Efficient Card Shuffler

ECE 445 Team 53

Alex Lo Faso, Steve Mathew, Matt Garrity

April 30 2026

Problem

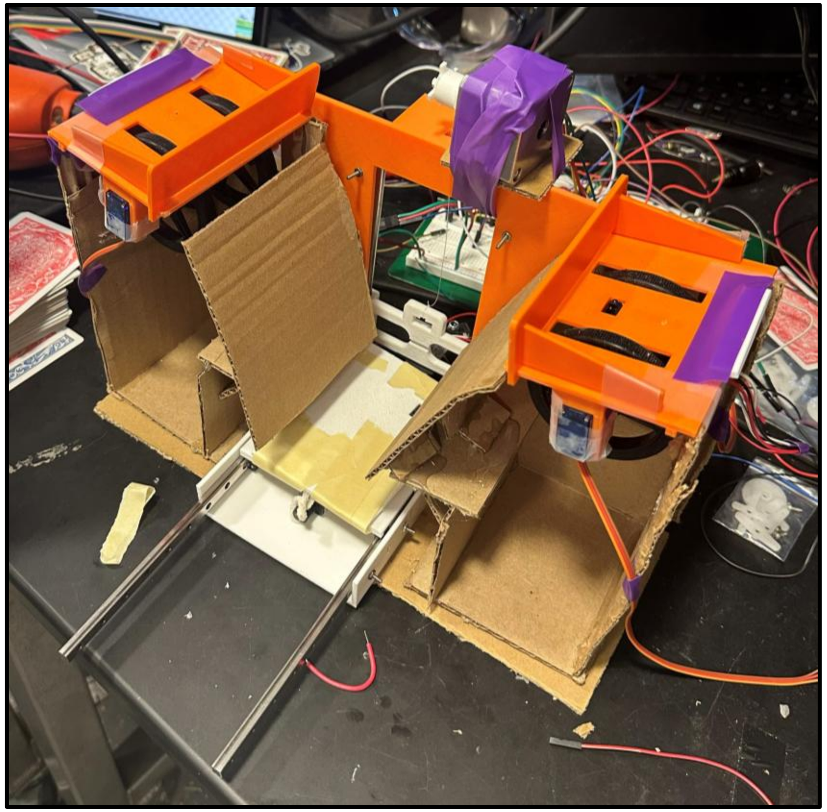
Card Shufflers Today

- Limited in deck size capacity
- Require significant human intervention
- Don't accommodate games requiring insertion of cut card
- Manual retrieval of shuffled cards

Solution

Our Solution

- Build a card shuffler with increased capacity of 4-6 decks
- Build automated shuffled card tray
- Build cut card insertion feature with customizable deck penetration



Design Overview

Design Overview

Machine Operation

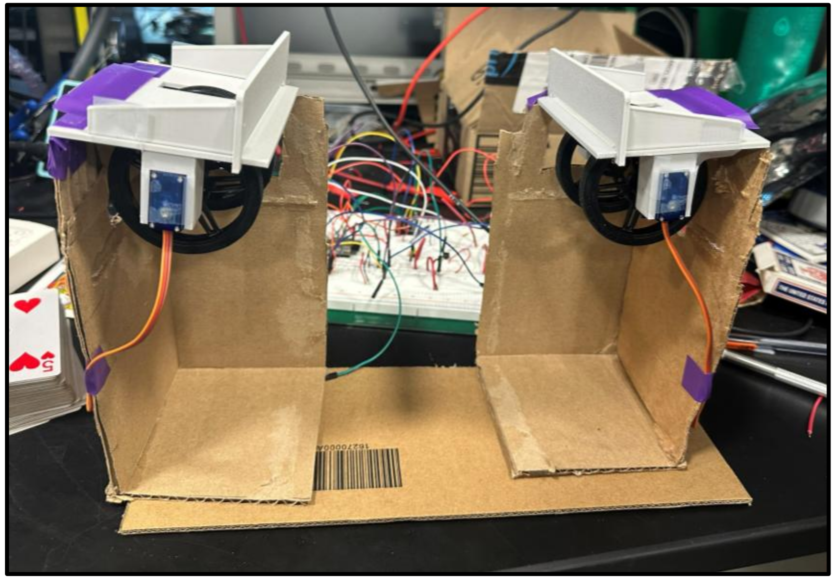
- Place cards to be shuffled in pre-shuffle trays
- Set cut card slot height using physical dial and press shuffle button to begin operation
- Insert cut card into card stack after shuffle operation is complete
- Tray extends once cut card is inserted
- Tray retracts once cards are taken out, ready for process to repeat

