ECE 445 Spring 2025

Senior Design Laboratory

Design Document

Automatic Card Deck Sorter

Team 37

Kyle Mahler - kmahler3

Rocky Daehler - walterd2

Alfred Hofmann - alfredh2

March 6, 2025

Table of Contents

1 Introduction

1.1 Problem

For centuries, card games have been a staple of entertainment. With just the same standard 52 card deck, hundreds of different card games have been produced over this time. However, in some of these games, there may be distinct and precise rules about setting up and managing the deck. For example, Euchre only uses the 9s, 10s, Jacks, Queens, Kings, and Aces, meaning players must manually sift through the deck before playing. Organizing the deck before playing games of this nature can be extremely tedious and time consuming. Players want to spend their time playing the game, not on the preparation of the game.

Beyond game-specific needs, many households, casinos, and clubs face the issue of reorganizing mixed or shuffled decks. Again, rearranging the cards back into a sorted order can take a long time and is by no means an exciting task. In a competitive setting, it may be essential to maintain a sorted deck before play to ensure fairness. At places with the need for a large number of decks to be sorted, the need for this process to be automated scales up drastically.

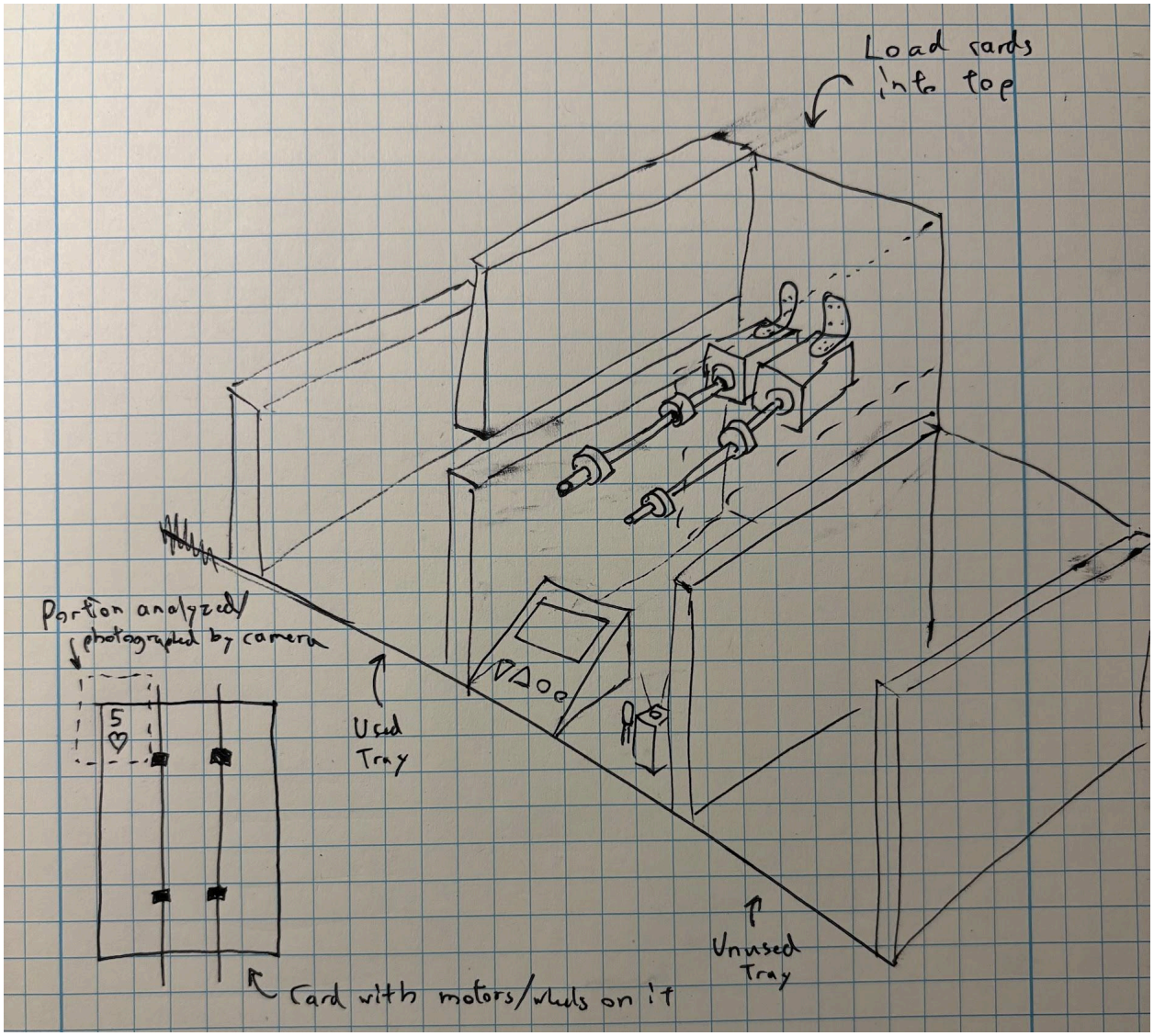
1.2 Solution

To address the inefficiencies of manual card sorting, we propose an Automatic 52-Card Deck Sorter. This device will quickly and accurately organize a mixed deck into an order specified by the user. This solution eliminates the need for players to manually separate cards for games like

Euchre, where only a subset of the deck is used. The sorter will incorporate a card recognition system to identify each card and a mechanical sorting mechanism to place them in the correct order efficiently. Additionally, a PCB based control system will manage the identification and sorting process, which will ensure accuracy and reliability. By automating this task, the device saves time, reduces human error, and enhances convenience for casual players and competitive tournament organizers alike. Whether preparing for a game, ensuring a properly ordered deck, or simply avoiding the hassle of manual sorting, this system provides a reliable and efficient way to manage playing cards, making it a valuable tool for both home and competitive settings.

1.3 Visual Aid

This is a rough sketch of what we think our final design will look like. A deck of cards is loaded into the top, as is labeled in the image, and there are walls on either side of this container to allow only one card to be moved at a time (note that the wall closest to the viewer is only partially drawn with dotted lines in this sketch).



1.4 High Level Requirements

The high-level requirements of our project are as follows:

- The camera can recognize the cards by suit and rank. It can recognize and sort 1 card in **4 seconds**, which translates to **3 minutes** for a 52 card deck
- Cards are successfully sorted into two piles, a ‘Used’ pile and an ‘Unused’ pile. The system goes until all cards are sorted into one of those two piles, and will automatically

