Automated Closet

ECE 445 Final Report

Shania Arora, Nikhil Parmar, William Doherty
Team 13
TA: Enliang Li

December 11, 2019

Abstract

The purpose of this document is to describe our ECE 445 semester long project, the Automated Closet. The Automated Closet suggests an outfit for the user based on weather, color, and fit of clothes. The closet is a wooden structure which uses a bidirectional bike chain as well as motors, motor controllers, and encoders to control movement and positioning of the clothes. The user presses "choose outfit" for the closet to spin to the chosen articles of clothing and "reset" to put the clothes back into the closet after the clothes have been washed.

Table of Contents

1. Introduction 1
    1.1 Problem and Solution Overview 1
    1.2 High Level Requirements and Justification 1
    1.3 Block Diagram 2
    1.4 Block-Level Overview 2
2. Design 3
    2.1 Design Procedure 3
        2.1.1 Power Supply Module 3
        2.1.2 User Interface Module 3
        2.1.3 Control Unit 3
        2.1.4 Sensor Unit 3
        2.1.5 Actuation Unit 3
        2.1.6 External Wi-Fi Module 4
    2.2 Design Details 4
        2.2.1 Physical Design 4
        2.2.2 Motors/Gearbox 4
        2.2.3 PCB 5
        2.2.4 Clothing Decision and Outfit Recommendation 7
3. Verification 10
4. Cost and Schedule 11
    4.1 Parts 11
    4.2 Labor 11
    4.3 Result 12
    4.4 Schedule 12
5. Conclusion 14
    5.1 Summary 14
    5.2 Accomplishments 14
    5.3 Uncertainties 14
    5.4 Future Work 14
    5.5 Ethics and Safety 15
References 16
Appendix A 18
Appendix B 20

1. Introduction

1.1 Problem and Solution Overview

Choosing what to wear is an unavoidable part of everyone's routine, and as of today, there is no "automation" to this process. Moreover, many people spend a lot of time in the mornings deciding on what clothes to wear. In Illinois especially, the weather varies drastically every day, and many people consult weather apps to decide their outfit for the day. This is a hassle and takes away precious minutes of sleep in the morning. According to a study done in 2016, women spend as much as 17 minutes choosing their outfit every morning, while men can take up to 13 minutes [1].

Our solution is an automated closet system that can select an outfit for the user based on the weather and types of clothes. The user inserts data about the clothes the first time the closet system is set up. For every article of clothing, a database stores details about each article of clothing such as color category, fit, database index, and type of clothing.

Although motorized closet systems exist, they do not have the decision making capabilities needed to optimize their functionality. Most solutions are similar to the Closet Carousel, which simply has a foot pedal to control movement [2]. This solves the problem of increasing usable closet space, however still puts the outfit selection process on the user. Our solution provides this additional functionality in an easy-to-use package. Additionally, these solutions can require significant effort to install, reaching weights upwards of 135 lbs with an exorbitant price tag of $3000 or more to match [3]. The Automated Closet is significantly lighter and requires little to no installation.

1.2 High Level Requirements and Justification

- The closet connects to an online database and can pull information on current weather and about clothing in the closet
- The closet suggests an outfit to the user
- The closet spins with all the clothes in the system and is able to track the position of clothes that are input into the system by the user

In order to meet our project's goals, we settled on these three high level requirements. In order for our closet to make a recommendation, it needs to be aware of both the clothes it has and the current weather, so it can plan for temperature and precipitation. Therefore, our first requirement is for our closet to connect to a database and reference the clothes in the closet, as well as be able to pull information about the current weather. Given the available clothes and current weather information, our closet must then be able to choose a complete outfit for the user to wear. This provides the functionality required to set our project apart from commercial options that require a user to move the closet themselves to their selected outfit. Lastly, in order to present our selected outfit to the user, we need our closet to be able to move clothes around while fully loaded and keep track of the exact position of each article of clothing within the closet.

