ECE 445

Fall 2023

Final Report

Team 1

Automatic Item Retrieval Bookshelf

By: Atharv Koshti (koshti2), Kashyap Ramachandrula (kramac3), Vraj Patel (vrajsp2)

TA: Nikhil Arora

Abstract

The document goes through the project that we built throughout the semester. The project we built is an automatic rotating bookshelf that is used to store electrical components. It is interfaced with an online web application that is used to insert and retrieve the items. The document goes through various aspects including our block diagram, design, costs, and results. Overall, our project ended up being a success as we were able to meet all our high level requirements.

1. Introduction

1.1 Problem

Oftentimes, when working on a project with a lot of small components, it can be difficult to keep track of everything leading to inefficiencies when working. Components such as resistors and capacitors that look similar are very difficult to keep track of, and can get lost and disorganized quickly. Something like a miniature bookshelf is what some use to organize items, but can often get very messy and tedious to find specific items. Through our project, we hope to find a more effective and efficient way to store small electric components while working on projects.

1.2 Solution

We developed a circular tabletop bookshelf with 4 shelves. Each shelf contains one section to place items, and the entire bookshelf rotates using a motor at the base of the bookshelf. Additionally, the bookshelf will be accompanied by a website to expand its functionality. This website will aid the user in finding their items easier by keeping track of where an item is inserted. When inserting an item, the user will enter the shelf ID into the app in order to mark that shelf with the current item. They will also enter a description of the item they are inserting. On retrieval, the user will be able to look up the item on the app, which will result in the bookshelf rotating to the correct side and lighting up the shelf and specific location where that item is located. The app will then mark that item as having been removed from the bookshelf.

The setup will be as follows. There will be shelf IDs placed on each side of the bookshelf labeled with a unique number. When the user looks to retrieve an item, the app will first figure out which section the item is located. Then a stepper motor will rotate the bookshelf so the correct side face faces the user. Next, the LED will turn on to illuminate the object being retrieved through a hole in the base of the shelf. Insertion is much easier as all the user has to do is type in the shelf ID of the section they put the item in and give a name to the item. This information will be stored by the website in order to ensure as seamless of an experience as possible for the user.

1.3 Visual Aid

A visual depiction of our rotating bookshelf can be found below. We used a stepper motor to control the rotation of the bookshelf and an ESP32 to control the data signals sent from the website application to the motor and LED.

