Lazy Bob

Omar Areiqat, CompE, Vedatman S. Basu, CompE, Redwan Rahman, CompE, and Jina Song, EE

***Abstract*—People with disabilities need ways to bring food from one side of the dining table to the other. The invention of the Lazy Susan or revolving tray was one solution which accomplished this, but its only application is for round tables designed to host it. By having a mobile coaster bring individual items around the table, we can accommodate any table shape. The proposed solution, Lazy Bob, plans to do just that by having a moving coaster go towards a destination once prompted by the user. By keeping track of the caller remote's relative position and checking for cliffs and collisions, the Lazy Bob system will bring items to the user.**

I. INTRODUCTION

Some people struggle with simple actions in life, such as passing dishes on a large table. Lazy Susans are partially effective, but can only be used on round tables where the user is required to be able bodied enough to reach beyond the edge of the revolving tray. What if there was a small, portable device that can move dishes across any table of arbitrary shape?

A. Significance

We as a society are ever more becoming aware of the immense burden that one has to carry when dealing with a physical disability. Impairments can make tasks which require mobility, wide motions, and fine motor skills more difficult. Lazy Bob intends to improve the daily lives of those who are physically challenged. In addition, this project will also increase general convenience even for those who are not physically impaired as many technologies designed to aid the disabled can also provide increased comfort and convenience to able-bodied people as well.

B. Context and Survey of Similar Solutions

From our research, there is currently no existing product for a moving coaster that travels on top of a table as a delivery system. However, there are solutions which have similar subsystems which can be used as proof of concept. Delivery robots designed to work on floors, such as the AirPorter [1] and the Starship Delivery Robot [2], showcase how an autonomous driving device is capable of avoiding obstacles. Similarly, robot vacuum cleaners are capable of both collision avoidance and cliff detection. Lastly, the Smart Coaster project by SDP20 Team 16 is similar in that it has wireless communication between the coasters and the hub, as well as having rechargeable batteries. However, what makes Lazy Bob different is that it will not only have autonomous driving capabilities and be able to be called to a specific destination, it also takes size and cost into consideration. All the similar solutions previously mentioned are not capable of efficiently delivering food on top of a table, which is what Lazy Bob will achieve.

The table below outlines the similarities and differences of all the partial solutions against the proposed features of a Lazy Bob. While the autonomous driving devices such as AirPorter and Starship have the most similarities to what Lazy Bob requires, their sizes would not be feasible in a regular household dining room, as well as the fact that they are not commercially available and cost tens of thousands of dollars to manufacture.

	Movement	Transport	Avoid obstacles	Avoid cliff	Size	Call to location	Cost
AirPorter							N/A
Starship							N/A
Robot Vacuums							
Smart Coaster							N/A
Lazy Bob							

Table 1: Comparison of Similar Solutions

C. Societal Impacts

The help that Lazy Bob provides around the dining table could help reduce the burden of eating with other people due to dishes being more spread out. This solution will aid those who are disabled, as well as take a general inconvenience out of every day work.

On the other hand, Lazy Bob may unintentionally add clutter to the household. Although our goal is to make Lazy Bob small enough to be unintrusive, it could be unnecessary for some households, which could end up doing more harm than good by taking up more space on the dining room table.

D. Goals, Specifications and Testing Plan

The specifications focus on the moving coaster portion of the system, such as its functions and how fast it should move . The majority of the corresponding testing plans for these specifications involve human observations as it is often a binary: either it works, or it doesn't. The engineering challenges associated with this project require qualitative specifications, as failing to avoid obstacles even 5% of the time would pose a risk due to the nature of the project. The specifications were split into two subgroups: functional and design. The functional specifications are what is required for Lazy Bob to operate successfully.

Functional Specification	Corresponding Testing Plan
Mostly full container of liquid doesn't spill when moving	Place an 80% full glass/bowl of water and visibly check that the moving coaster doesn't spill
Transport items without the items falling off	Put objects on moving coaster and watch if object falls
Detect the edges of the table and avoid falling off	Watch if it falls off when Caller is away from the table
Avoid objects in its path and arrive at its destination	Put objects in path and check if it bumps into object
Arrive at the Caller to an accuracy of 10 cm from the edge of the coaster	Use a tape measure to check the distance from destination

Support up to 3 lbs.	Place an object that weighs 3 lbs and check if system is stable while moving

Table 2: Functional Specifications and Testing Plan

The design specifications are additional features that would enhance Lazy Bob and therefore would ideally like to pursue, but may not be possible within the duration of the provided time frame.