1.3 Block Diagram

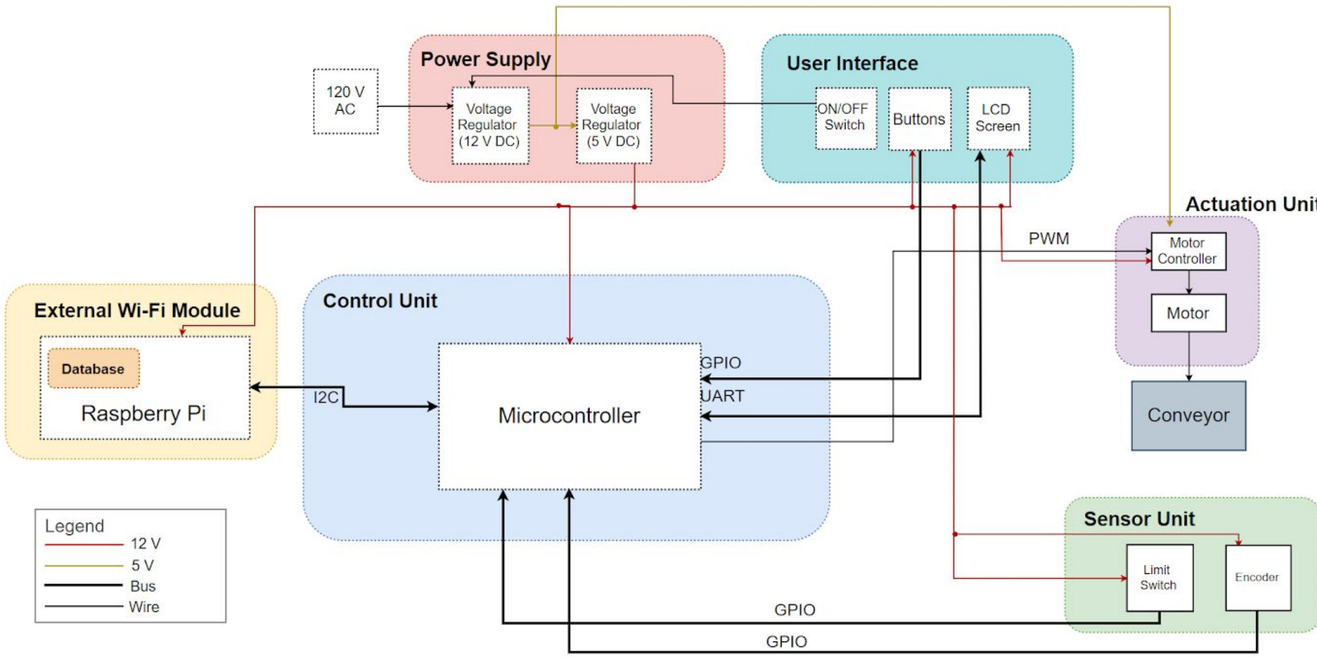

Figure 1: Block Diagram

1.4 Block-Level Overview

As Figure 1 shows, our project consists of 6 modules: power supply, user interface, actuation unit, sensor unit, control unit, and the external Wi-Fi module. The power supply module supplies 12V to the motor controller and 5V to the microcontroller. The user interface contains the "choose outfit" and "reset" buttons, so the user can interact with the closet. It also contains an LCD screen to display error messages, clothing indices, and other relevant information. The actuation unit uses PWM input from the microcontroller to control the bidirectional DC motor used on the conveyor system. The sensor unit produces signals for the microcontroller to indicate the rotational speed and position of the conveyor. The control unit processes user input from the UI module, tracks the position of the conveyor using sensor input, controls the motor speed and direction through PWM, and sends serial messages to our external Wi-Fi module so it can make API requests to query weather data. Lastly, the external Wi-Fi module connects the microcontroller to the internet so it can make API requests, which we use to retrieve information about the weather.

2. Design

2.1 Design Procedure

2.1.1 Power Supply Module

The design decisions for this module were straightforward. We needed two voltage regulators, one that supplies power to the motors and one that supplies power to the microcontroller. We had originally tried to use the provided power supply to step down from 120V to 12V. However, we noticed that the max amperage supplied was not enough for our motors. Because of this problem, we added another design consideration. Our 120V to 12V converter also had to be able to supply at least 7A of current for both motors to spin at full speed.

2.1.2 User Interface Module

Our original plan was to design an app for the UI component, but this added a layer of complexity to the project that would make it even harder for us to complete the project within the given time constraints. Thus, we opted for a simpler, two-button and LCD screen, UI instead which was easier to implement on our microcontroller.

We planned to put a “reject outfit” button to give the user more control over the outfit selection process, but we decided to take this feature out for the sake of simplicity. Now, the closet suggests an outfit to the user and if the user does not like it, the user may put the clothes back using the “reset” button functionality.