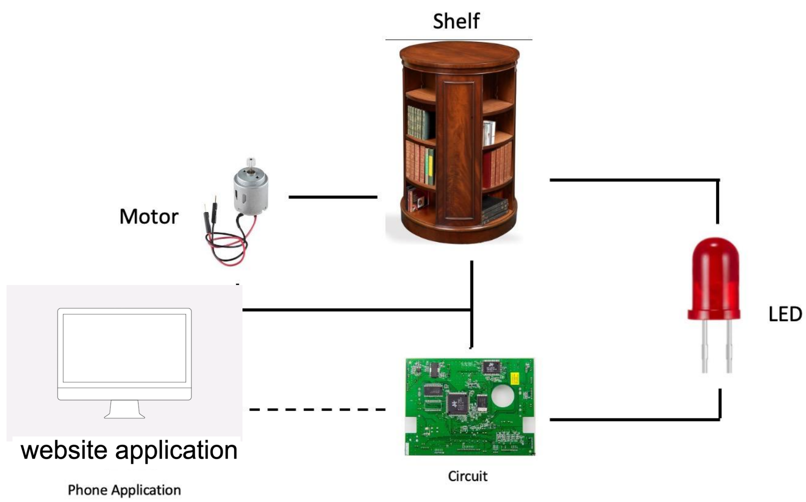

Figure 1: Visual Representation of Rotating Bookshelf

1.4 Physical Design

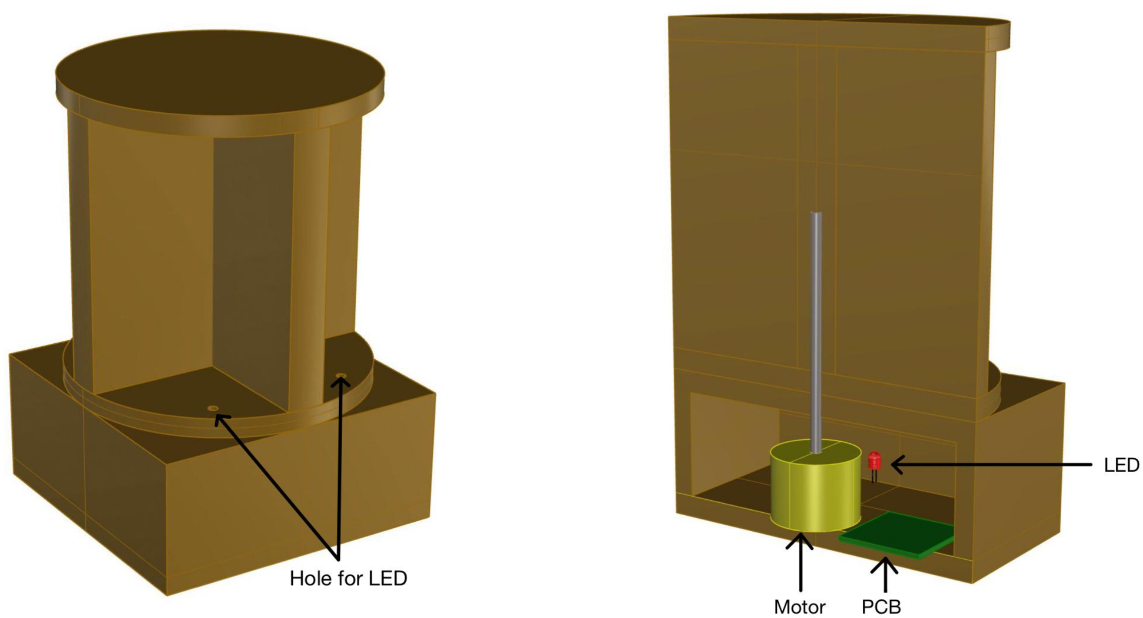

Figure 2: Isometric and Section View of Rotating Bookshelf

1.5 Block Diagram

Bookshelf System - Contains the Control Subsystem, Sensing Subsystem, and Power Subsystem that will be present on the bookshelf itself.

Remote System - Contains the User Interface Subsystem on the website application. The User Interface Subsystem is connected to the rest of the bookshelf system through wifi.

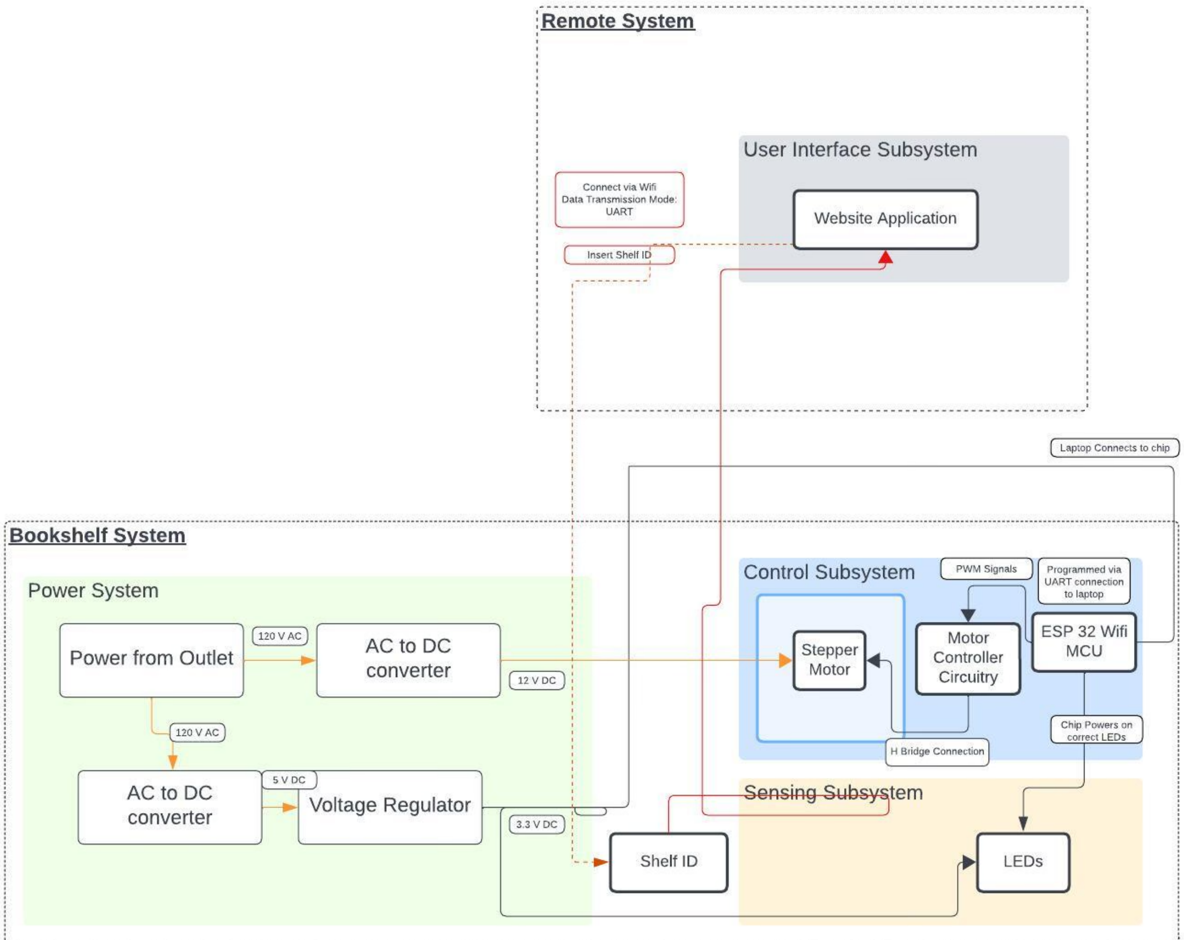

Figure 3: Block Diagram Representation of Rotating Bookshelf

1.6 High-Level Requirements

The following are the high-level requirements we hoped to accomplish with our bookshelf:

- Once prompted with the app, the bookshelf will turn to either angle 0, 90, 180, or 270 to face the user with the item that they requested to retrieve
- In the app, the user must be able to see 100% of the items they placed on the bookshelf (at least 8 items)
- Once the bookshelf has spun to the respective location based on what device the user is looking for, the LED will turn on within 2 seconds to illuminate that shelf

1.7 Subsystem Overview

1.7.1 User Interface Subsystem

Overview: This subsystem is what allows the user to interact with our device. On a website application, the user will be able to add items to the bookshelf storage system, remove items from the system, and see what items are currently stored in the bookshelf. Additionally, this subsystem will allow the website application to communicate with the PCB on the bookshelf system. It will use a wifi connection to send data between the User Interface System and the Bookshelf System.