Design Specification	Corresponding Testing Plan
System supports at least 2 callers at the same time	Press both callers and see if the coaster follows first-in, first-out sequence
Coaster diameter is within 30 cm	Use tape measure
Be able to use the moving coaster regardless of table shape	Find and put on irregular shaped tables and check if they fall off

Table 3: Design Specifications and Testing Plan

II. DESIGN

A. Overview

Lazy Bob is a moving coaster system that has the user initiate contact using a Caller. This Caller will send out a signal to the moving coaster for it to move towards the Caller. Lazy Bob will be moving using mecanum wheels and brushed DC motors; as it travels, it will sense its environment using an ultrasonic sensor as the forward sensor, so that it can see if it is about to collide with another object; two ToF sensors are angled downwards to check if the coaster is about to fall off the table. Lazy Bob can also find where other parts of the systems are relative to one another through the ultra-wideband sensors so that the moving coaster has a clear destination to head towards when moving around the table. Lazy Bob is also battery powered so that it can freely move around the table in a direction of the user's choosing. The Callers are used to notify and direct the moving coaster towards the Caller.

B. Hardware Block Diagram

The hardware block diagram shown in Figure 1 displays which components would need to be connected physically or wirelessly using a solid or dashed line, respectively. The blocks in the hardware show the subsystems that will be required inside the moving coaster, such as motion, power, sensors, and positioning. The communication will occur wirelessly between the moving Coaster, Caller(s), and the hub, with the hub acting as a command center (or the "brains") where all the computations will occur; more specifically, the microcontrollers in the moving coasters and the Callers will communicate with the client/server system through Bluetooth. The Caller block will contain a method of user input, as well as a method of wireless communication so that it will be able to communicate the user input to the moving coaster.

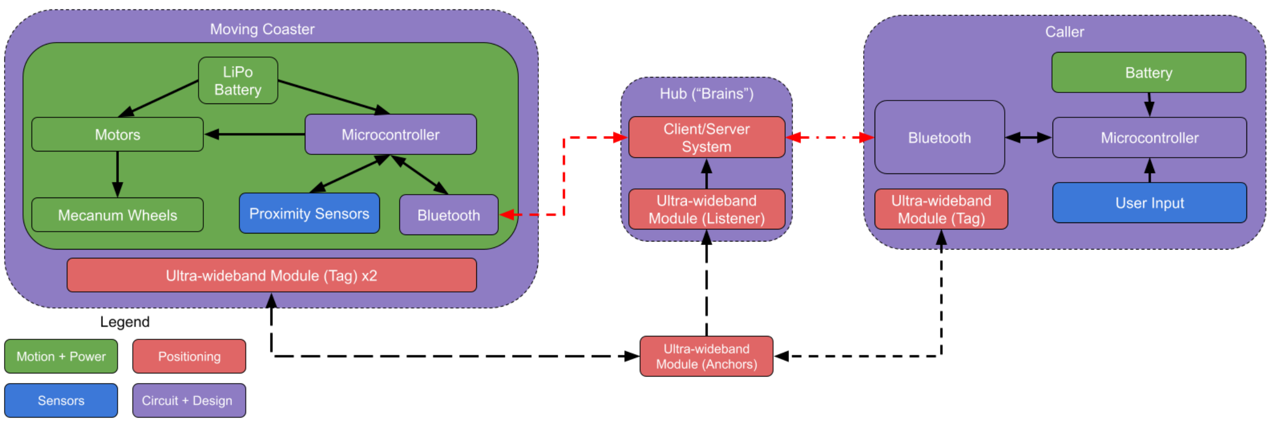

Figure 1: Hardware block diagram shows how all hardware components are connected to one another

C. Software Block Diagram

The software block diagram, shown in Figure 2, demonstrates the path that Lazy Bob system follows. Everything starts when the Caller notifies the client/server system that the moving coaster should move towards the Caller. The hub will gather the relative location of all relevant systems and tell the moving coaster which direction to move. While the coaster is moving, the ultrasonic sensors mounted on the coaster are constantly bouncing signals in order to search its proximity for potential objects or edges in its path. If it detects either, Lazy Bob will reroute so that it can eventually path to the Caller, and stop once the coaster arrives at the Caller.