2.1.3 Control Unit

We decided to use an ATmega2560 chip, the same chip that is inside an Arduino Uno, for our microcontroller. Because we are all familiar with Arduinos from ECE 110, we chose this chip to serve as our microcontroller. This chip has the necessary number of GPIO and PWM pins to interface with the sensor and actuation units. IThe ATmega2560 t is also well-documented, so it was naturally the best option to use as our microcontroller.

2.1.4 Sensor Unit

We chose to use an encoder over a limit switch since encoders are mechanically locked and would rotate at the same rate as the chain in the closet. This would give us a higher degree of accuracy and precision. The limit switch would depend on being activated by the moving clothing hangers on the chain and the chances of a mechanical or index malfunction would be higher.

2.1.5 Actuation Unit

Originally, we had made the decision to use an H-bridge to drive our motors. H-bridges are simple, and we can buy them off-the-shelf. In fact, we planned to solder them directly onto the

PCB. However, after our PCB became unresponsive, we were unable to use these H-bridges. To make sure we could finish our project on time, we quickly decided to use prepackaged motor controllers. An added benefit of using these motor controllers is that they can handle higher current loads than H-bridges, which mitigates motor stall conditions.

2.1.6 External Wi-Fi Module

Our original plan was to use Google's Firebase database to hold information about each article of clothing. We decided to replace the database with a CSV file to make data access faster and our project simpler. In addition, the Raspberry Pi we were already using contained internet functionality.

2.2 Design Details

2.2.1 Physical Design

We designed the closet to fit an envelope of 4 feet long by 2 feet wide by 3 feet tall, which gives us a balance between the clothing capacity and the space used. In order to provide an easy system to both move and attach clothing to, we used a chain and sprocket system with specialized tabs that hangers can attach to. The closet has two BAG motors that are geared down 200:1, which can spin the chain bi-directionally. They are mounted in a block that slides to adjust tension to the closet. Figure 2 shows the conceptual design and the final product.

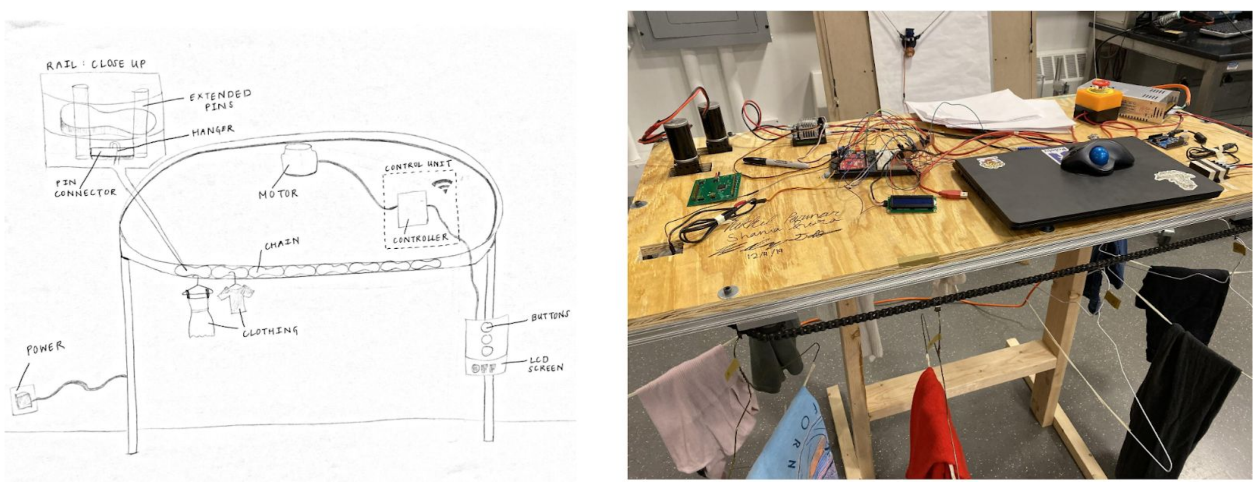

Figure 2: Visual Aid vs. Final Project

2.2.2 Motors/Gearbox

In order to determine an appropriate gearing for our motors, we began with a worst-case assumption that we would have all of the clothes weigh about the same as a fall jacket, which is roughly 6lbs. This assumption gives us an expected load of 66 lbs. We also wanted to keep the speed of the closet low for safety and usability, so we decided to plan for a speed of roughly 5 inches/second. After we knew the size of the sprocket that was selected by the machine shop, we were able to determine an appropriate gearing for our motors. We decided on a 200:1 ratio as seen in Table 1, which results in a current draw of 3.33A per motor. This current draw meets the manufacturer's provided specifications.