Subsystems Design

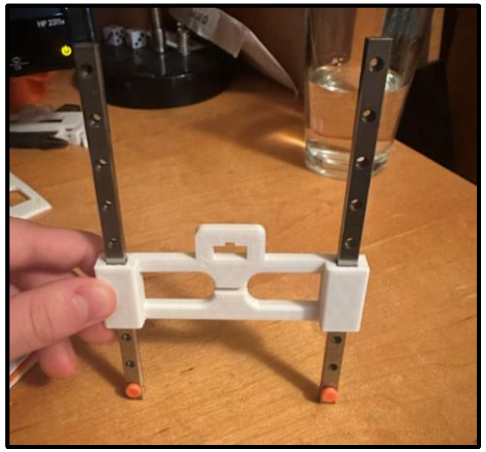
1. Card Deck Detection
2. Deck Shuffling Mechanism
3. Cut Card Insertion
4. Deck Tray Extension
5. Power Supply
6. Control Module

Physical Build

- Materials
    - Plastic(PLA), cardboard, metal(rails)
- Assembly
    - Hot glue, screws, tape
- Inputs
    - Pushbutton, Dial



Changes to Design

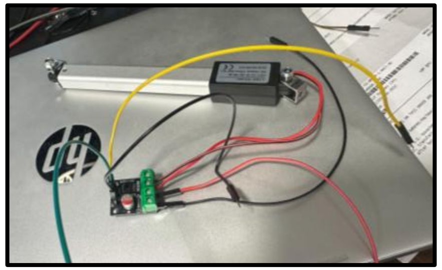


Beam-break Sensor Removal

- Original: Beam-break sensors send pulse to MCU for jam detection
- New: Realized this wasn't necessary since jams would be visually identified

Cut Card Slot Vertical Adjustment Redesign

- Original: Lower slot to trigger limit switch, then slot would raise to predetermined level
- New: Visually raise slot with live dial; allows for live readjustment and deck size variability