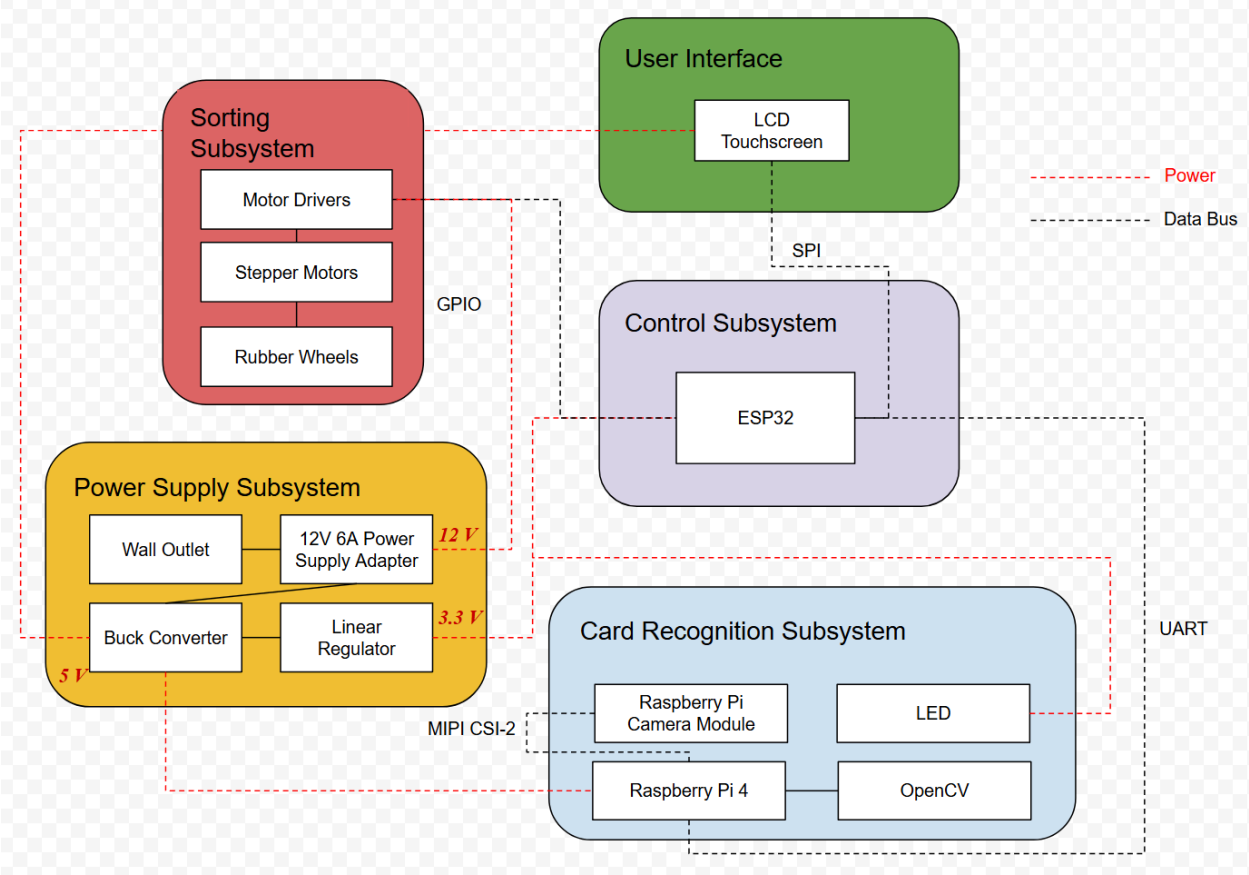
stop once there are no more cards to be sorted.

- The system can detect and display a warning sign or an error message when something goes wrong. It will do this when the same card is detected twice in a row (indicating a jam), a card is unrecognized by the camera, or the deck is incomplete.