Linear Mechanism

	Free Speed (RPM)	Stall Torque (N*m)	Stall Current (Amp)	Free Current (Amp)
BAG Motor	13180	0.43	53	1.8
# Motors per Gearbox	Gearbox Efficiency	Travel Distance (in)	Applied Load (lbs)	Pulley Diameter (in)
2	70%	12	66	1.38

Driving Gear	Driven Gear		Elevator Linear Speed	Arm Time to move Travel Distance
1	4	No Load:	4.8 in/s	2.52 sec
1	5	Loaded:	4.6 in/s	2.63 sec
1	10			
1	1			
200.00 : 1	<-- Overall Ratio		Current Draw per Motor (loaded)	Stall Load
			3.33 amps	1544.33 lbs

Table 1: BAG Motor Specifications

2.2.3 PCB

The purpose of our PCB is to allow an ATMega2560 to receive input from a Raspberry Pi over a serial bus; receive GPIO input from buttons, switches, and encoders; output text to an LCD screen over I2C; and send PWM signals to motor controllers. In order to achieve this functionality, we added many subcomponents that are used to ensure smooth operation of an ATMega. Figure 3 shows the standard 16MHz oscillator circuit used to allow the device to operate at a standard clock speed, which is necessary for reliable PWM output. Figure 4 shows the hardware associated with the reset button, allowing the device to start fresh in the event of unforeseen circumstances. Figure 5 shows the pinout required for the ISP interface, which is critical as this is how we upload code onto the PCB. Figure 6 shows all connections to the external Raspberry Pi, which in this case only involve power delivery, the serial bus we communicate over, and connection over I2C as a backup in case we ran into issues with the serial bus. Lastly, Figure 7 shows the debounced sensors and user interface buttons that are connected to the ATMega over GPIO. The complete schematic is included in Appendix B.

