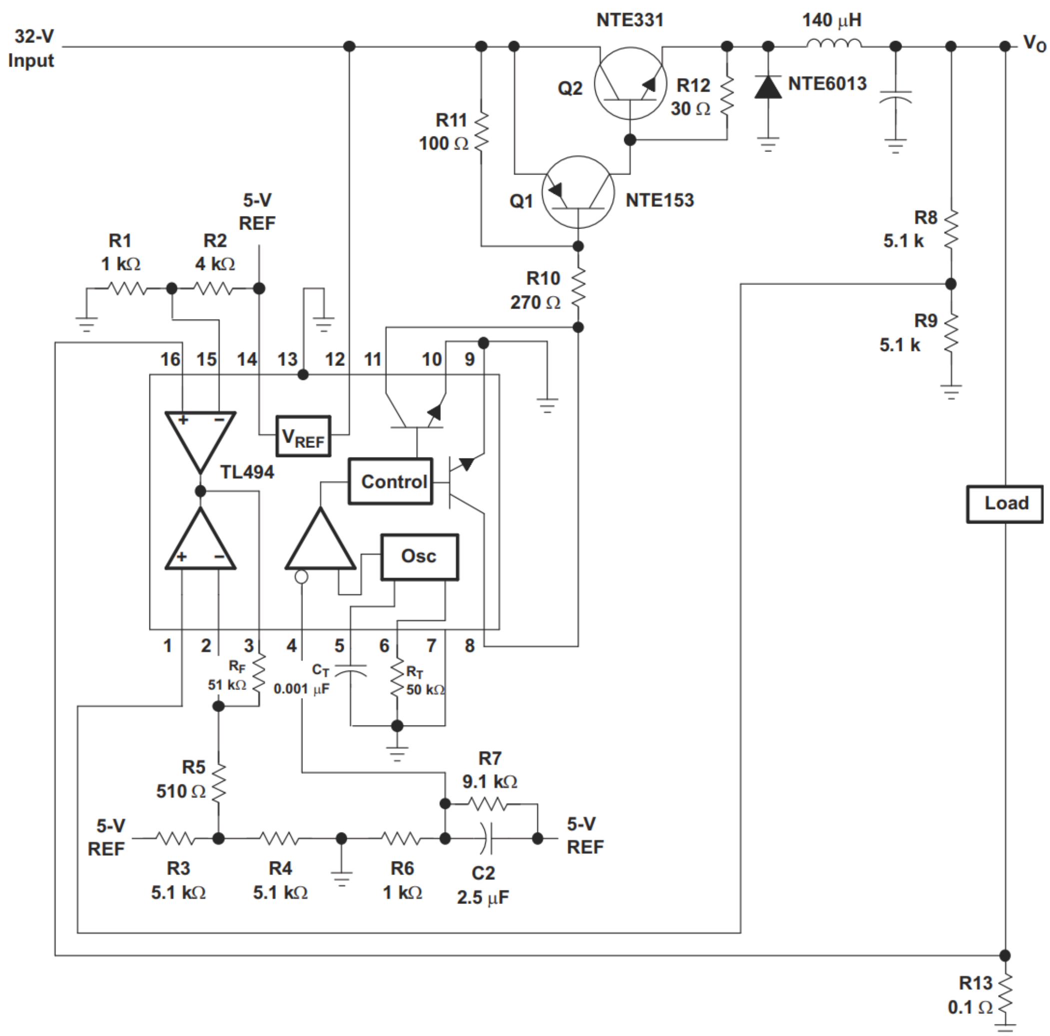


Figure 7: Example Buck Converter with TL494 PWM Controller

2.5 Software Design

2.5.1 Controlling the robot

The core component of our project will control the robot based on computer motion commands; the robot will obey each motion command using different states to give control to the different position and motion subsystems to achieve its goal. The possible states are as follows.

- **Start:** At the beginning of boot up for the ESP32 the startup mode sets up the internal wifi module which can connect to the computer.
- **Startup Wifi:** In case initial startup for wifi driver fails this mode will repeatedly attempt to restart the wifi module until it is ready.
- **Connect:** In this mode the ESP32 is waiting for the computer to send it an agreed upon test string to verify that the robot can receive commands.
- **Listening:** During listening mode the robot informs the computer it is listening and then fills up a linked list of movement commands prefaced by a motion start command fed in by the connected computer until given a motion done command. Can also be ended by the computer sending a no command signal.
- **Sense:** During sensing mode the position subsystem reads the output from each IR sensor and the ToF sensor to verify if the next motion position is clear for the robot.
- **Move:** During the movement mode the motion subsystem is given a vector and a distance to travel and does the appropriate calculations passing it into the motor driver
- **Halt:** Instantly stops all motor motion updating the movement list with the motions left in the previous instruction using data from the imu and motor encoders.
- **Verify:** The motion and position subsystem work together with the motor encoders, imu, to determine the xy motion of the robot and if the robot needs to do extra motions to arrive at the correction position
- **Corrective Motion:** Control subsystem calculates what the next motion in the list should be based on the measured motion error in the previous motion
- **Ride:** Control subsystem waits in front of an elevator until the door opens and enters it. If already the inside system does not do anything. Uses barometric sensor to get z motion