1.7.2 Control Subsystem

Overview: This subsystem contains the microcontroller and motor. The microcontroller on the PCB will receive data from the remote system via a WiFi connection. Based on this, the microcontroller will output the appropriate signals to the motor to turn it to the correct orientation.

1.7.3 Sensing Subsystem

Overview: This subsystem contains the LED and shelf IDs, and in a way also acts as the main way the user interacts with the bookshelf system. The LED will alert the user to the exact shelf that their item is located, and the shelf ID allows them to input where they are placing the item to the system.

1.7.4 Power Subsystem

Overview: This subsystem will contain the circuitry to convert the power to appropriate voltage levels. It will distribute the power to the motor and all the components on the PCB such as the microcontroller.

1.8 Block-Level Changes Made

Throughout the development process, a number of changes were made to the design. These affected 2 main subsystems on the project. The power subsystem and the user interface subsystem. The modification made to the power subsystem involved how power was stepped down and distributed to the various components on the PCB. Originally, the design had a single 12V power supply that was stepped down to 3.3V for the microcontroller. The stepper motor ran directly off of the 12V supply. This design was originally chosen for simplicity, but for it to work, a buck converter was required as a linear regulator would not be able to handle the large voltage drop. However, the buck converter that we originally chose ended up being very small and we were not able to solder this component onto the PCB. To solve this problem, we decided to make a change on the final PCB order and use a linear regulator to step 5V down to 3.3V for the microcontroller. A 12V supply for the motor was still required, so this design requires 2 power supplies. However, we were familiar with linear regulators and thought that the drawback was worth it. Since we were working on tight timelines and did not have much time, being familiar with regulators and knowing that they would work was an important reason why it was chosen. Another change that was made was switching the user interface from being a mobile application to a website. This change was made to ease development, as interfacing with the microcontroller through WiFi was something that worked more smoothly than bluetooth. Using a website has the added benefit that the user does not need to directly connect to the microcontroller in any way. This is because all the communication between the user and microcontroller can occur between an intermediate server. These were the two largest changes made to the design. A number of smaller ones were made as we tested the project, but these were not significant enough to change the block diagram.

2. Design

2.1 Design Procedure

For each block of the project, we discussed what the goals of that block were. For example, for the remote system, the goals of the block were to keep track of all of the inventory in the bookshelf, and to be able to communicate with the microcontroller. Based on this, high level design decisions for that block were made. The main design decision within the remote system was whether to create a mobile application or a website. These decisions also affected the communication method we would use, as a mobile application lends itself to bluetooth, whereas with a website, WiFi is better. In the end, we chose to create a website as we were all more experienced in this area and we knew the microcontroller supported WiFi. The power system had a similar design process. We knew that the rest of the bookshelf needed 3.3V and 12V, so the power system needed to output these voltages. The original design called for a 12V input from a AC-DC converter which was stepped down to 3.3V with a buck converter. This was a simple design that accomplished the goals of this project. However, in practice, we were not able to solder the buck converter that we used. Additionally, we were nearing the end of the project timeline and so it was imperative for us to design something that we knew would work. So we created the power subsystem design we mentioned earlier that involved two inputs. A 5V input that would be stepped down to 3.3V, and a 12V input for the motor. We chose this as we had soldered the linear regulator we put in the design and we were confident it would work. Given more time, we would likely pick a more elegant solution. The final block in the project is the control subsystem on the bookshelf. This block had to receive communication from the web application, interpret that data, and rotate the bookshelf to the correct position. For this we chose an ESP-32 microcontroller as we knew it had the wireless communication capability we needed. We also chose a stepper motor as it would be able to precisely rotate to the angles that we require.

2.2 Design Details

For the power subsystem, care had to be taken when using the linear regulator to ensure that the linear regulator would not overheat. A summary of the current draw of the 3.3V components is shown in table 1, and the various parameters needed for this calculation are shown in table 2.

Part	Worst Case Current Draw at 3.3V (mA)
ESP32-S3	340
Red LED	15
Current Sensor (4)	9.6

Table 1: Current Draw of Various Components

Variable	Value
Max Temperature	150°C
Iout	364.6 mA
Vin	5V
Vout	3.3V
Θ_{ja}	100 C/W
T_a	40°C

Table 2: Parameters of Linear Regulator Calculations

$$T = i_{out} * (V_{in} - V_{out}) * (\Theta_{ja}) + T_a = 0.3646A * (5V - 3.3V) * 100\ C/W \tag{1}$$
$$T = 61.982°C + 40°C = 101.982°C$$

As this calculation shows, in the worst case the temperature of the regulator reaches 101.982°C, which is well below its maximum temperature. The final linear regulator circuit is shown in figure 4. A capacitor was added to smooth the output voltage, and a solder jumper was added so that the output voltage of the regulator could be measured before outputting it to the rest of the PCB. The test point was added so the voltage could be easily measured, however in a future revision this test point would be moved to the left of the solder jumper so that the regulator output could be easily measured.