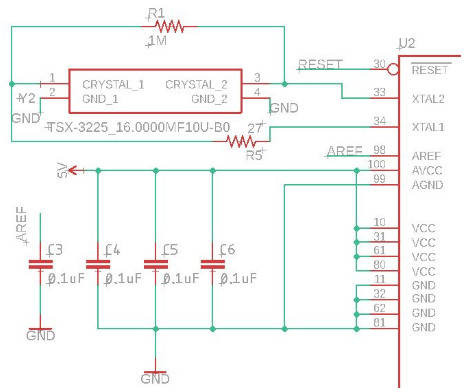

Figure 3: Oscillator Circuit

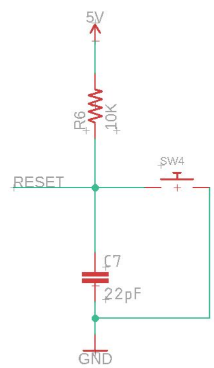

Figure 4: Reset Circuit

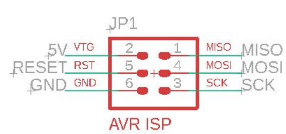

Figure 5: ISP Interface

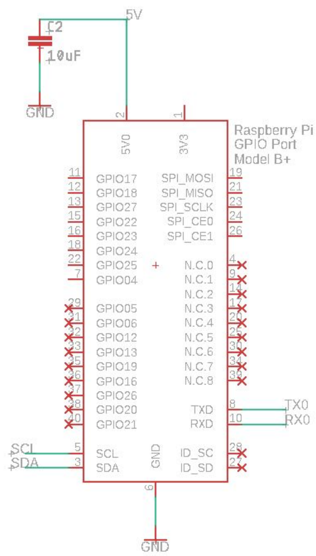

Figure 6: Raspberry Pi Connections

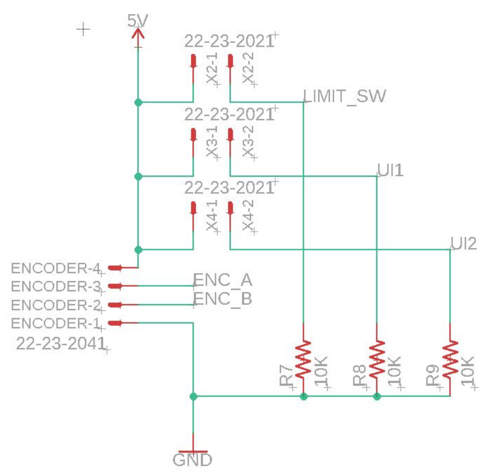

Figure 7: Sensors and Input

2.2.4 Clothing Decision and Outfit Recommendation

One of the high-level requirements for this project is that our closet can suggest an outfit for the user. The first step in this process is to determine what type of clothing is required by using our clothing-type recommendation algorithm. This algorithm uses weather data to determine what type of clothing should be recommended to the user. For example, if it is raining outside, the user needs a rain jacket. The clothing-type decision algorithm is detailed in Figure 8.

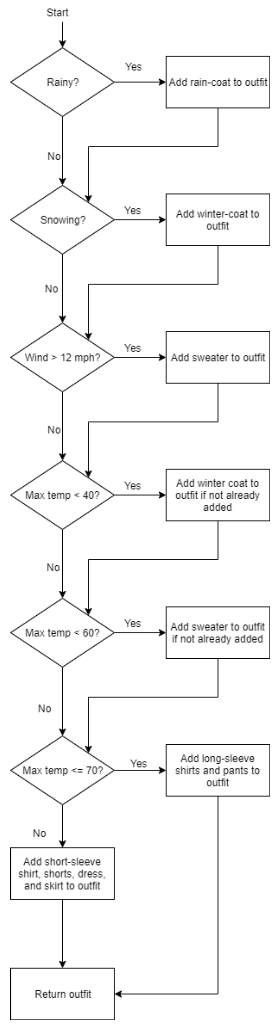

Figure 8: Clothing-type Decision Algorithm

After the clothes in the closet are filtered according to the weather, the clothes are filtered further based on the color and fit of the clothing. The algorithm starts by choosing a random pair of bottoms from the filtered list of clothes. It then attempts to match a top based on the color and fit of these bottoms. If there are no tops in the closet that match with the chosen pair of bottoms, the algorithm then randomly chooses another pair of bottoms and tries to match a top again. If the algorithm tries every possible pair of bottoms and no top is able to be matched with a pair of bottoms, the algorithm will display an error message which indicates that there are not enough clothes in the closet. If the algorithm is able to successfully match a top with a pair of bottoms, it then attempts to match a sweater and/or outerwear with the outfit if given weather conditions require those layers. The selected outfit is then presented to the user.

Instead of storing the color of each article of clothing in the database, we store the color category that the piece of clothing falls in. If an article of clothing is blue, we classify it into 1 of 3 color categories: light, medium, or dark. For any other color, we place the clothing into 1 of the following categories: hot (red, yellow, orange, pink, etc.), cold (blue, green, purple, violet, etc.), pastel, pattern, neutral (black, white, gray, beige, cream, etc.), hot dark (maroon, magenta, burgundy, etc.), hot cold (navy, dark green, dark purple, etc.). Clothes are classified into 1of 3 fit categories: relaxed (loose fitting), normal (neither loose or tight fit), or tight.

Within the greater design flow, this clothing-type decision and outfit recommendation algorithms are triggered from the controller module. Figure 9 displays how the above algorithms fit into the larger project.

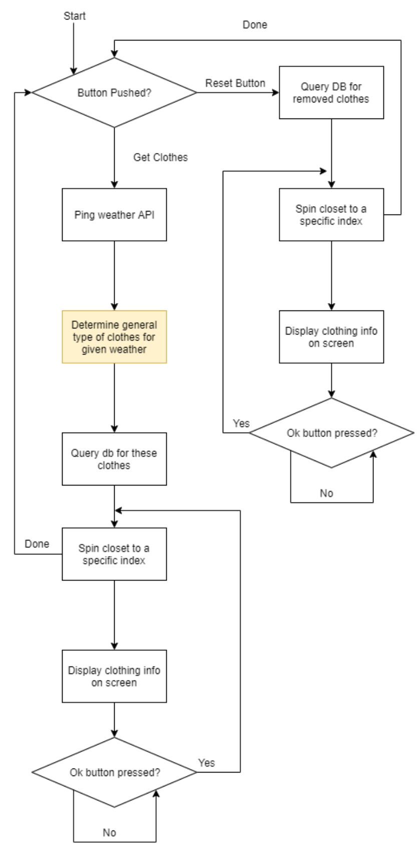

Figure 9: Control Module Flowchart

3. Verification

As seen in the tables in Appendix A, all our requirements have been verified.