- **Exit:** Robot adds movement commands to exit the elevator before performing the rest of the computer given commands.
- **Done:** Informs the computer that it completed its movement list and is waiting at the desired room

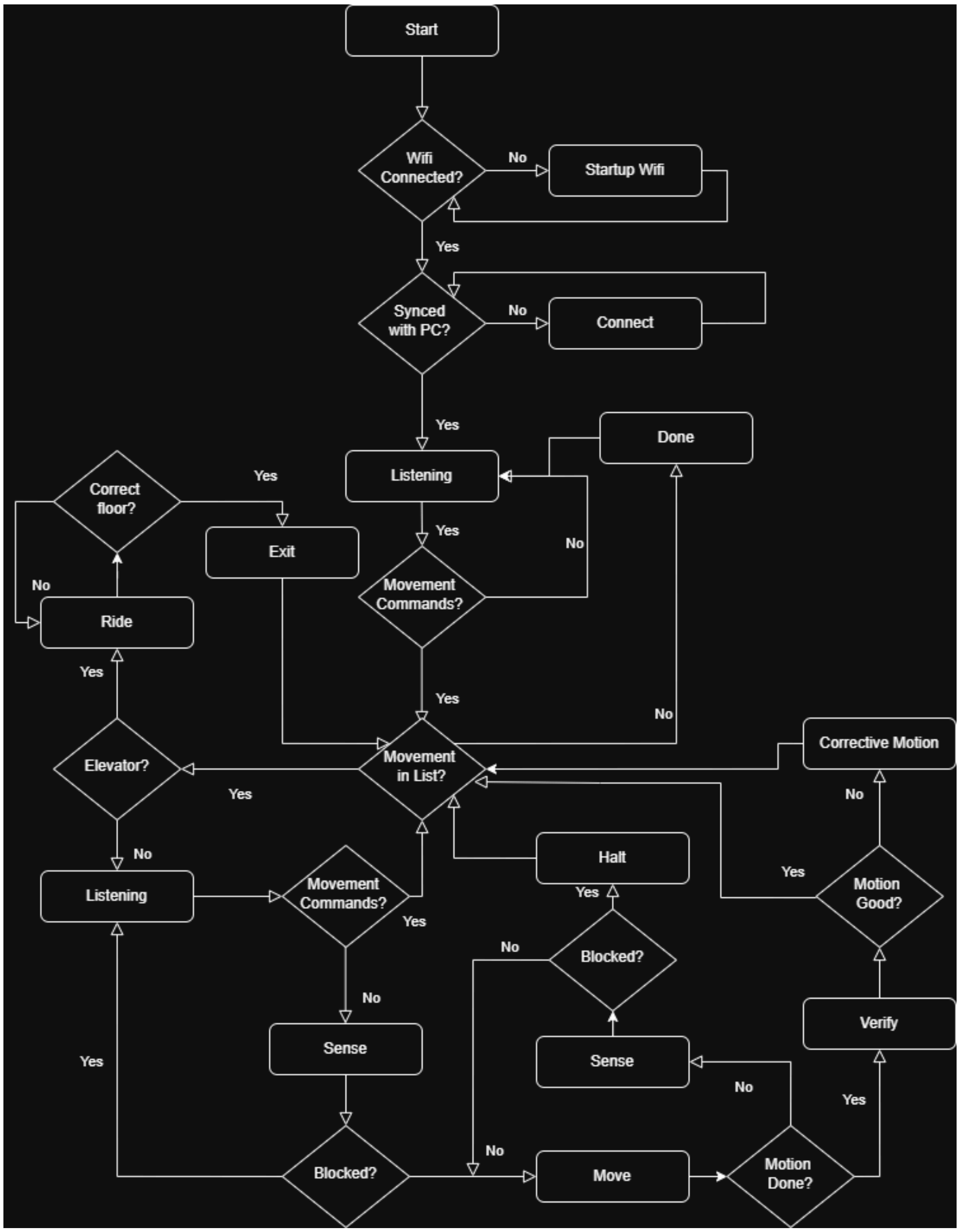


Figure 8: FSM of software inside of the robot

2.5.2 The Computer

The core component of our project will control the robot based on computer motion commands; the robot will obey each motion command using different states to give control to the different position and motion subsystems to achieve its goal. The possible states are as follows.