The rerouting algorithm works by brute force checking that there are no obvious obstructions in the way of the coaster such as a cliff or object. If there is an obstruction, the coaster will stop, back to provide clearance, check that it has an unobstructed side to turn to. Make the rotation to the unobstructed side and then move forward a preset amount before reversing the rotation and proceeding forward with intent to repeat if the an obstruction still remains. Currently, we are using a heuristic method, but will be shifting to an optimized traversal algorithm.

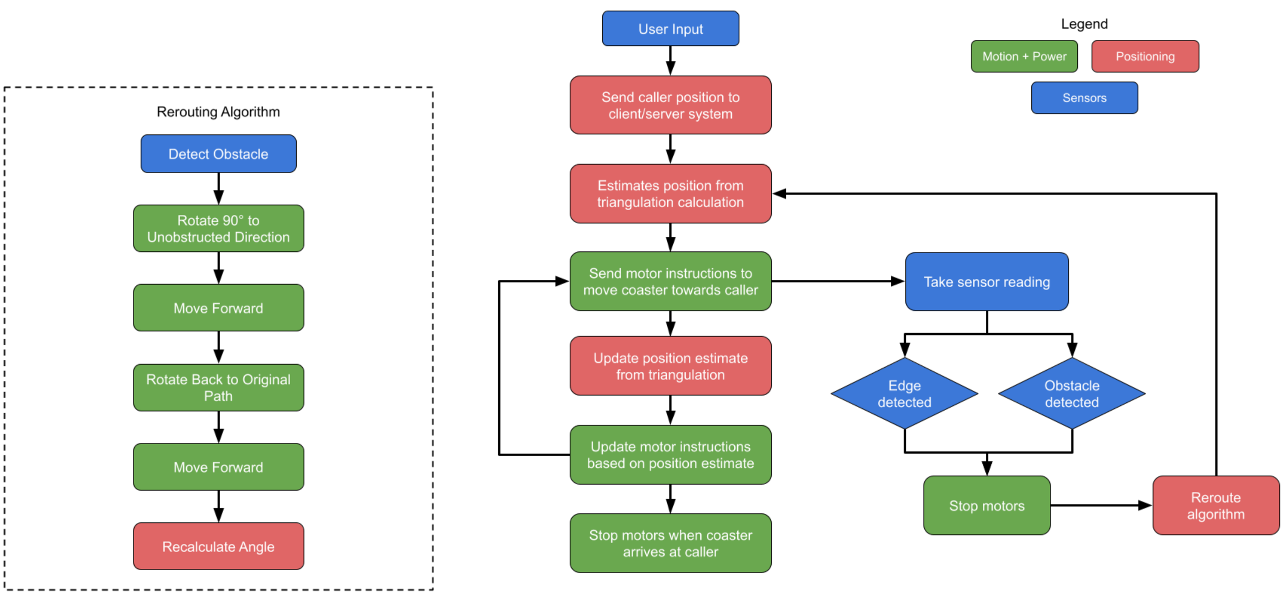

Figure 2: Software block diagram that shows the flow of data as the system works to move the moving coaster to the Caller

III. THE PROTOTYPE

A. Prototype Overview

The current prototype implements most of the systems together at a rudimentary level. Starting with the positioning subsystem, 2 white plastic panels can be seen on the right side of Figure 3 on opposite sides of the coaster. Those panels are used to connect the ultra-wideband module tags to the coaster. Along the perimeter of the working surface, there are 4 more similar modules that act as anchors. The position of the coaster relative to these anchors is obtained using time of flight signals sent in the ultra-wideband from these sensors and put in a coordinate grid centered around the anchors. These sensors take this data and returns the coordinates of the tags on the coaster and the Caller. Since both objects are relative to the anchors, we can use these coordinates to find the angle and range between the Caller and the coaster, and send that information to the motion subsystem. A Raspberry Pi works as the brains of the system by receiving the positioning information, then calculating the angle and distance that the coaster has to move, and finally sending the appropriate signals to the coaster via Bluetooth. The Pi is connected to both the coaster and the Caller such that when the Caller sends the correct Bluetooth signal, the Pi will then send appropriate Bluetooth signals to the coaster to start moving. In the center of the coaster sits the PCB used for the motion subsystem and all the power requirements for all connected parts. The Pico W microcontroller on the PCB controls the logic for the motors. When it receives the correct Bluetooth signal, it will move or stop depending on the signal sent. The PCB used for the motion subsystem also features a voltage regulator due to the fact that the components have varying voltage requirements. The sensors are also connected to this Pico W, which obtains the data from the sensors and decodes it to prevent any collisions or falling off.