To satisfy our power supply requirement of supplying 10 A of continuous current, we replaced our original 120V to 12V converter. Upon testing, it was only able to supply 0.7A of continuous current, which does not meet our stated requirement. This led to the direct failure of the motors even when they were attempting to spin without load. When replaced with a converter that was able to provide up to 30A, these issues were resolved and the closet was able to rotate as designed.

4. Cost and Schedule

4.1 Parts

Subsystem	Product Name	Quantity	Price ($)	Total ($)
Actuation	Talon Motor Controller	2	$59.00	$118.00
Actuation	Bag Motor	2	$29.99	$59.98
Actuation	Gearbox	2	$62.96	$125.92
Controller	Arduino Uno	1	$22.00	$22.00
External Wi-Fi Module	Raspberry Pi 3 B+	1	$35.00	$35.00
Power Supply	12V-5V Regulator	1	$9.59	$9.59
Power Supply	120V AC- 5V DC Regulator	1	$39.99	$39.99
Sensor	Encoder	1	$25.00	$25.00
Sensor	Limit switch	1	$1.50	$1.50
UI	LCD Screen	1	$10.95	$10.95
UI	Emergency Stop	1	$102.24	$102.24
UI	Pushbutton Switches	1	$5.85	$5.85

Table 2: Demoed Parts Table Cost

The sum total of the electronic components that we used on our demoed project is $556.02. However, since we already owned many of these components, the cost of the electrical components of our project was closer to $66 for our team.

Our project also has a significant mechanical portion. The machine shop handled the purchasing of all of the materials to build the closet, which included the wooden frame, the metal mounting brackets, and the bike chain. We were given an estimate for all these components of approximately $200.

In total, the cost of our demoed project was $756.02.

4.2 Labor

To calculate the labor costs, we used the formula:

ideal salary (hourly rate) * actual hours spent * 2.5 * number of people.

We spent approximately 14 weeks working on this project. Each week, each partner worked for around 15 hours, which gives us 210 hours spent per person. The number of hours worked is 50% larger than our original estimate. Using the $40/hour ideal salary from our design document, we get $40 * 210 hours * 2.5 * 3 for a total labor cost of $63,000.

Assuming that the machine shop spent a week on our project and that their salary is the same as our ideal salary, we add $40 * 80* 2.5 = $8000 to our labor costs, which brings the new labor total to $71000.

4.3 Result

In total, the cost of our demoed project was $71000 + $756.02 = $71756.02.

4.4 Schedule

Week	Billy	Shania	Nikhil
9/29 - 10/5	Design review; Talk to machine shop about physical design	Design review; Talk to machine shop about physical design	Design review; Talk to machine shop about physical design
10/6 - 10/12	Order parts; Design PCB for early order	Come up with code framework; Start writing Python code for outfit recommendation algorithm	Come up with code framework; Set-up database; Determine necessary parameter fields for clothing recommendation based on weather
10/13 - 10/19	Continue PCB design; Figure out interface between microcontroller and external Wi-Fi module	Continue coding; Python script to query database	Continue coding; Figure out interface between microcontroller and external Wi-Fi module
10/20 - 10/26	Work on individual reports	Work on individual reports	Work on individual reports
10/27 - 11/2	Wait on components, PCB, and machine shop	Wait on components, PCB, and machine shop	Wait on components, PCB, and machine shop
11/3 - 11/9	Machine shop finishes physical design; Fix PCB design	Machine shop finishes physical design; Work on controller module	Machine shop finishes physical design; Merge outfit recommendation and clothing recommendation

			algorithms
11/10 - 11/16	Work on/test power supply module. Order new power supply	Work on whatever modules that need to be finished	Work on interfacing between controller and external Wi-Fi modules
11/17 - 11/23	Work on actuation module	Start working on final presentation	Start working on final presentation
11/24 - 11/30	Fix power supply issue; Issues with AT-Mega chip; Switch to using Arduino	Prepare for mock/final demo and organize final presentation; debug encoder	Prepare for mock/final demo; help debug encoder issues
12/1 - 12/7	Closet working; Prepare for mock/final demo and organize final presentation	Closet working; Prepare for mock/final demo and organize final presentation	Closet working; Prepare for mock/final demo and organize final presentation

5. Conclusion

5.1 Summary

Overall, our project functions as expected and all the high-level requirements have been met. We are able to pull weather information and connect to a database. The outfit recommendation algorithm is able to suggest an outfit to the user. Lastly, the closet spins with all the clothes, and we are able to track the distance the closet has spun.