- **Start:** The startup of the computer application starts up the GUI and the map for the environment as a static prefixed graph.
- **Connecting:** The pc attempts to repeatedly connect to the ESP32
- **User Input:** Display inputs allow the user to input classroom locations
- **Path Calculating:** The computer operates a A* search on a graph for the 2D map to find the fastest path between the robots starting position and ideal ending takes into account rotating the robot to turn
- **Give Motions:** Computer waits until the ESP32 device is listening and gives a list of motions for it to complete to arrive at its destination
- **Cancel**: In the middle of an operation if the robot is blocked in the current plan is to either give the operator a chance for the robot to get unblocked or cancel the trip returning back to the home station.
- **Go Home**: Calculates a path from the robots current location to a prefixed home point

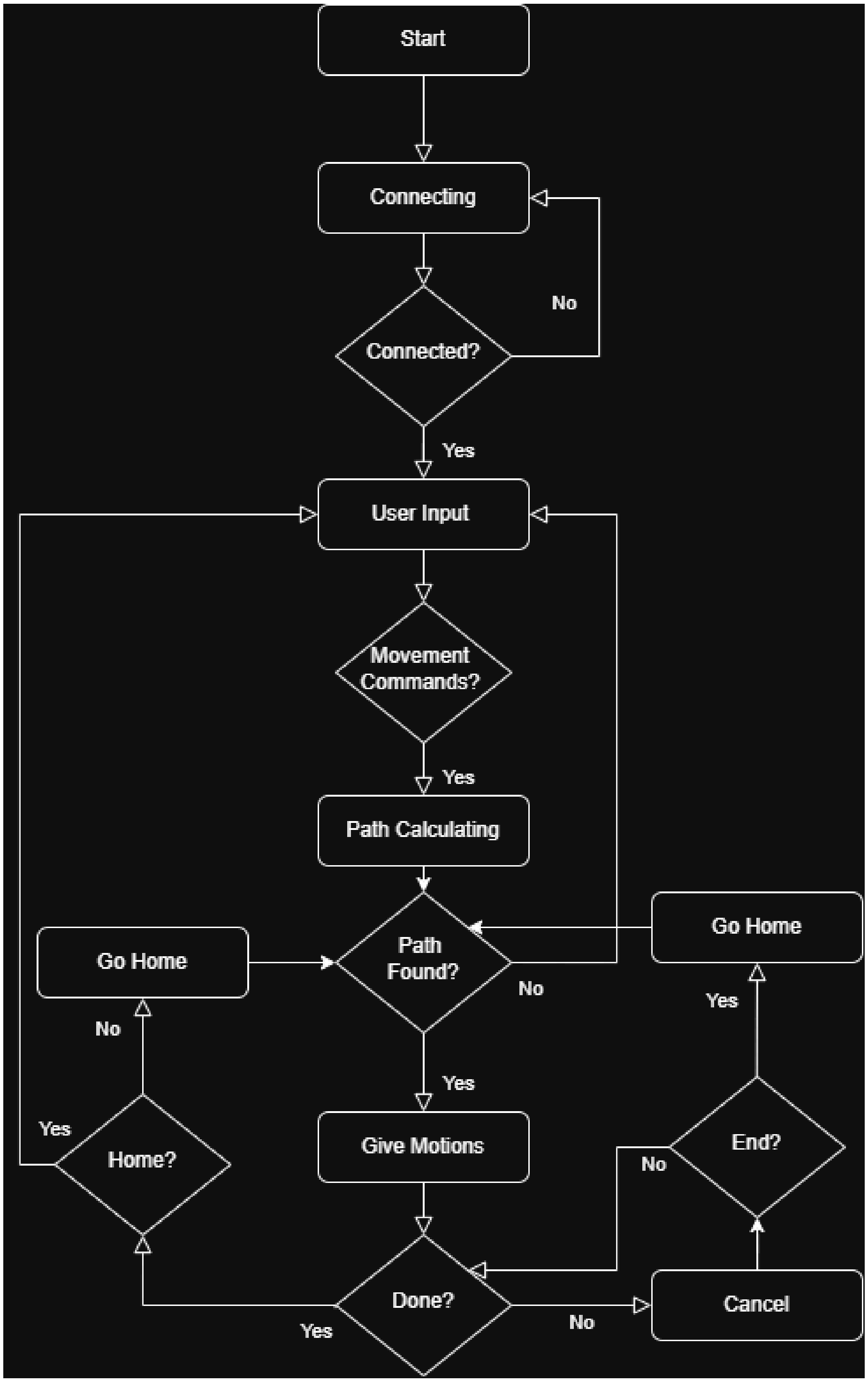


Figure 9: FSM of software on the computer to control the robot

2.5.2 A* Search

To get the most efficient pathway between the two points the A* search is applied. The way a A* search works is a depth first graph search choosing nodes with the lowest euclidean distance to the destination. To do this search we will be breaking down the floor plan of the ECEB into feet by feet grids that robots can easily fit within and connecting them to their adjacent grids in a graph-like manner. Walls and common obstacles (chairs desks) will be preloaded into the map removing graph connections where the robot is unable to traverse through. Below is a figure showing an example of a A* search algorithm [16].