2 Design

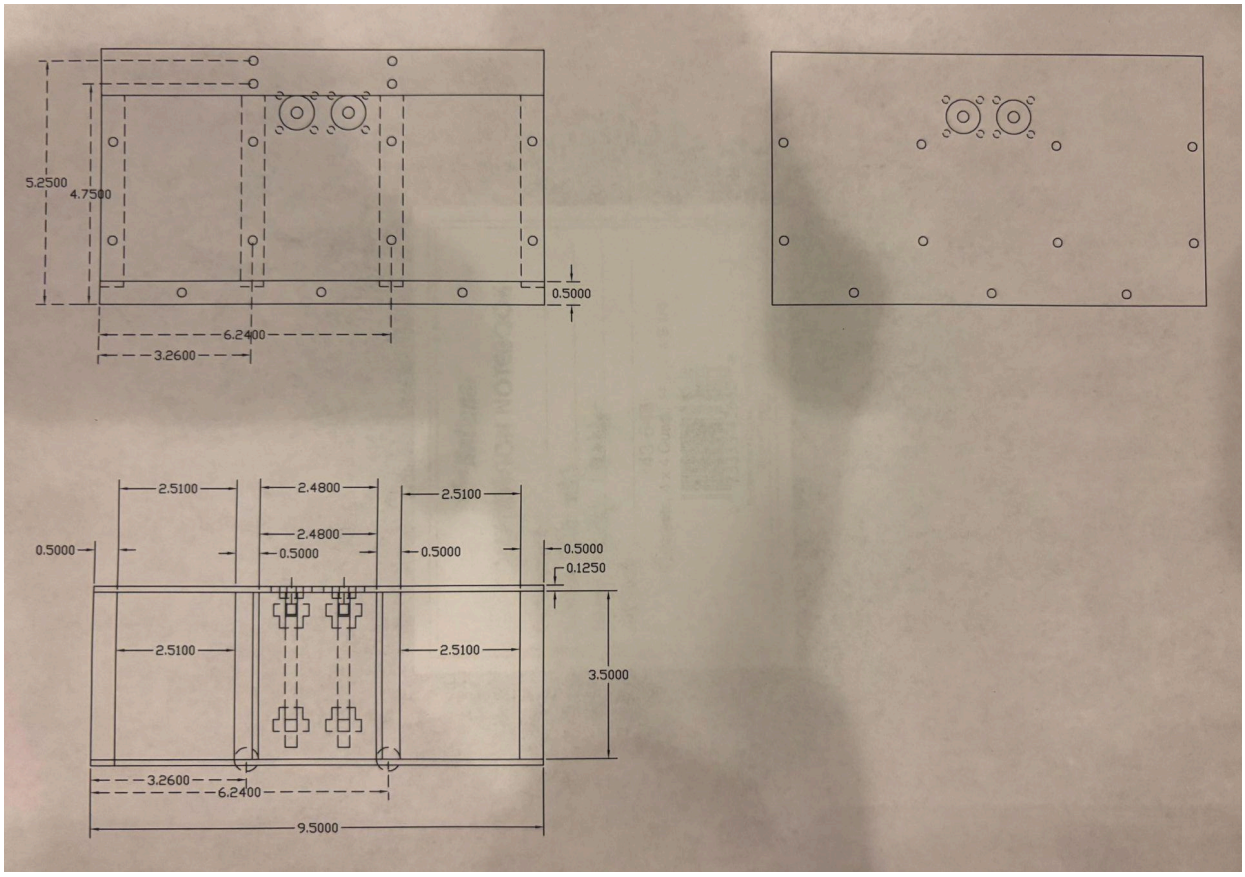
2.1 Block Diagram

This is the high level block diagram for our design.



2.2 Physical Design

For the physical design of the device, we're working with the ECE design shop. Skee Aldrich has helped us to design the physical body of our project, and it's physical dimensions are shown in the multiview projection below:



The body will be primarily made out of Medium Density Fibreboard (MDF), with the front and back face being made out of metal so as to be more easily drilled through.

2.3 Subsystem Overview

Our design is divided into 5 subsystems, which are implemented in both hardware and software.

2.3.1 Camera

The Camera subsystem, upon receiving a signal from the control system that a new card has appeared, will take a picture of the upper right corner of the card. It will then send that image to the Raspberry Pi to be analyzed. This process gets repeated until the camera sends an image of there being no more cards.

Requirement	Verification
The Camera can take a clear picture of the card corner and send that card to the Raspberry Pi. The Raspberry Pi will identify the rank and suit of the current card, including labeling some cards as unrecognized, with 95% accuracy or higher.	1. Setup the camera pointing at a set of cards with the same spacing/dimensions as in the final design. 2. Have the camera identify all of the cards in the deck, including jokers. 3. Calculate what percent of the cards were identified correctly (jokers should be identified as unrecognized) and make sure that 95% or more were calculated correctly.
The Camera subsystem can turn an LED on within 1 second of sorting starting and turn the LED off within 1 second of the sorting ending.	1. Connect the camera subsystem up to the Control System. 2. Send a ‘sorting starting’ signal from the Control System and measure how long it takes the LED to turn on. Verify that it takes less than a second. 3. Send a ‘sorting ending’ signal from the Control System and measure how long it takes the LED to turn off.

	Verify that it takes less than a second.

2.2.2 User Interface

The User interface will be a small screen with an assortment of buttons; Up, Down, Select, and Start. When the system is powered on, the screen will show a menu with a list of different games which have been preloaded for the system to sort into. Using the buttons, the user can navigate to their game of choice and hit the select button. Some examples of these games are Euchre, Short Deck Poker, Crazy Eights, or a custom range inputted by the user. Then, when the Start button is pressed, the system will begin sorting the cards. While sorting, the screen will display any pertinent information or errors - such as if a card jam is detected, if any cards are unrecognizable, etc.