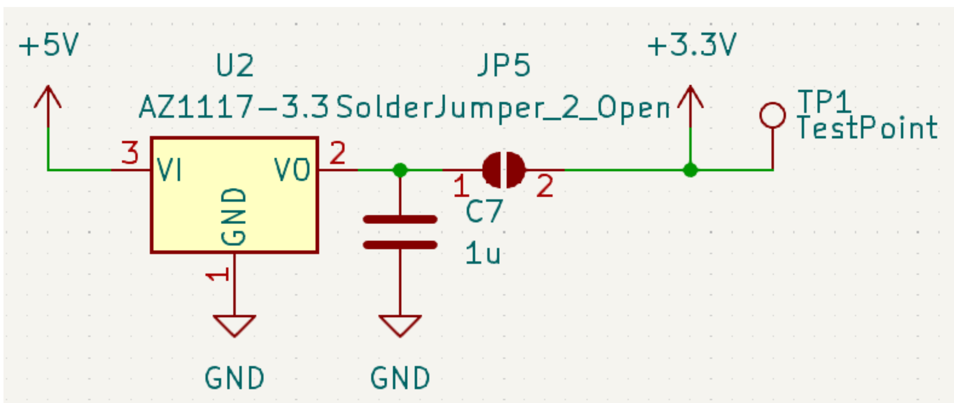

Figure 4: Power Regulation Circuit

When designing the control subsystem, it was important to make sure that all the components would be able to perform their necessary tasks. For example, the stepper motor was responsible for rotating the entire bookshelf assembly. Care had to be taken to ensure that the stepper motor would be able to generate the necessary torque to rotate the bookshelf. For this, we consulted the machine shop as they had the mechanical experience that we lacked. They were able to give us guidance on how to size the stepper motor and said that a 12V motor would suffice for our application. Additionally the L293DNE H-Bridge circuit had to be able to output the necessary current. For this, we compared the datasheets of the H-Bridge IC and the motor. The main thing we were concerned about here was how much current the motor would draw and whether the H-Bridge could handle it. The H-Bridge could handle 1A continuously and 2A at peak, while the motor would only draw 0.33A [8]. Therefore, we concluded that the L293DNE would be a suitable driver circuit for our application. The driver circuit is shown in figure 5, and consists of connections to the microcontroller as well as the motor.

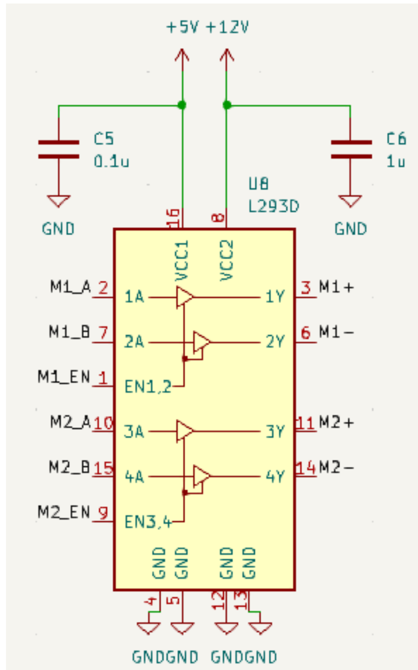

Figure 5: L293DNE H-Bridge Circuit

The final main piece of circuitry that we required was the current sense circuit. This was not implemented in time for the final demo, however given more time it would be something that we would finish. This circuit consists of a current sensor that outputs a signal to the microcontroller through a buffering op amp. This signal is proportional to the current that is being measured. The microcontroller would read the signal, and would trigger an overcurrent fault when the signal crosses a threshold. This threshold would be determined experimentally, but depends on the value of the shunt resistor and how precise the analog to digital converter of the microcontroller is. There are two of these circuits, as each phase of the stepper motor requires current sensing. A single current sense circuit is shown in figure 6.

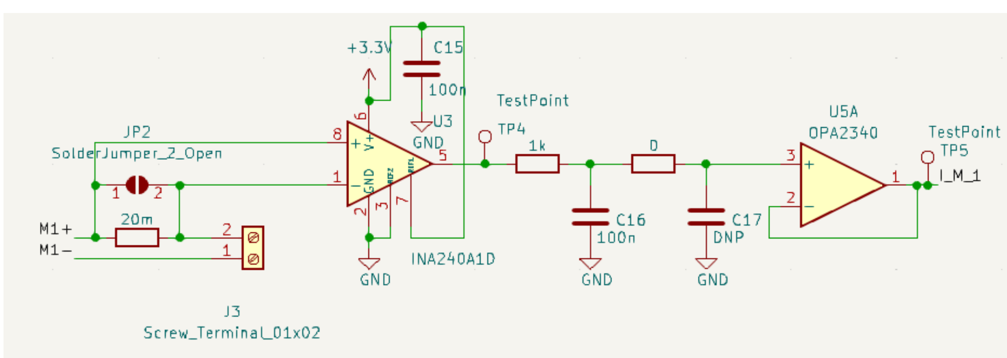

Figure 6: Current Sensing Circuit

2.3 Verification

2.3.1 Sensing Subsystem

The sensing subsystem had 3 major requirements that we needed to fulfill: the shelf ID should be assigned to a specific storage space in the application, the LED should only turn on when the hole of the correct shelf is directly above it, and the LED must be able to light up within 2 seconds.

In order to test the first requirement of there being a specific storage space for each shelf ID in the application, we had to check the web interface itself. We inserted 10 items on the entire bookshelf across various sections and inputted each item along with the corresponding shelf number in the website. Once we had done that, we checked the "Check Storage" tab on our website, and ensured that each item we had placed on the shelf was visible on the website. This process was repeated for all 4 shelves until we had determined that this requirement was being fulfilled.