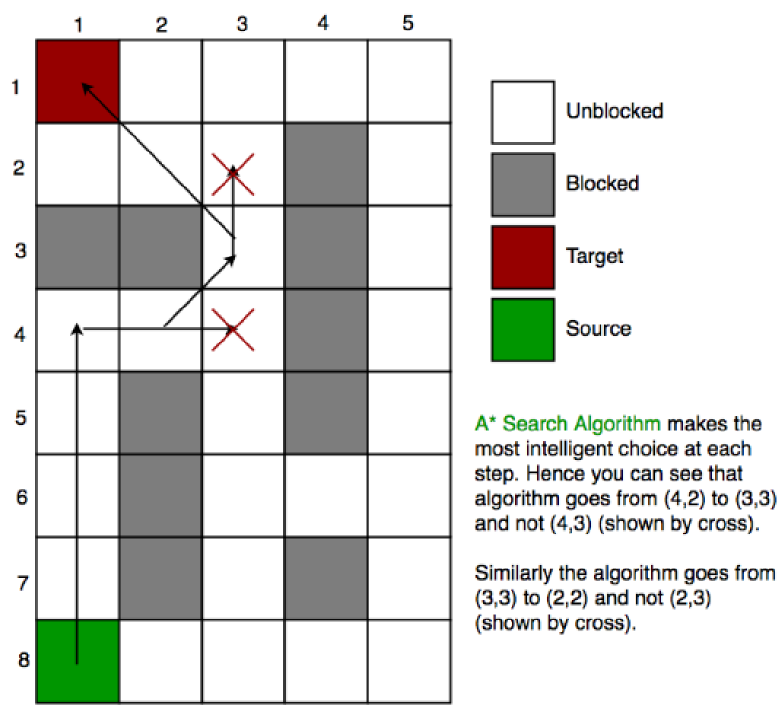


Figure 10: Example of A search algorithm*

2.6 Tolerance Analysis

2.6.1 IR Sensor

The IR sensor system is based on detecting reflected light from an object. The color and material of the object can influence the amount of reflection resulting in different depth capabilities for the sensor. Figure 3 shows experimental data done on an IR sensor pulsing out a

37.5 kHz signal. It can be seen that to account for this tolerance we will only be able to detect obstacles ~5cm away from the robot and should calibrate our sensors accordingly.