5.2 Accomplishments

We designed the automated closet according to our original plan and all features are functional. When the user presses the "choose outfit" button, the closet queries OpenWeatherMap API to check the weather in the user's location and filters the clothes in the closet according to weather conditions. The clothes are then filtered by color and fit. If necessary, a matching sweater or jacket is then added to complete the outfit selection process. The closet then spins to the chosen articles of clothing. Pressing "reset" cycles to the indices of the articles of clothing that were just chosen so the user can put back the clothes after they were washed.

5.3 Uncertainties

One of the major stumbling blocks our group hit during the development of our project was the sudden unresponsiveness of our PCB microcontroller. Having been able to upload code that could interact with devices over both a serial bus and I2C previously, the loss of this functionality late in the design cycle was sudden and unexpected. After investigating the error message that told us our device signature was invalid, we determined that due to an error during code uploads, the fuses inside the microcontroller were set to a state where a unique clock frequency needed to be supplied in order to allow the controller to function under normal circumstances [4], and we did not have the ability to supply this in a timely manner. This forced us to substitute in additional hardware to meet our functionality goals.

5.4 Future Work

We enjoyed working on this project and would like to see it develop into a full product. However, there are multiple changes that would be required before we can put this on the market. First, we want to move our clothing database and make it external. Currently, the database sits as a CSV file on our Raspberry Pi. For demonstration purposes, this CSV file was sufficient. However for an actual product, users would need a much more streamlined way to modify their closets when they buy new clothes or get rid of old clothes. Moving this database externally would mean that a user no longer would have to remove the memory from the Raspberry Pi, modify the CSV file, then move it back in. The user could use a web-application to modify their closets via the internet.

Our second modification lies within the fashion algorithm. We wish to add functionality that allows a user to program their style preferences into the system. For our demonstration, we used a set of rules that was designed by Shania. We made the decision that Shania's rules would be good enough to determine a good-looking outfit for the user. However, as a finalized product, we wish to know how other people determine what type of clothes to wear. With this information, we can make improvements to our outfit recommendation algorithm.

5.5 Ethics and Safety

There are a few safety concerns with our project. Since the automated closet has moving parts that are driven by a motor and the user interacts directly with them, the moving parts pose a safety hazard. Every moving part of the system was individually tested to make sure its behavior was safe and predictable. For instance, we chose that the motor will move the rail at ~5 inches per second since that speed is still fast enough for the user to look at the clothes, yet slow enough that the user is able to react if an unexpected event occurs. The average human reaction time is between 150 ms - 200 ms [5] which means our system will give the user a lot of time to act in case there is an issue (i.e. something gets caught, clothing item slips off hanger, etc.)

Moreover, we have an emergency stop to help the user if an unexpected event occurs. With any moving system, there are a multitude of ways a user can get injured. Although unlikely, an example of a way that someone can get injured is that their hair gets caught in the moving chain. To alleviate any problem with these moving parts, we connected an emergency stop in series with the motor controller power. That way, when the emergency stop is hit, all power to the motors is cut instantly. This switch is also located in an easily accessible place so if the system needs to be powered off, the user can do so quickly.

In addition to the moving parts, another potential hazard is exposed mechanical parts. We alleviated this safety hazard by ensuring that all edges were smooth and polished to mitigate the risk of injury.

We designed the automated closet because we wish to help people's lives by automating the task of choosing clothes. According to number 5 on IEEE's code of ethics, it is our responsibility to brief others on the capabilities of new technology [6]. We hope our product will be helpful for people and we plan on educating others about our product. Number 1 on IEEE's code of ethics states that the safety of the public is most important and that any potential concerns are addressed and appropriately handled [6]. We acknowledge all the possible safety issues and will make sure to take preventative measures to ensure our product is safe for the public.