To check the other two requirements of the sensing subsystem in regards to the LED, we had to use the website again. Based on the items that were currently placed on the bookshelf, we input an item in a shelf that was currently not facing us. Once we inputted the item, we waited for the bookshelf to move to the correct shelf, and using a stopwatch we checked how long after the bookshelf stopped at the desired location the LED turned on. Throughout this verification stage, we were able to identify that our LED was turning on after an average of 1 second after stopping at the right shelf location, allowing us to determine that both requirements of the LED turning on, and it turning on within 2 seconds were satisfied.

2.3.2 Power Subsystem

The power subsystem had one major requirement of being able to supply at least 350mA at 3.3V ± 0.1V. A change we made to our power subsystem was also adding a 12V power supply to drive the motor.

In order to test our power subsystem, we had to first solder all components including the barrel jack, voltage regulator, power rail, ground, and required resistors. Once that was completed, we connected the 5V power cable into the barrel jack to ensure that the circuit was getting power. Once plugged in, we used the multimeter's voltage reader to check if 5V ± 0.1V and 3.3V ± 0.1V was being delivered at the correct points on the PCB. The values the multimeter read once probed were 4.998V for the 5V locations, and 3.288V for the 3.3V location, meaning that our circuit was getting the correct power at the correct locations. The same steps were followed to check the power being delivered by the 12V power supply, and we determined that it was working as the multimeter read a value of 11.988V at the desired locations.

2.3.3 User Interface Subsystem

The user interface had 2 main requirements that we needed to fulfill including the user being able to accurately insert and retrieve items from the bookshelf on the app interface.

In order to test users being able to accurately insert items into our bookshelf on the application, we had to insert an item into the bookshelf. On the web application home page, we clicked the "Insert Item" tab, typed in the shelf ID of where we placed the item, and typed the name of the item that we just placed. We tested this functionality by going to the "Check Storage" tab on the Home Screen, and checking if the item we had just placed showed up on the screen. For all tests we ran, we were able to see 100% of the items we had placed on the bookshelf.

Testing the retrieve functionality was very similar. Based on the items we already had on the shelf, we clicked the "Retrieve Item" tab on the Home Screen, typed the name of the item we wanted to retrieve, and waited for the shelf to rotate to the shelf that the item was at. Once the item was retrieved, we went back to the "Check Storage" tab to see if the item was no longer on the shelf. For all tests we conducted, 100% of the items we retrieved from the shelf were getting removed on the application correctly.

2.3.4 Control Subsystem (Bookshelf)

The bookshelf control system had one main requirement of allowing users to choose what item they want to retrieve, and having the bookshelf turn towards the user with the requested item.

We ran multiple tests on the bookshelf as this was the final product we wanted to demonstrate. Similar to the user interface test, we essentially input an item on the bookshelf that we were looking to retrieve from the bookshelf on the homepage. Once inputted on the website, we waited for the bookshelf to turn from the shelf that it was currently on to the shelf that had the item we were looking for. We ran this test to check if all shelf-to-shelf combinations worked, including 1 to 2, 1 to 3, 1 to 4, 2 to 3, 2 to 4, 2 to 1, 3 to 4, 3 to 1, 3 to 2, 4 to 1, 4 to 2, and 4 to 3. For all combinations we soo 100% accuracy, which allowed us to determine that the bookshelf control subsystem was working.

2.3.4 Control Subsystem (Remote)

The remote control system had one main requirement of being able to accurately transmit data from the web application to the bookshelf.

The test for the remote subsystem was essentially combining the tests we ran for the other subsystems we had. Once the web application was completely implemented, we ran tests for retrieve and insert to see if the website was effectively communicating with the bookshelf circuit.

For retrieval, we typed an item that was already stored on the bookshelf to retrieve, and checked if the correct item on the correct shelf was being transmitted through the ESP32 to the bookshelf through the Arduino IDE [7]. Once that was confirmed, we checked the bookshelf to see if it was turning to the correct shelf with the item we were trying to locate, and turned on the LED. As mentioned previously, the retrieve function worked 100% correctly for all shelf-to-shelf combinations.

For insert, we followed a similar methodology in which we used the implemented insert tab on the website to pick a shelf ID to insert an item in, along with inputting the name of the item. We checked if the shelf ID and item name was being transmitted through the ESP32 to the bookshelf on the Arduino IDE, and confirmed if the item showed up on the check storage tab after inserting. As mentioned previously as well, the insert function worked 100% correctly for all shelf IDs.

2.4 Costs

2.4.1 Labor Costs

We are assuming that the average ECE graduate from UIUC makes $50 per hour. We worked on this project for approximately 10 hours per week for the last 11 weeks. Therefore, we plan to work for 330 hours with all three of us combined at a rate of $50 per hour. As a result, our total labor costs are $13,500 * 2.5 = $41250. We estimate that we require around 20 hours of labor from the machine shop over the course of the semester to finish our project.