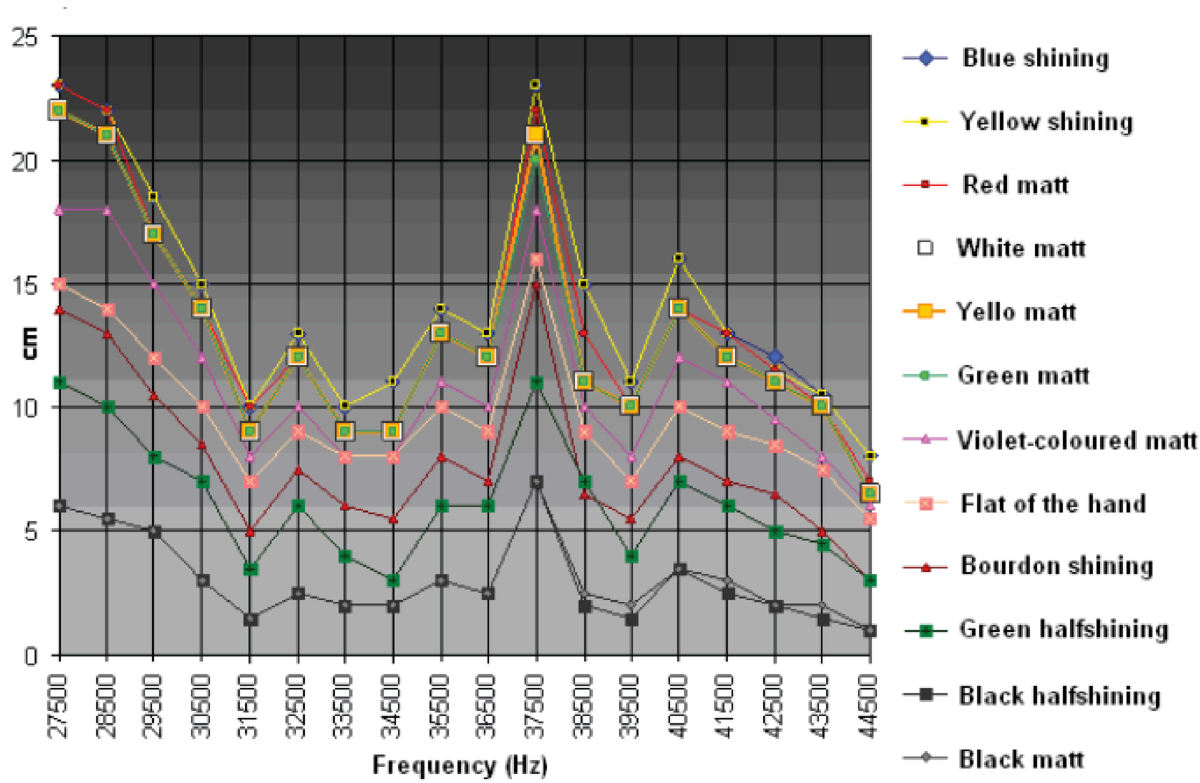


Figure 11: Experimental data on IR sensor detection capabilities based on the frequency [1].

2.6.2 Motor Torque

To move the robot the motors will need to exert enough force to overcome the friction force allowing it to drive. The motor speed is 600 RPM which will be connected to two 6 inch diameter wheels. Using equation 1 we can calculate our maximum velocity under no load is 15.70 ft/s. This speed is incredibly fast so we will need to step it down using a gear box ratio (with help of the machine shop).

$$v = \frac{RPM}{60} * \pi * D = 15.70 \frac{ft}{s} \tag{1}$$

We want the robot to move at walking speed for the safety of our surroundings and to give sensors enough time to build a map of the surroundings (~4 ft/s). Equation 2 shows the relation for slowing down the RPM using the gear ratio (N) and Equation 3 shows the increase in torque (Eta is the efficiency of the gear box system which is a mechanical value ~ 90-98% efficient).

$$RPM' = RPM/N \quad (2)$$

$$\tau' = \tau N \eta \quad (3)$$

Using Equations 2 and 1 to achieve a speed of 4ft/s we need a gear ratio N = 3.925. The motors initial rated torque is .25 kg/cm. Plugging this into Equation 3 gives us an expected maximum torque force from both wheels at around 0.98125 kg/cm (65.94 lb/ft). This is sufficiently high which means using these motors is operable.

2.6.3 Voltage Control

Different subsystems will need to run on different voltages, motion(atleast 12V), control/sensors(3.3V), raspberry pi (5V). The buck converter will be used to convert the 12 Volts signal into 5 Volts. Figure 4 shows simulation output of a 12 Volt input buck converter circuit successfully creating 5 volts output current at 3.5 Amps [2].

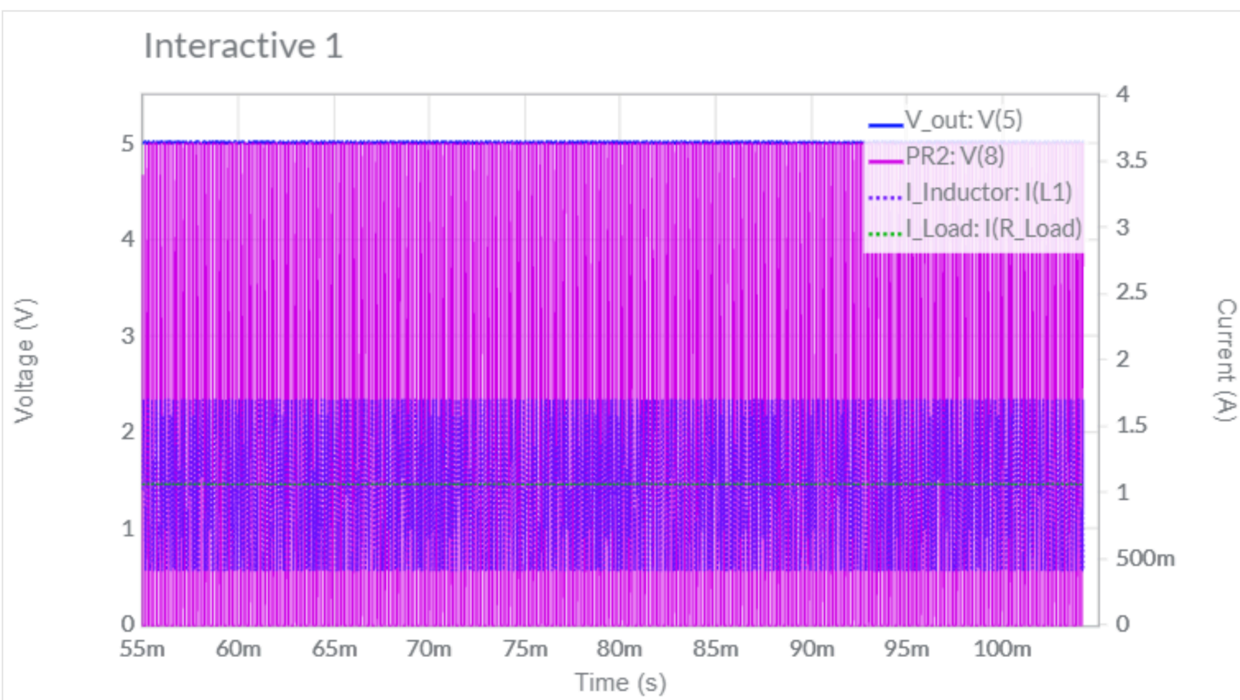


Figure 12: Simulation of Buck converter 12V to 5V [2].