References

[1] The Daily Mail Online. (2016). *Women spend SIX MONTHS of their working lives deciding what to wear*. [online] Available at: https://www.dailymail.co.uk/femail/article-3626069/Women-spend-SIX-MONTHS-working-lives-deciding-wear.html [Accessed 19 Sep. 2019].

[2] The Closet Carousel. (2019). [video] Available at: http://vimeo.com/162454530 [Accessed 19 Sep. 2019].

[3] Rotating Closets. (2019). *All Products | The Closet Carousel*. [online] Available at: https://rotatingclosets.com/pages/all-products [Accessed 19 Sep. 2019].

[4]"Device Signature 0x000000 Bootloader Atmega 1284P", *AVR Freaks*, 2019. [Online]. Available: https://www.avrfreaks.net/forum/device-signature-0x000000-bootloader-atmega-1284p. [Accessed: 11- Dec- 2019].

[5] K. Amano, N. Goda, S. Nishida, Y. Ejima, T. Takeda, and Y. Ohtani, “Estimation of the Timing of Human Visual Perception from Magnetoencephalography,” The Journal of Neuroscience, Apr. 2006. [Accessed 1 Oct. 2019].

[6] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available at: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 19 Sep. 2019].

Appendix A

**Any changes made to the R&V tables are indicated in the Results column*

Power Supply Requirements	Power Supply Verification	Results
The first regulator must be able to provide up to 10A of continuous current	Attach a resistor at the output of each regulator and use a multimeter to measure current through the circuit when skipping over downstream components. The first regulator should be able to handle a current of 10A. The second should be able to handle a current of 4A.	Verified successfully.
The second regulator must be able to handle current draws of 4A.		

Control Unit Requirements	Control Unit Verification	Results
Must be able to read input to and send output from a Raspberry Pi	Connect Raspberry Pi to Control Unit and send a serial message from the Raspberry Pi and see if we can read the serial message on the control unit	Verified successfully.
Must be able to send a PWM signal to move closet at an appropriate speed	Command a known PWM signal frequency and duty cycle, and measure the output with an oscilloscope to ensure the output meets expectations	
Can make closet stop at a specific index	Spin encoder a measured distance, use response from callback functions to ensure no counts were missed.	

Actuation Unit Requirement	Actuation Unit Verification	Results

Produce a voltage anywhere from -12V - 12V	Attach a multimeter to the output of the motor controller and gradually change the commanded output from -12V to 12V. Ensure that output meets these levels and that behavior is linear.	Verified successfully. Replaced H-bridge with the motor controller in verification.
Supply at least 6A	Attach a resistor at the output of the motor controller and use a multimeter to measure current through the circuit without the motor attached. The motor controller should be able to handle a current of 6A.	Verified successfully. Replaced H-bridge with the motor controller in verification.
Move a load of 20 lbs at at least 4 in/sec	Hang clothes at every index in the closet and verify the closet is still able to move at 4 in/sec.	Verified successfully.

Sensor Unit Requirement	Sensor Unit Verification	Results
Hanger index must be able to be calculated using the encoder position	Spin the closet system one hanger index at a time.	Verified successfully. Changed requirement to include encoder position instead of limit switch.
Control unit must be able to determine the magnitude and direction of the speed of the closet system.	Attach encoder output pins to oscilloscope and turn in both directions. Ensure that both pins produce a visible output and reverse their phase when the encoder is turned the opposite direction.	Verified successfully.

UI Module Requirement	UI Module Verification	Results
Display must show message for type of clothing when closet stops at a piece of clothing	At any index, print the index number on the screen	Verified successfully.
Emergency stop stops the closet immediately.	Make the closet spin by powering the motor controller, the emergency stop button should cut power at the motor controller	Verified successfully. Replace H-bridge with motor controller in the verification
Closet must be able to spin to all positions so the user can put clothes back one at a time when hitting the RESET button	Hit the RESET button and verify that the closet system correctly updates the database	Verified successfully.

External Wi-Fi Requirements	External Wi-Fi Verification	Results
Must be able to talk to control unit.	See Verification A for Control unit	Verified successfully.
Must be able to fetch information from weather API	Ping openweathermap.org and check for a valid response	
Must be able to connect to a system that stores information about the clothing	See “Database” for verification process	

Appendix B