2.4.2 Parts Costs

Part Type	Manufacturer	Model Number	Cost	Quantity
Microcontroller	Espressif Systems	ES32-S3-WROOM-1	$2.95	1
Stepper Motor	Mercury Motors	sm-42byg011-25	$15.00	1
Linear Regulator	Diodes Incorporated	AZ1117-3.3	$1.10	1
5-pin connector	Sullins	PPTC051LFBN-	$0.43	1

	Connector Solutions	RC		
H-Bridge	Texas Instruments	L293DNE	$4.08	1
Current Sensor	Texas Instruments	INA240	$3.23	2
Op Amp	Texas Instruments	OPA2340	$4.87	1
Barrel Jack Connector	CUI Devices	PJ-102Ah	$0.74	2
NPN Transistor	Fairchild Electronics	SS8050	$0.29	2
12V Power Supply	TriMag	L6R06H-120	$6.36	1
5V Power Supply	TriMag	L6R06H-050	$7.72	1

2.4.3 Total Costs

The total cost for the parts was $51.03. The total labor costs were approximately $41250. This results in a total project cost of $41,301.03

2.5 Schedule

Week	Task	Person in Charge
Week of 10/2	• Design Review with Instructor and TAs • Review order with TA and submit order request • Visit machine shop to submit final design for shelf • Begin designing PCB	All

Week of 10/9	• Determine what parts need to be ordered • Begin designing website application • Check-in with machine shop on status for shelf	Atharv (Ordering Parts) Kashyap (Website) Vraj (Machine Shop)
Week of 10/16	• Continue building mobile application • Parts should be delivered by EOW • Add motor to shelf and test if it is able to turn smoothly	Atharv (Parts) Kashyap (Website) Vraj (Motor)
Week of 10/23	• Complete PCB design • Connect mobile app to bookshelf via bluetooth • Start adding LED functionality to shelf	Atharv (LED) Kashyap (Website) Vraj (PCB)
Week of 10/30	• Complete LED functionality • Integrate PCB, motor, and LED together with shelf and app	Atharv (LED) Kashyap (Integration) Vraj (Integration)
Week of 11/6	• Run thorough tests to ensure motor, app, and shelf are working properly • Run through practice demos for next week	All
Week of 11/13	• Mock demo with TA	All
Week of 11/20	• Fall break	All
Week of 11/27	• Final demo with TA	All
Week of 12/4	• Final presentation	All

3 Conclusions

3.1 Accomplishments

We had countless accomplishments during the course of this project. To start off, the website application was fully functional. This website application had a fully developed frontend, backend, and was able to connect to a server successfully. In practice, the website application is able to send an HTTP POST request to the server containing the shelf ID, which is then picked up by the ESP32 through an HTTP GET request. Another success of this project was the motor control. Although the motor itself was quite old, we were able to gain control and use it to spin the shelf to its correct position. Likewise, the LED turned on right after the bookshelf stopped spinning as we defined under our high-level requirements.

3.2 Uncertainties

In our project, the biggest issue we dealt with was getting the PCB to work. The issues started with the ESP32 not programming after it was soldered onto the PCB. Despite the many times we resoldered the ESP onto the PCB, there was always some underlying issue with getting it to flash. In fact, the ESP broke on several occasions and resulted in massive setbacks throughout the semester. On the final day of work, the PCB was finally working, but the power regulator stopped working. We believe that this issue came from the equation we used to figure out the values of the power subsystem. The equation was as follows:

$$T = i_{out} * (V_{in} - V_{out}) * (\Theta_{ja}) + T_a = 0.3646A * (5\,V - 3.3V) * 100\,C/W$$

$$= 61.982℃ + 40℃ = 101.982℃$$

We believed that this temperature was significantly below the maximum operating range for the AZ1117-3.3 [5]. Based on this preliminary result, we assumed that the AZ1117-3.3 will be able to output the required power while not overheating. However, we were incorrect in this assumption, as when our PCB started to work, the power regulator overheated and broke.

3.3 Future Work

There are three main additions we would like to make to this project in the future. First, we want to make the LED face the user or increase its brightness to make it more visible when lit. Currently, the LED is not too visible due to two reasons. For one, the base of the bookshelf is extremely thick. As a result, when the LED lights up, there is a lot of surface area that blocks the light from getting through. Second, the LED itself is not powerful. In order to fix this issue, we would make the bookshelf slimmer while also finding an LED that is brighter. The second big change we would make is to add horizontal shelves to separate components within each of the 4 shelves. The main purpose behind this addition is to add more storage space to the bookshelf so that the user can store more items in their work environment. The last addition would be to create a tab on the website to see items in all shelves at once. At the moment, there is a section on the website to view the storage of each individual shelf. However, it would be extremely beneficial if the user could see the total storage of the bookshelf at once.

3.4 Ethical Considerations

There are some ethical and safety concerns that we considered when building our automatic item retrieving bookshelf. Additionally, as discussed in the IEEE code of ethics section III, we treated all people involved fairly and respectfully, regardless of race, gender, disability, age, origin, sexual orientation, gender identity, or gender expression [3]. We were open to any and all criticism, and made sure to fix any mistakes made along the way while prioritizing the safety, health, and welfare of everyone involved.

Privacy: Privacy is a very prevalent ethical concern that many devices face. It was imperative that we considered the privacy of bookshelf users to ensure that we weren't collecting and storing any personal information that could be used negatively. When designing the app, we adhered to the guidelines of the IEEE code of ethics section 1, and ensured that privacy was considered throughout the design process. Our team was transparent with the users involving what data is being collected and how it is being used.