Requirement	Verification
The user can use the User Interface to select a game to sort the cards for. The User Interface will send a signal indicating which game was selected within 1 second of the button being pressed.	1. Connect the User Interface to the Control System. 2. Select a game on the User Interface. 3. Measure the time between when the game is selected on the User Interface and when the signal is received by the Control System. Verify that this takes less than 1 second.

	4. Repeat this for all preloaded games.
When the user presses Start on the User Interface, a signal communicating what game/sorting has been selected will be sent to the Control System. This signal will be sent within 1 second of the button being pressed.	1. Connect the User Interface to the Control System. 2. Press 'Start' on the User Interface. 3. Measure the time between when the button is pressed and the signal is received by the Control System. Verify that this takes less than 1 second.
The User Interface can display an error message when given a signal from the Control System. Specifically, the User Interface can display A Jam error message, An Unrecognized Card error message, or A Missing Card error message within 1 second of receiving a signal from the control system.	1. Connect the User Interface to the Control System. 2. Send an error signal to the User Interface from the Control System. 3. Measure the time between when the signal is sent and the error message appears on the User Interface. Verify that this takes less than 1 second. 4. Repeat this for all three error types.

2.2.3 Sorting System

The sorting system is responsible for moving the cards in a way to ensure they are appropriately sorted. This will be done by placing the deck of cards on the top of two rubber tires, powered by

two motors that will slide each card to one pile or another through a slot the size of one card. Some sort of “metal card” will also be placed on top of the deck to add weight to the deck. This will ensure that as the deck thins out weight and friction with the rubber tires is not an issue.

Requirement	Verification
The Sorting System can move a card to either side based on an electronic input. It will move cards all the way into the selected tray, only resulting in a jam 15% of the time or less.	1. Connect the Sorting System to the Control System. 2. Send randomized sorting signals to the Sorting System. 3. Count how many times the system jams/fails to move a card all the way into one of the sorting trays. 4. Calculate the number of jams/number of cards and verify that this is less than 0.15
The Sorting System can move just the bottom card of a stack of cards without affecting the rest of the stack. It will move only 1 card 85% of the time or more.	1. Connect the Sorting System to the Control System. 2. Send randomized sorting signals to the Sorting System. 3. Count how many times the system moves multiple cards at a time. 4. Calculate the number of multiple card moves/number of move signals and

	verify that this is less than 0.15
The Sorting System will move a card fully into one of the trays within 2 seconds of receiving a signal from the Control System.	1. Connect the Sorting System to the Control System. 2. Send a signal to the Sorting System from the Control System and verify that the card is fully sorted within 2 seconds of the signal being sent.

2.2.4 Control System

The control system will be responsible for directing traffic within the device. It will work closely with the user interface and the Raspberry Pi to receive information about the card type and game in order to make a decision about which stack the card should be sorted into. The control system must also work closely with the sorting system, essentially telling it where each processed card should be moved to. Finally, the control system must communicate with the UI early on in this whole process to store which cards we care about and which we do not.

Requirement	Verification
The Control System will give one of two signals to the Sorting System (indicating which direction to move the card) within 1	1. Connect the Control System to both the Sorting System and the Camera System.

second of receiving the card's information from the Camera System.	2. Send card information to the Control System from the Camera System. 3. Measure the time between the Camera System sending the information and the Sorting System Receiving the information, and verify that this is less than 1 second.
The Control System will recognize various errors and send a signal to the User Interface when they occur. The Control System accurately detects Jams (seeing the same card twice) 90% of the time or more.	1. Connect the Control System to the Camera System. 2. Create false jams by showing the camera the same card multiple times. 3. Calculate how many of these false jams the Control System accurately detects and verify that it is over 90%.
The Control System will recognize various errors and send a signal to the User Interface when they occur. The Control System accurately detects Missing Cards (detecting less than 52 cards in the deck) 90% of the time or more.	1. Connect the Control System to the Camera System. 2. Purposefully present the camera system with less than 52 cards and have the camera system go through all of these cards then stop. 3. Repeat this process 100 times. 4. Calculate how many times the Control

	System accurately detects the missing cards and verify that it is over 90%.