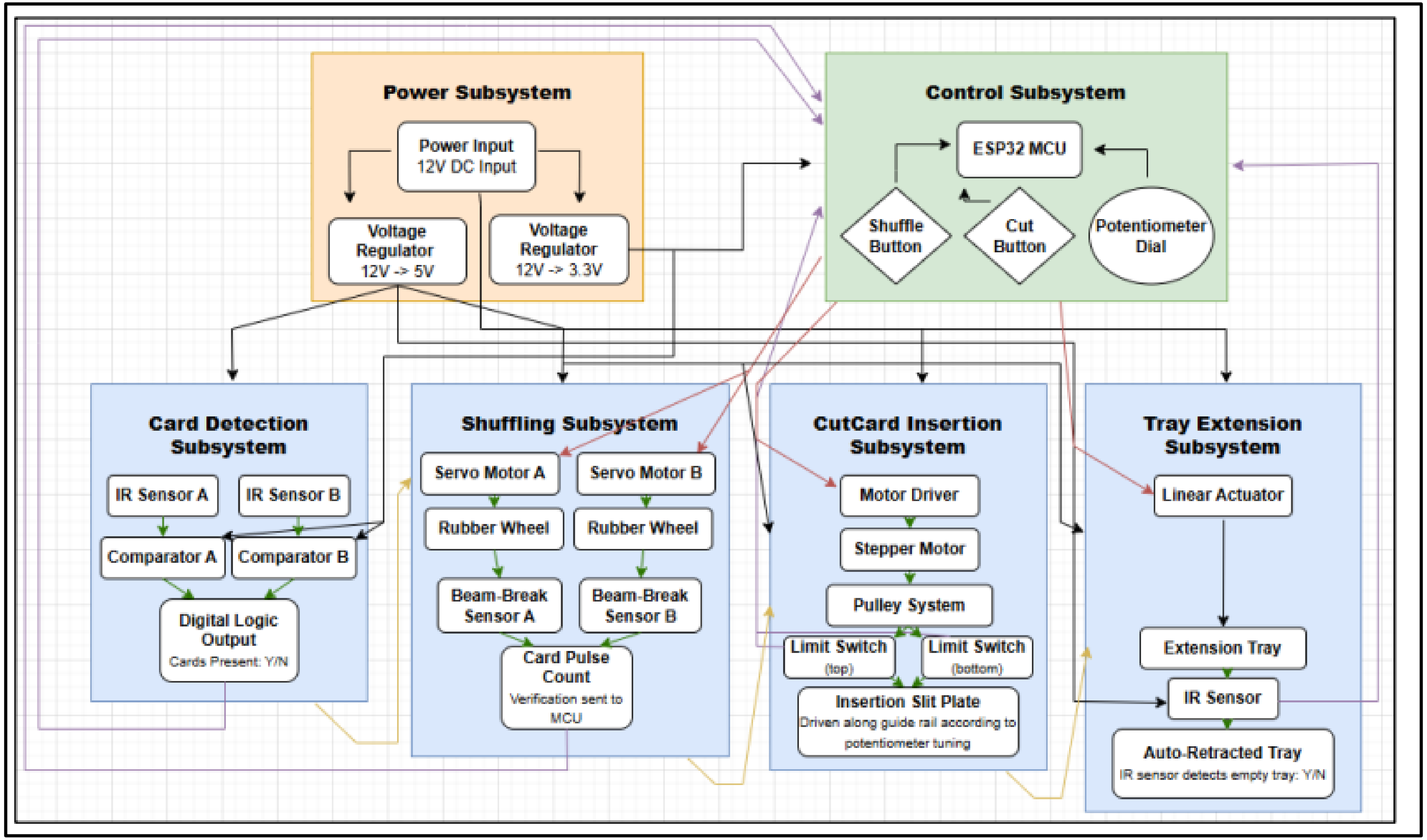
Linear Actuator Motor Driver

- Added motor driver to interface MCU with linear actuator

3D Printing

- Became apparent more parts needed to be 3D printed to make parts more rigid and subject to less unintentional movement

Block Diagram



Power lines

Signal Lines

Data Lines

Card Progression

System 1: Card Deck(s) Detection

Overview:

Use IR sensors to detect card presence in trays

No input = HIGH

Input = LOW

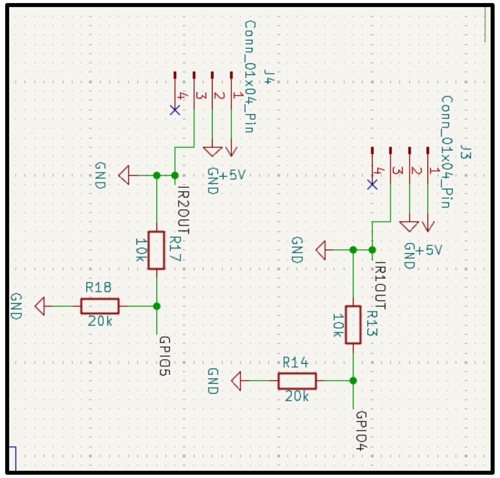
Requirements	Verification
• The subsystem must have a latency of less than 200ms to ensure the MCU can halt the motors immediately upon the last card being drawn	• 1. Connect Oscilloscope to the sensor's Data out pin • Set the oscilloscope to 'Normal' trigger mode on a falling edge • Rapidly remove a card from the sensor's field of view • Measure time form physical removal to signal spike
• Sensors should be able to identify card presence accurately. Comparator outputs a "high" level when cards are present and "low" level when cards are not present.	• Place cards in pre-shuffle trays and verify the comparator analog output is greater than 2 volts. Remove cards from pre-shuffle trays and verify the comparator output is less than 0.8 volts. • This can be measured with multimeter from the data output pin of comparator
• Comparator must produce output signals within an interpretable and safe range to the ESP 32	• Using multimeter, ensure that the comparator low value outputs are within 0 < value < 0.8 volts and that the high values are 2 volts < value < 3.3 volts.
• The comparator threshold must be adjustable to account for different card back colors (red / blue)	• Place a dark-colored card in the tray and measure output voltage. • Place a light-colored card in the tray and measure output voltage. • Verify that a stable Logic HIGH is maintained for both colors.