Data Security: For data that we were collecting, it was important to protect the data effectively to ensure that no sensitive information is being stored in insecure locations. We also made sure to not collect any more data than is necessary for our design to work. We also adhered to IEEE Code of Ethics section 10, and made sure that the entire team upholds these ethics. We ensured that our fellow team members followed the rules, and upheld each other to the highest standards.

3.5 Safety Concerns

Physical Safety: When thinking about physical safety with the design of our bookshelf, it was important to consider that the mechanical and electrical components of the device are up to industry standards. With the mechanical components, we had to make sure that the motor was strategically placed in locations that wouldn't be able to physically harm users of the bookshelf. With electrical components, it was important to ensure all wires were properly grounded and weren't exposed. Since this design contained a 120V AC power, we made sure to design a safe power distribution system that followed industry standards. We also ensured that there are no pinch points in the mechanical design that could harm the user. We followed IEEE Code of Ethics section 1 and kept the safety of our users paramount throughout the design process. Our primary goal throughout the design process was to create a system that prioritizes safety.

Fire Safety: As we were using electrical components and various power sources, it was important to ensure that we didn't input too much power at a given time to reduce overheating and potential fire hazards. We also sized all conductors correctly to reduce the risk of overheating, and had to use appropriate overcurrent protection methods. This was an especially important point to consider because our design contained a motor which had a high inrush current. We made sure to keep these considerations in mind when we developed our system.