2.2.5 Power System

The power system will deliver power to all of the other subsystems. It will use linear regulators to ensure that each component gets the voltage that it needs to operate without negatively affecting the performance of the other components/subsystems.

Requirement	Verification
Provide consistent and accurate voltage to the other components, primarily a 12V ± 0.5V line to turn the motors, a 5V ± 0.5V, and a 3.3V ± 0.3V line for most other items.	1. Connect all subsystems to power, either to the 5V line or to the 3.3V line from the batteries or voltage regulator respectively. 2. Use a multimeter to verify that the 12V line is within its tolerance range, the 5V line is within its tolerance range, and the 3.3V line is within its own tolerance range

2.4 Tolerance Analysis

An aspect of our design that we think poses a notable risk is the Power System. The table below shows some of our components and their typical voltages and currents.

Component Name:	Voltage:	Current:	Comment:
Motor Driver	12V		
Stepper Motor	5-12 V	350 mA	Two motors used, 350 mA current per motor. Using continuous voltage. [1]
Raspberry Pi	5 V	7 mA	Only minimum current listed [2]
Screen	3.3 V	60 mA	[3]
ESP32	3.3 V	500 mA	[4]
White LED	3.3 V	30 mA	[5]

From the table we can conclude we need 2 different voltages: 5 volts and 3.3 volts. To achieve this, we can use a 5 volt battery with a linear regulator translating 5 volts to 3.3 volts.

We can determine the temperature of these linear regulators using a modification of an equation from the wiki [6].

$$T_j = i_{out}\Theta_{ja}(V_{in} - V_{out}) + T_a \quad (1)$$

We'll be using an LM 317 Linear Regulator in a KCS (TO-220) package. Based on this device's datasheet, we can get the junction to ambient thermal resistance value of 23.5 ^{o}C/W for both linear regulators [7]. Similarly, both regulators will have a 12 Volt V_{in} value. Our intended use case for this device is inside a house, so we'll estimate a max air temperature of 80^{o} Fahrenheit or 27^{o} Celsius (rounded up).

We'll use these values to calculate the linear regulator's heat. The following table fills in all of the values for this equation.

Variable:	**Value:**	**Comment:**
i_{out}	590 mA	ESP32, White LED, and Screen in parallel.
V_{in}	5 V	
V_{out}	3.3 V	
θ_{ja}	23.5 ^{o}C/W	[7]
T_a	27 ^{o}C	

Plugging these into our equation we get:

$$T_j = 0.590(23.5)(5 - 3.3) + 27 = 50.571 \quad (2)$$

This gets us a temperature value of just under 51 ^{o}C. This is an acceptable temperature for our product as these temperatures are well below the maximum operating temperature of the LM 317 Linear Regulator (150 ^{o}C) [7]. Additionally, we can move this equation to get our maximum ambient temperature.

$$T_{a\,(max)} = 150 - 0.590(23.5)(5 - 3.3) = 126.43 \quad (3)$$

This indicates that our design should be able to withstand ambient temperatures of up to 126.43 ^{0}C or 259.57 ^{0}F, which is well above any reasonable temperature for our user to be in.

3 Cost and Schedule

3.1 Cost Analysis

Our cost analysis consists of three components. The first is the part list. This includes all items that we had to order to assemble the device, whether purchased from online vendors or ordered within the ECE building. Additionally, the machine shop costs are included in the cost analysis. We reached out directly to ask for a quote. Lastly, we include the total labor cost among all group members.