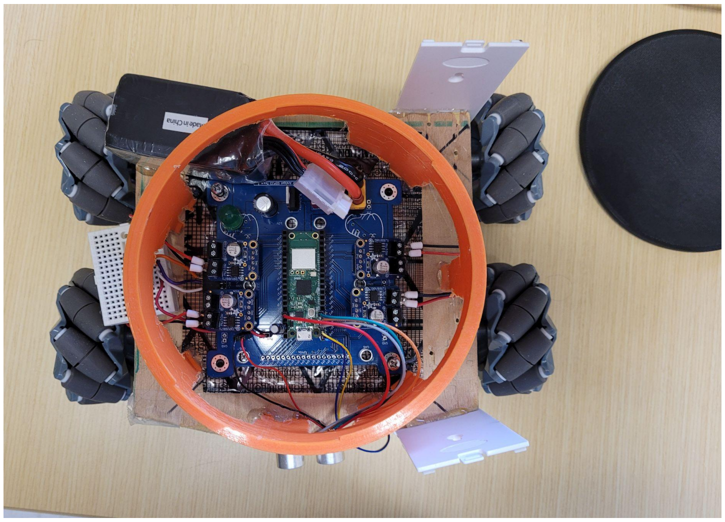

Figure 3: Top down view of the coaster. The black circle off to the right side goes on top of the orange walls to make a lid to place objects on top of. The battery used is visible on the top left of the coaster. The PCB in the center of the coaster is used to control movement of the coaster and to receive the sensor data.

B. List of Hardware and Software

The motion subsystem consists of mecanum wheels, brushed DC motors, DRV8871 motor drivers, and a Pico W responsible for the logic. Mecanum wheels, combined with brushed DC motors powered by motor drivers, enable the coaster's movement. The Pico W not only manages the motion logic but also facilitates Bluetooth functionality for communication between subsystems and powers the motors and controllers as well as sensor data.

The positioning subsystem employs the MDEK1001 kit, which includes 12 monoliths for positioning purposes. Out of these, three to four serve as anchors, and three more as tags and one more as a listener to relay the information to the Pi. The remaining units are currently unused. The sensor subsystem incorporates one ultrasonic sensor and 2 VL53L1X laser distance sensors connected to the Pico W, which in turn communicates with the Raspberry Pi to initiate appropriate actions.

Lastly, the coaster's body is constructed from scrap wood, ensuring resourceful utilization of materials. There is a raised platform to place objects on top of the coaster that we 3D printed.

For software, we coded all of the microcontrollers using Arduino code for both the motion and sensor subsystems. For the brains, we have some Python code running on a Raspberry Pi that reads in the data sent to the listener from the positioning subsystem and converts that to a range and angle. That is sent via Bluetooth to the microcontroller responsible for the motion.

C. Custom Hardware

The Caller PCB as shown in Figure 4 was designed by us to hold a Pico W microcontroller, a Bluetooth module and a button. To power the parts, we added a battery pack and some voltage regulators to give the desired voltage to all components. A better PCB will be designed for all components so that they can all fit in a 3D printed enclosure for all parts.