The output from the Buck converter will also be fed into a 3.3 Volt power regulator. From the data sheets the output voltage has an expected error of 3% which is well within the margins of the ESP32 microcontroller and accelerometers safety limits. Figure 5 shows experimental data for Voltage in versus Voltage out for low voltage values [3].

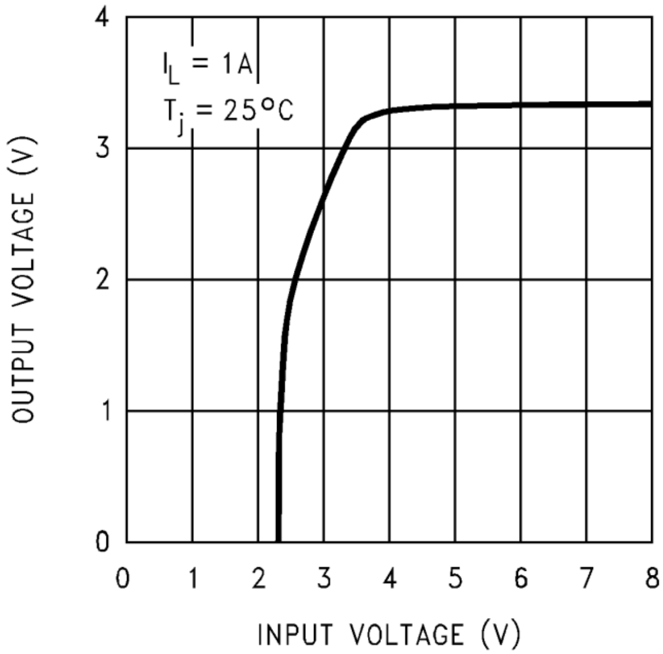


Figure 13: Experimental data of Voltage output for Voltage regulator [3]

3. Cost Analysis and Schedule

3.1 Costs

The total cost of all parts listed in Figure 5 below sums up to $179.01. After factoring in extra fees such as shipping/handling and taxes, the holistic total for ordering parts is **$203.98**.

For labor, each lab member will be paid at a rate of **$40/hr**, so after factoring in the multiplier and total hours each member is expected to put in to complete the project, we arrive at a total of $40/hr * 2.5 * 100 hours = **$10000 per member**. The total labor cost of our 3 member team is **$30,000**.

We are also commissioning the ECE Machine Shop for labor in building the physical robotic structure and foundation. The estimated quoted time of labor from the machine shop is 3 working days of labor, and using a quoted $56.12/hr salary we get **$1262.70** for the estimated total machine shop labor cost. The total cost including all labor, parts and materials comes out to **$31466.68**.

Description	Manufacturer	Part #	Cost Per Unit	Qty.	Total Cost
12V Hall Encoder DC Geared Motor	Bemonoc	B07GNDG2NC	$20.99	2	$41.98
ESP32-S3 VROOM Microcontroller	Espressif Systems	ESP32-S3-WROOM-1-N16R2	$5.92	1	$5.92
ESP32-S3 Dev Kit	Espressif Systems	ESP32-S3-DEVKITC-1-N8R8	$15.00	1	$15.00
Motor Driver Devkit	Texas Instruments	DRV8871	$7.50	1	$7.50
Motor Driver	Texas Instruments	DRV8871	$3.82	2	$3.82
3.3 V Regulator	Texas Instruments	LM3940IT/LM350T	$1.71	1	$1.71
PWM Controller	Texas Instruments	TL494CN	$0.51	1	$0.51
Pressure Sensor Devkit	Bosch Sensortec	BMP585	$14.95	1	$14.95
Pressure Sensor	Bosch Sensortec	BMP585	$4.32	1	$4.32
ToF Sensor Devkit	Silicon Labs	SI1151-AB00-GM	$9.95	1	$9.95
ToF Sensor	Silicon Labs	SI1151-AB00-GM	$3.93	1	$3.93
IMU Devkit	STMicroelectronics	LSM6DSO32	$12.5	1	$12.5
IMU	STMicroelectronics	LSM6DSOTR	$3.51	1	$3.51
IR LED	EverLight	IR333-A	$0.32	4	$1.28
Amplifier for IR sensing	Texas Instruments	LM358	$1.00	2	$2.00
SILICON PHOTODIODE	Excelitas Technologies	VTP1188SH	$3.58	4	$14.32
Battery	CITYORK	XZ02	$21.99	1	$21.99
Simple Circuit Components	-	Resistors Capacitors BJT	$10	1	$10

Total Cost:	**$179.01**

Figure 14: Bill of materials and total costs associated with parts.