3.1.1 Part List

Description	**Manufacturer**	**Part Number / Model**	**Quantity**	**Extended Price**	**Link**
Nema 11 Stepper Motor 28mm	Iverntech	Nema 11	3	$51.27	Link
TFT Touch Screen LCD Display Module	Hosyond	ST7796S	1	$19.79	Link
Raspberry Pi Camera Module V2-8 Megapixel	Raspberry Pi	RPI-CAM-V2	1	$14.00	Link
DRV8825 Stepper Motor Driver Carrier	Pololu	DRV8825	2	$25.90	Link

LM1117IMPX-3.3/NOPB Linear Regulator	Texas Instruments	LM1117	1	$1.01	Link
LM1117T-3.3/NOPB Linear Regulator	Texas Instruments	LM1117	1	$1.59	Link
DFR0379 Buck Converter	DFRobot	DFR0379	1	$4.90	Link
Jack Plug Adapter Barrel Connector	California JOS	N/A	1	$3.97	Link
USB Type C Connector Board	Teansic	IC354	1	$7.99	Link
12V 6A Power Supply Adapter	COOLM	YU1206	1	$14.59	Link
Raspberry Pi 4 Model B 8GB	Raspberry Pi	Model B	1	$74.99	Link
ESP32 Microcontroller	Espressif Systems	ESP-WROOM-32	1	$8.99	Link
Total				**$228.99**	

3.1.2 Machine Shops Costs

The majority of the physical design is constructed by the ECE machine shop. The construction of the device is estimated to take 2 weeks working 7.5 hour days. The labor wages for the machine shop are $56.12 per hour. We can find the total Machine Shop Labor cost as follows:

10 days x 7.5 hours per day x $56.12 per hour = $4,209

Total cost for the machine shop labor: **$4,209**.

3.1.3 Labor Costs

All of the members in our group are computer engineering majors. To compute the total labor cost for this project, we will find the expected per hour wage of a computer engineering graduate and multiply it by the expected number of hours we are working on the project.

To find the expected per hour wage, the average starting salary for a computer engineer is $109,176. The total number of hours worked per year is 2,080. The expected per hour wage is calculated as follows:

$109,179 / 2,080 hours = $59.49 / hour

Now, to find the total number of hours we will be working on the project, we can simply multiply the average number of hours we work per week on the project (15) by the total number of weeks spent working on the project (12).

12 weeks x 15 hours = 180 total hours

Finally, multiply these two numbers together, along with 3 since there are three group members.

Total labor cost:

$59.49 / hour x 180 total hours x 3 people = $32,124.60

Total Cost of Project: $228.99 + $4,209.00 + $32,124.60 = **$36,562.59**

3.2 Schedule

Week	**Task**	**Person**
February 23rd - March 1st	Pick up parts for prototyping.	Everyone
	Meet with Gregg and Skee at the machine shop, get them parts to start designing the outer shell.	Everyone
	Research on stepper motors.	Kyle
	Research on computer vision and card identification.	Alfie
	Research on the screen and the ESP32.	Rocky
March 2nd - March 8th	Consolidate research and order more parts when appropriate. (motor drivers, buck converter, power adaptor).	Everyone

	Each member begins board assembly for the specific subsystem they have been assigned.	Everyone
	First round PCB design and ordering (test microcontroller upon delivery).	Everyone
	Continue stepper motor research.	Kyle
	Continue computer vision and card identification research.	Alfie
	Continue screen and ESP32 research.	Rocky
March 9th - March 15th	Each member finalized their breadboard subsystem. Breadboard Demonstrations from March 10th - 12th (Motors, computer vision, screen)	Everyone
	MUST order first/second PCB.	Everyone
	Research OR figure out PCB on chip setup.	Kyle & Rocky
March 16th - March 22nd	Spring break. Individual work week.	Everyone
March 23rd - 29th	Construct a final PCB, last orders	Everyone

	available on March 31st and April 7th.	
	Work with machine shop and their design of the outer shell to plan placement / wiring of each subsystem.	Everyone
March 30th - April 5th	Construct a final PCB, last orders available on March 31st and April 7th.	Everyone
	Add wiring and each specific subsystem to the device shell.	Everyone
	Test the camera subsystem thoroughly.	Alfie
	Test the sorting subsystem thoroughly.	Kyle
	Test the screen subsystem thoroughly.	Rocky
	Finalize the power subsystem and test thoroughly.	Everyone
April 6th - April 12th	Finalize the assembly of the full project.	Everyone
	Test each high level requirement.	Everyone
	Minor bug fixes.	Everyone
April 13th- 19th	Minor bug fixes.	Everyone
	Extend the project somehow if we would	Everyone

	all like to.	
April 20th - 26th	Mock demo.	Everyone
	Work on final presentation, final report.	Everyone
April 27th - May 3rd	Final Demo.	Everyone
	Finalize final presentation, final report.	Everyone
May 4th - May 10th	Win every single award at the award ceremony.	Everyone, including Skee, Gregg, and Sanjana. (Maybe Jack Blevins)