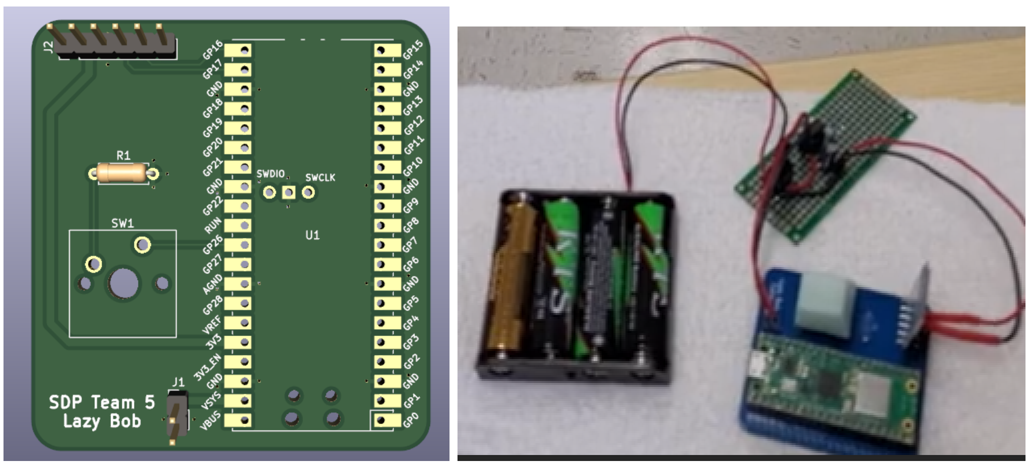

Figure 4: PCB for Caller. Left is the PCB render. Right is the built version. There is a protoboard connected to the PCB at the top of the right image due to the new Bluetooth module needing a different voltage as well as regulating the voltage for the microcontroller.

D. Prototype Functionality

The coaster can be connected via Bluetooth to either a phone or a computer. When it is connected to the phone, we have an app to send discrete movement options to test if the coaster moves as intended and if it actually stops when it detects an edge or a cliff. When connected to the Raspberry Pi, we can also hook up the other components and test if everything works together. The listener module from the positioning subsystem can connect to a serial port and send the position data and the Caller will also send a signal to a serial port notifying if the coaster should move to the Caller when the button is pressed. The computer runs its code and sends the Bluetooth signal to the coaster to move accordingly. The sensor subsystem is also active in this test meaning that all systems are working in tandem.

E. Prototype Performance

The motion and sensor subsystems function well together. When prompted, the Caller can send a signal to the computer, instructing the coaster to move towards it. The positioning system then calculates the distance and angle required for the coaster to travel. Based on this information, we send the appropriate signals (turn or move straight) for a specified duration.

However, there is a challenge in determining the correct duration for the turn signal. Although we have an angle to relay to the coaster, its orientation at a given time remains unknown, which leads the brain to send the rotate signal for either too long or too short of a period. Apart from this issue, all other features are performing as intended.

IV. CONCLUSION

ACKNOWLEDGMENT

References

[1] "Delivery: (주)원익로보틱스 Wonik Robotics," *Wonik Robotics*. [Online]. Available: https://www.wonikrobotics.com/delivery. [Accessed: 30-Oct-2022].

[2] Newsdesk and I. Krug, "Starship robots may return in future under new dining contractor," *The Pitt News*, 23-Sep-2020. [Online]. Available: https://pittnews.com/article/160259/news/starship-robots-may-return-in-future-under-new-dining-contractor/. [Accessed: 30-Oct-2022].

Appendix

A. *Design Alternatives*

We had multiple alternatives to different aspects of our current design. The one we put most thought into, which is also the hardest part of our project, is the accurate and precise location tracking of objects. We considered using a camera and markers on our objects, and utilizing image processing to keep track of everything. We also considered placing markers on the table itself, such as specific grid distribution markers, so that the moving coaster knows where in the grid it currently is, which simplifies the process in finding where other objects are relative to the moving coaster. These options are frankly simpler than the option we are now pursuing, but the reason we decided on the solution that we did is due to the fact that they don't allow for plug-n-play. Keeping the consumers in mind, there is a lot of setup involved, and it would be too much of a hassle for the end user, which is why we decided on a solution that does not need any pre-setup. This also allows for all of the location tracking to work relative to each other.

B. *Technical Standards*

We are using ultra-wideband modules (UWB) to communicate between embedded systems that are utilizing the RP2040 Pico W and Raspberry Pi 4 microcontrollers. C++ and Python are used to send instructions and process information between each system.