3.1 Schedule

Week	Ethan	Johan	Nikhil
Feb. 23rd - Mar. 1st	Talk to the machine shop, review sketch design and go over dimensional goals. Work on a general 3d model of ideal design. Work on the design document.	Research motors and suitable gear ratios for wheel size, speed and necessary torque to carry weighted load. Add selected motor to design doc. Work on the design document.	Begin working on designing schematics for pcb and MCU. Work on the design document. Submit orders for initial parts like MCU.
Mar. 2nd - Mar. 8th	Confirm parts and assembly process with the machine shop. Begin to test components in the lab after delivery. Help organize pickups for ECE supply center and machine shop equipment.	Research sensor implementation for IR detection, and read up on wireless communication and transmission for our robot from the main computer.	Continue refining pcb schematics. Help go into the lab and test components when they are delivered. Compile order of more parts (breakout boards for initial phase).
Mar. 9th - Mar. 15th	Start writing backbone software portions necessary for navigation and map tiling.	Test motors and other hardware, compile all data in an organized lab documentation.	Work on finalized pcb schematic, make any necessary changes from test results or design adjustments. Place pcb order for the week.
Mar. 16th - Mar. 22nd	Spring Break: Update individual lab notebook entries.		
Mar. 23rd - Mar. 29th	Continue developing software for path aligning algorithms and map floor layout in tiles. Debug program and test with physical hardware.	Begin to implement wireless communication protocol between central computer and robotic platform. Help with software and hardware testing.	Work in the lab on putting together breakout board tests, connecting all parts and soldering if applicable. Place pcb order for the week if necessary.
Mar. 30th - Apr. 5th	Work on integrating and interfacing hardware with	Work on putting all hardware onto the physical	Work on integrating all parts of board assembly,

	software, help put together physical projects.	robot and run finalized tests for sensor subsystem modules.	complete wiring around enclosures and physical assembly.
Apr. 6th - Apr. 12th	Debug any issues with software interface, hardware connections, communication protocols and any other problems with subsystems.		
Apr. 13th - Apr. 19th	Work on the final paper, compile all test results and lab documentation. Prepare for the final demo and presentation. Finish individual lab notebook entries.		
Apr. 20th - Apr. 26th	Mock Demo, final tests and validation. Finalize assembly of system and all working parts. Finish integration tests and documentation.		
Apr. 27th - May 3rd	FINAL DEMO + PRESENTATIONS		

Figure 15: Schedule for expected sequence of design and construction work.

4. Conclusion

4.1 Accomplishments

This project should accomplish several goals revolving around its physical capabilities, and provide an affordable solution to autonomous room service style robots that can make deliveries autonomously across a building. Our project aims to develop an indoor-operated service robot capable of navigating building hallways, delivering items to specific designated rooms, and interfacing with obstacles and elevators to enable improved operation. Our proposed system above will integrate our own custom-designed PCB, which connects our various sensors that track live positioning and conduct proximity sensing for safe navigation in dynamic indoor environments. By incorporating localization techniques and mapping pre-determined paths alongside our sensors and odometry, the robot will be able to follow routes that lead to room destinations while reacting to unexpected obstacles such as pedestrians or furniture. Successful completion of this project will accomplish a demonstration of a fully integrated robotics system with functioning hardware, sensing, controls, and autonomous decision-making suitable for applications such as hotel room service, hospital delivery assistance, or campus logistics.

4.2 Uncertainties

Some of the uncertainties regarding our project during the proposal and design process so far include reliable connection throughout the elevator, consistent obstacle detection, and other design choices that may change going forward during testing and implementation phases. Reliable elevator interaction presents a key challenge, including detecting elevator arrival, ensuring safe entry and exit, and coordinating room or floor selection. Some of our more ambitious goals relating to these algorithms or extensive features may be limited due to our available time and resources throughout a single semester, and design deadlines that conflict with our individual workloads may pose challenges in the future during the building process.

4.3 Ethical considerations

As responsible engineers, we recognize that ethical considerations associated with developing autonomous robotic systems that interact with people in public spaces are important to address and keep in mind when making design decisions for our project. Safety is a primary concern that is addressed through features such as obstacle detection, where we will implement emergency stop mechanisms and fail-safe behaviors to minimize risk of injury or property damage, adhering to the second code in the IEEE Code of Ethics. This also coincides with the University of Illinois policies with university facility use, as we plan to demo our completed and functioning project in the ECE building on campus and utilize various university resources in order to achieve this.