4 Ethics and Safety

4.1 Ethical Concerns

In designing the Automatic Card Deck Sorter, we recognize the responsibility to uphold ethical standards as outlined in the IEEE Code of Ethics. While our device is intended to improve efficiency and convenience in card games, it is important to acknowledge potential misuse and mitigate risk, particularly in gambling or competitive play settings.

A major ethical concern is the possibility of our device being misused to gain an unfair advantage in gambling or competitive card games. To align with the IEEE Code of Ethics [8], we must ensure that our design does not facilitate deception or unlawful conduct. Specifically, Section I.4 emphasizes the importance of maintaining ethical behavior in professional activities and rejecting any form of corruption, including actions that could enable cheating. To uphold these principles, our system will be designed with transparency in mind, ensuring that it functions solely as a fair and unbiased card-sorting tool.

Moreover, Section II.9 of the IEEE Code highlights the responsibility to avoid causing harm to others, whether through direct actions or by enabling unethical behavior. Our project should not be used in a way that compromises the integrity of games, damages reputations, or results in financial harm.

However, since our card sorter is designed for convenience, it will prove difficult to use for other purposes. There are no extreme size restrictions or intentional hidden mechanisms that would be

ideal for the unethical rigging of a card game, so unlawful use of the device is naturally discouraged. We will clearly communicate that the sorter is intended only for *legitimate, fair gameplay purposes*. By addressing these ethical concerns, we ensure that our technology aligns with professional integrity and responsible engineering practices.

4.2 Safety

Ensuring safety in the design, development, and use of the Automatic Card Deck Sorter is a priority. Our project involves low voltage electrical components and mechanical moving parts, both of which present potential risks. To address electrical safety, we will use proper insulation, secure wiring, and fuse protection to prevent short circuits, overheating, or other hazards. We are not working with high voltage, but we still will follow standard lab electrical safety protocols. Mechanical safety concerns will be mitigated by using low torque motors to minimize the risk of pinching hazards. The device is also not designed to use any sharp objects.

In compliance with ECE lab safety guidelines, we will always work in a group of at least two. Proper handling of soldering stations, power tools, and electronic testing equipment will be observed. For end-user safety, we will ensure a stable structure, low-voltage power supply, and no exposed sharp parts. While the overall safety risks are relatively low, our plan includes adherence to lab protocols and a low risk design to ensure a safe and reliable automatic card sorter.

5 References

[1] Adafruit, “1.8º 42MM Torque Hybrid Stepping Motor”, XY42STH34-035A datasheet

[2] Raspberry Pi, “Raspberry Pi 4 Model B”, June 2019 [Revised March. 2024]

[3] Display Visions, “DOGM 132 Graphic” 132x32 INCL. CONTROLLER ST7565 datasheet

[4] Espressif, “ESP32 Series Datasheet Version 4.8”, Aug. 2016 [Revised Jan. 2025]

[5] Vishay, “Ultrabright White LED 5mm Untinted Non-Diffused Package”, VLHW5100 datasheet [Revised May 2019]

[6] Engineering IT Shared Services, “:: ECE 445 - Senior Design Laboratory,” *Illinois.edu*, 2025. https://courses.grainger.illinois.edu/ece445/wiki/#/regulators/index (accessed Feb. 13, 2025).

[7] Texas Instruments, “LM317 3-Terminal Adjustable Regulator”, SLVS044Y datasheet, Sept. 1997 [Revised Apr. 2020]

[8] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).