C. *Testing Methods*

Our goal for Lazy Bob is to efficiently navigate a mapped environment while adapting to dynamic changes within that space. To test this, we will call Lazy Bob to various locations and observe its ability to find the clearest route. We will also introduce obstacles periodically to assess how well Lazy Bob adjusts and identifies valid paths, if any exist. In an edge case test, we will create a scenario with no valid paths by placing a series of obstacles, expecting Lazy Bob to recognize the situation, and subsequently inform the user that no available path exists.

D. Project Expenditures

We diligently monitor all our expenditures using Google Sheets. Fortunately, we managed to source most of the necessary components from M5. However, due to exhausting our initial budget, we had to seek additional funds from the department to ensure we could acquire all the required parts for the Final Project Review (FPR).

Purchasing List		
Item Name	Supplier	Price $ (with tax and shipping)
UWB DIY Starter Kit	[Tindie]()	74.98
2 x Pico W	Digikey	12
Caller PCB1	JLC	22.05
Caller PCB2	JLC	15.44
Motion PCB1	JLC	6.5
Motion PCB2	JLC	6.76
6 Dof Sensors	Amazon	15.95
10x HC05	Amazon	37.98
3D Filament	Amazon	38.99
New Wheels	Adafruit	28.17
VL53l1x	Sparkfun	113.81
FPR PCBs	JLC	27.84
LiPo Batteries	Amazon	24.99
Wheels	Amazon	44.99
3D Filament	Amazon	24.99
4x Motors	Amazon	43.56
539		

E. Project Management

Our team consists of 4 peers (one electrical engineer and three computer engineers) and an advisor. Due to having an even number of members we know that avoiding information silos and communicating is extremely important. We are working on getting better at this as we tend to communicate in pairs, but need to spread the information across the entire group. We meet in person and on Discord frequently and spend casual and work hours in the senior design lab.

If there is to ever be a major disagreement, we have our advisors able to help push us in a direction to keep motion; however, this situation has yet to happen. Our advisor has been able to grant a significant amount of insight into our project. The team is working; however, we are encountering more hurdles than we anticipated.

Jina (EE): Well-rounded with expertise in circuits and electronics. Working towards PCB and 3D design.

Redwan (CompE): Good knowledge in embedded systems and good understanding of most ECE topics like programming and basic circuits.

Soham (CompE): Good knowledge in software and embedded programming. Front end development of team website, low level programming of sensors and high level programming of mapping and pathing algorithms.

Omar (CompE): Possessing a strong background in software, the primary focus will now be on enhancing hardware skills and expanding knowledge in that domain to achieve a well-rounded and versatile technical skill set.

We've been helping each other out as much as possible, especially with finding good and appropriate parts for our project. As we progress more with the project, we will be working together more often. We've had certain times where there was a lack of communication between team members, where some forgot to update others on their progress, which is very essential for making sure we are all on the same page. We are working to make sure everyone stays up to date and on the same page with all progress on the project.

F. Beyond the Classroom

Jina: Although I've had a little bit of experience with PCBs, this project really made me consider other aspects of PCB design that I have not focused on, such as decoupling elements and making sure everything is efficient and better routed. Learning KiCAD has been a great experience as it is a great tool for not only using in the industry but also for hobbies as well. I am also in charge of designing the chassis, which pushed me to get acquainted with Autodesk Fusion360 and how it can be used for 3D design. I think that learning how to use new software will be a useful skill regardless of the industry that we may go into in the future, which is why

this is a good experience for all of us.

Redwan: As my subsystem deals with communication with the sensor, the information gained in ECE 231 is proving to be quite helpful. Both parts of connecting the sensor as well as communicating with it were covered in that class which eases the learning curve for what I have to do a good bit. I still need to learn the specifics of my part, but I can focus on that rather than on how to do everything.

Soham: Revising C and Python for the embedded software and learning new libraries to provide functionality to the sensors. I have been reading up on mapping and pathing algorithms along with addressing edge cases that may come up in the obstacles of the bots environment. Along with software, I am helping design the chassis which requires me to learn Fusion360 and 3D designing skills.

Omar: Through this project, I've learned some PCB Design, which has proven to be quite fascinating. Furthermore, I've had to refresh my knowledge of physics, particularly in understanding the intriguing motion vectors of our wheels that enable omnidirectional movement. Throughout this project, I learned how to utilize the pico w along with a bluetooth module to perform wireless communication over many different devices.