Privacy considerations are also an important consideration in ethics, also highlighted in the IEEE Code of Ethics. Any sensing data used for navigation will avoid unnecessary collection or storage of personally identifiable data. If any vision or environmental related sensing is employed, it will be limited to navigation purposes only and not used for surveillance of any sorts.

Potential misuse must also be considered to avoid any possible ethical breaches by users. Autonomous delivery robots could be repurposed in ways that create safety hazards or privacy risks. To mitigate this, the system will include clearly defined documentation clarifying intended use for service needs, incorporate speed limits, and limited authentication for who has permission to access controls. We will also adhere to standard lab safety documents [15] and training when dealing with potentially volatile electronic components such as a battery.

Societally, autonomous service robots may also have potential economic impacts, including increased efficiency in hospitality settings, but also possible workforce displacement which can be negatively perceived in terms of ethics. Our project focuses on augmenting and aiding human service roles rather than replacing them completely, especially since many facets of proper operation still require human oversight or control, demonstrating how technology can assist people with repetitive logistical tasks.

4.4 Future work

In the future, we can implement improvements beyond the scope of our senior design project. Some upgrades we have in mind include more advanced localization techniques and enhanced human to robot interaction such as voice interfaces or mobile app integration. There is always room to improve on or expand on autonomous mapping algorithms as well as efficient rerouting calculations. As this product will support logistical tasks involving humans, interactions can be later improved by developing more intuitive user interfaces for control, or software/mobile app connections that allow remote access to our robot. We would also love to implement additional sensor functions and machine learning methods which we believe could improve navigation reliability in crowded or unfamiliar environments. On top of those features, some general performance enhancements include further expanding battery efficiency, payload/weight capacity, and long-term reliability testing which would all be necessary in the case of any plans for potential commercial deployment. Ultimately, further developments and advancements could enable broader applications in more environments including hotels, hospitals, office buildings, and campus facilities.

References

[1] *I. Matijevics, "Infrared Sensors Microcontroller Interface System for Mobile Robots," 2007 5th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia & Montenegro, 2007, pp. 177-181, doi: 10.1109/SISY.2007.4342647..*

[2] NI Multisim Live, "12V to 5V DC Converter," Accessed: Feb. 13, 2026. Available: www.multisim.com/content/ExAz7JQzFgNMpcjwbcCgGo/12v-to-5v-dc-converter/.

[3] Texas Instruments, LM3940IMPX-3.3 Low Dropout Voltage Regulator Datasheet, available: https://www.ti.com/lit/ds/symlink/lm3940.pdf.

[4] Espressif Systems, *ESP32-WROOM-32E Datasheet*, available: https://documentation.espressif.com/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[5] Texas Instruments, LMD18245 Motor Driver Datasheet, available: https://www.ti.com/lit/gpn/lmd18245.

[6] STMicroelectronics, *LIS2DH12TR Accelerometer Datasheet*, available: https://www.st.com/content/ccc/resource/technical/document/datasheet/12/c0/5c/36/b9/58/46/f2/DM00091513.pdf/files/DM00091513.pdf/jcr:content/translations/en.DM00091513.pdf

[7] Elonics.org, "IR Proximity Sensor / Obstacle Detector Circuit Using LM358 OPAMP," Accessed: Feb. 13, 2026. Available: https://elonics.org/infrared-ir-proximity-obstacle-sensor-using-lm-358/

[8] ETF Trends, "Coming Labor Crisis: America's Robot Revolution," Available: https://www.etftrends.com/disruptive-technology-content-hub/coming-labor-crisis-americas-robot-revolution/

[9] Hoover Institution, "Robotics: Build, Train, Electronic Workforce," Available: https://www.hoover.org/research/robotics-build-train-electronic-workforce

[10] International Federation of Robotics, "Service Robots See Global Growth Boom," Available: https://ifr.org/ifr-press-releases/news/service-robots-see-global-growth-boom

[11] STMicroelectronics. LSM6DSO IMU Datasheet. STMicroelectronics, https://www.st.com/resource/en/datasheet/lsm6dso.pdf.

[12] Bosch Sensortec. BMP585 Barometric Gauge Datasheet. DigiKey, https://www.digikey.com/en/products/detail/bosch-sensortec/BMP585/19522589.

[13] Texas Instruments. TL494 PWM Controller Datasheet. DigiKey (if applicable) / Texas Instruments, https://www.ti.com/lit/ds/symlink/tl494.pdf.

[14] Silicon Labs. SI1151 ToF Detector Datasheet. DigiKey, https://www.digikey.com/en/products/detail/silicon-labs/SI1151-AB00-GM/9842913.

[15] University of Illinois Urbana-Champaign. Laboratory Safety Guide. https://drs.illinois.edu/site-documents/LaboratorySafetyGuide.pdf.

[16] "A* Search Algorithms." GeeksforGeeks, https://www.geeksforgeeks.org/dsa/a-search-algorithm/.