4. Appendix

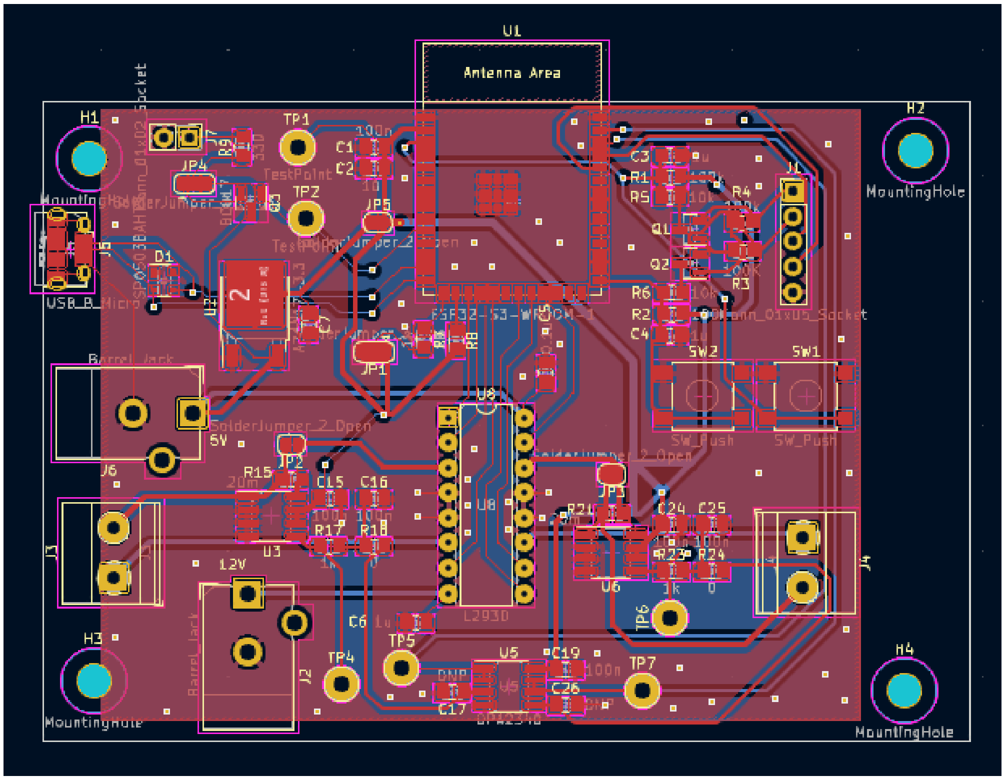

Figure 7: Final PCB Layout

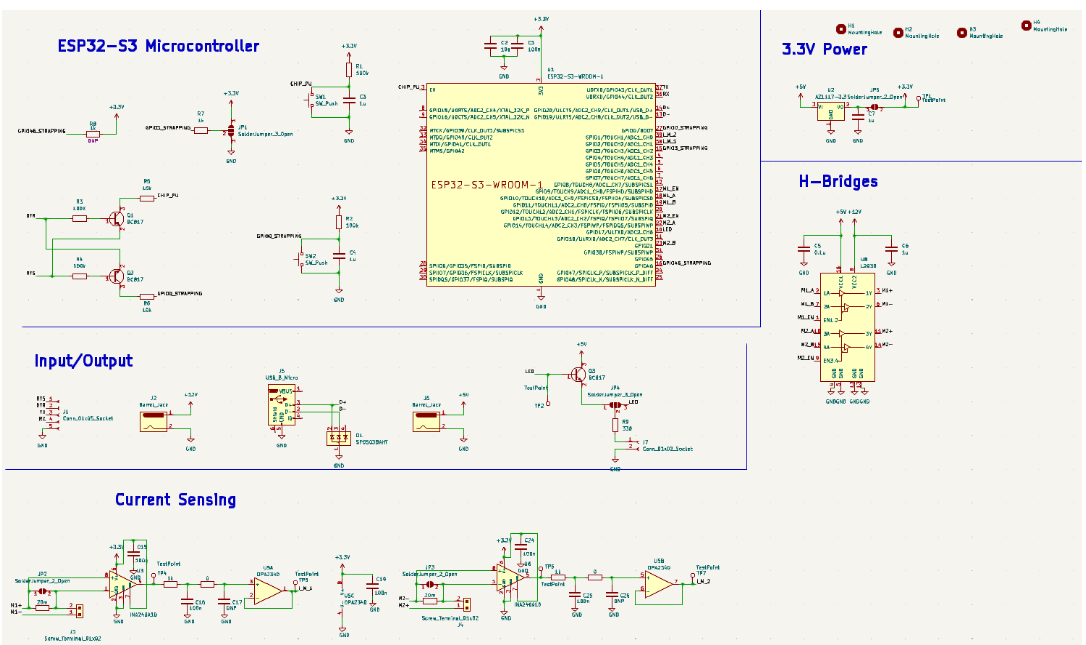

Figure 8: Final PCB Schematic

4.1 Requirements and Verification Tables

Requirements	Verification
• The shelf ID should be assigned to a specific storage space in the app	• Insert the shelf ID for 10 objects into the app • Using the app, check the contents specific shelf ID • Verify that only the items of that shelf are showing up under that shelf ID • Repeat steps 2 and 3 for the other three shelves until all 10 objects have been verified in the correct location
• The LED should turn on only when the hole of the correct shelf is directly above it	• Insert the shelf ID for 10 objects into the app • Using the app, retrieve one item in one of the shelves. • After the motor rotates to the correct location, check the LED displacement from the hole that is above it • If the LED is within +/- 1 cm when it turns on, the LED is in the proper location
• LED lights up within 2 seconds of retrieval	• Insert the shelf ID for 10 objects into the app • Using the app, retrieve an item of choice. • After the bookshelf rotates and comes to a stop, start a timer • Stop the timer when the LED turns on • Ensure that the timer reads less than 4 seconds • Repeat steps 2-4 until the LED has been turned on within 4 seconds for each of the four shelves

Figure 9: Sensing Subsystem RV Table

Requirements	Verification
• The Power Subsystem must be able to supply at least 400mA to the rest of the system continuously at 5V ±0.1V and at 350 mA 3.3V ±0.1V.	• Connect to power and ensure all systems are attached to the bookshelf. • Solder wires to the power rail and ground. • Connect the multimeter to the probes to measure the voltage of the device of a

	multimeter to ensure it is 5V ±0.1V and 3.3V ±0.1V after the step down. ● Measure the current by connecting a resistor to the power supply and placing the probes of the multimeter on either side of the resistor ● Check to make sure that the current is greater than or equal to 400mA/350mA

Figure 10: Power Subsystem RV Table

Requirements	Verification
● The system should allow users to choose an item to retrieve, and have the bookshelf turn towards the user with the requested item	● Using the app, choose an item from the bookshelf you want to retrieve ● Once chosen, the bookshelf should spin to "angle 0," which faces the user ● If the bookshelf stops within +/- 10 degrees of "angle 0," it can be assumed that it works properly

Figure 11: Control Subsystem (Bookshelf System)

Requirements	Verification
● The system should allow users to accurately input bookshelf items onto the app interface.	● Choose an item from the bookshelf to add to the app. ● Input a description of the item, and the shelf ID of where the item has been placed. ● Once added, verify that the item is shown on the app with the correct description and bookshelf location.

• The system should allow users to accurately choose an item from the bookshelf to retrieve.	• Choose an item from the app that you want to retrieve. • Once an item has been chosen, observe the bookshelf to see if it turns to the correct location and lights up within 15 seconds.

Figure 12: User Interface Subsystem

Requirements	Verification
• The system should be able to accurately transmit data from the mobile app to the bookshelf	• Once app has been completely implemented, verify if all items on bookshelf are shown on the app interface • Once all items on bookshelf are verified, choose one item that you would like to retrieve • Once item has been selected, observe bookshelf to see if it turns to the correct location with the item you are looking for, and if the LED turns on • Repeat for different items on the app to confirm if app and bookshelf has synced properly

Figure 13: Control Subsystem (Remote System)

5 References

[1] Express - Node.js web application framework, https://expressjs.com/. Accessed 19 November 2023.

[2] Render: Cloud Application Hosting for Developers, https://render.com/. Accessed 20 November 2023.

[3] IEEE, 16 June 2023, https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 6 September 2023.

[4] Texas Instruments, 16 June 2023, http://www.ti.com/lit/ds/symlink/ina240.pdf. Accessed 20 September 2023.

[5] Mouser, 16 June 2023, http://www.mouser.com/datasheet/2/115/AZ1117-1596017.pdf. Accessed 15 October 2023.

[6] “Connect to Wi-Fi networks with the ESP32.” uPesy, 20 January 2023, https://www.upesy.com/blogs/tutorials/how-to-connect-wifi-acces-point-with-esp32. Accessed 8 October 2023.

[7] “ESP32 Technical Reference Manual.” Espressif Systems, http://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. Accessed 20 September 2023.

[8] Hertz, Jake. “H-bridge DC Motor Control Using Complementary PWM, Shoot-through, and Dead-time - Technical Articles.” All About Circuits, 29 August 2022, http://www.allaboutcircuits.com/technical-articles/h-bridge-dc-motor-control-complementary-pulse-width-modulation-pwm-shoot-through-dead-time-pwm/. Accessed 20 September 2023.

[9]“Quick Start – React.” React, https://react.dev/learn. Accessed 6 November 2023.