Key Challenges & Verification

- Sensitivity to light
- Stable mounting to shuffle trays

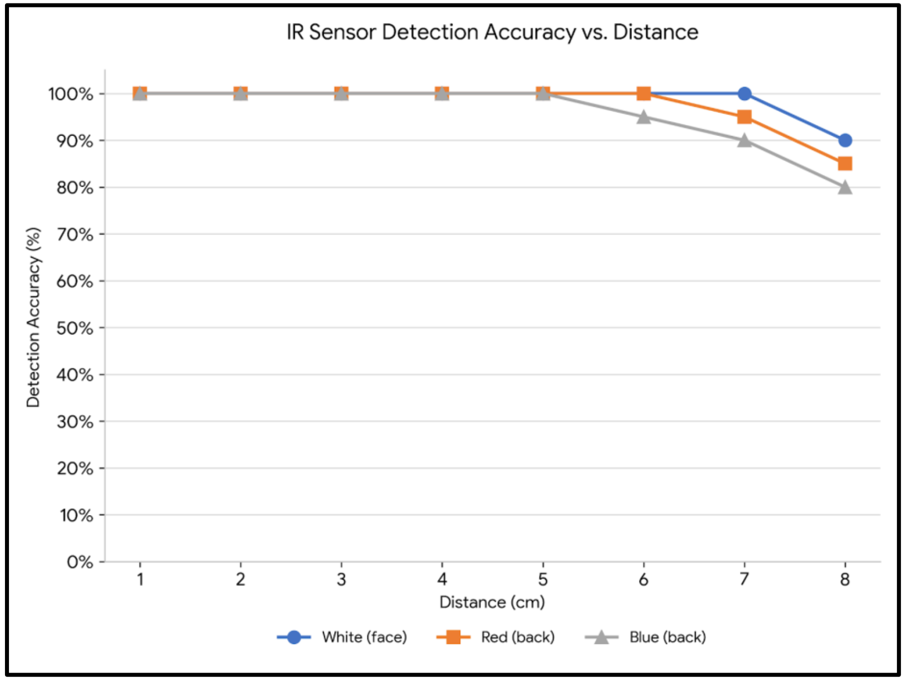
Solutions:

- Tuned sensor to only respond to severe lack of light (complete card blocking)
- Iterated through several designs, to add mount to tray allowing for sensors to be screwed in



High Test Values	Low Test Values
3.21v	0.112v
3.18v	0.113v

System 1: Card Deck(s) Detection Results



System 2: Deck Shuffling Mechanism

Overview:

Use 2 pairs of FSR servo motors to propel cards downward with wheels

Once cards are launched forward on tray, they pass through 3D printed slot into funnel

Requirements	Verification
• Motor wheel must advance one card per motor actuation under nominal conditions to ensure acceptable shuffling (success rate > 90%)	• Place a deck of red colored cards in one pre-shuffle tray and a deck of blue colored cards in the other pre-shuffle tray. After shuffling operation, examine how well the cards are mixed • Repeat operation several times to ensure consistency
• Shuffle operation will not begin until collection tray is fully retracted.	• Visually verify that the collection tray is fully retracted before the cards begin shuffling.
• Beam-break sensors must generate a pulse for at least 1ms duration so that it can be detected by MCU. Shuffling should pause in event of a jam.	• Connect oscilloscope to beam-break signal output pin • Slide card manually through dispensing slot at anticipated motor speed • Capture pulse width on oscilloscope and verify if pulse > 1 ms
• Servo motor torque must be able to overcome card friction without deteriorating cards	• Examine quality of the cards every 2 shuffle operations to ensure no damage is being done. • Confirm no anomalies and repeat procedure several times

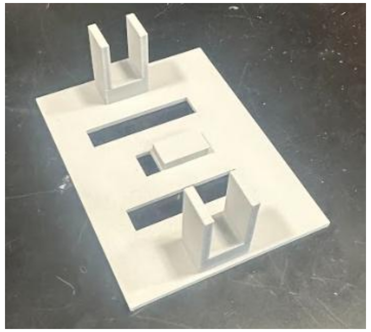
Key Challenges & Verification

Problems:

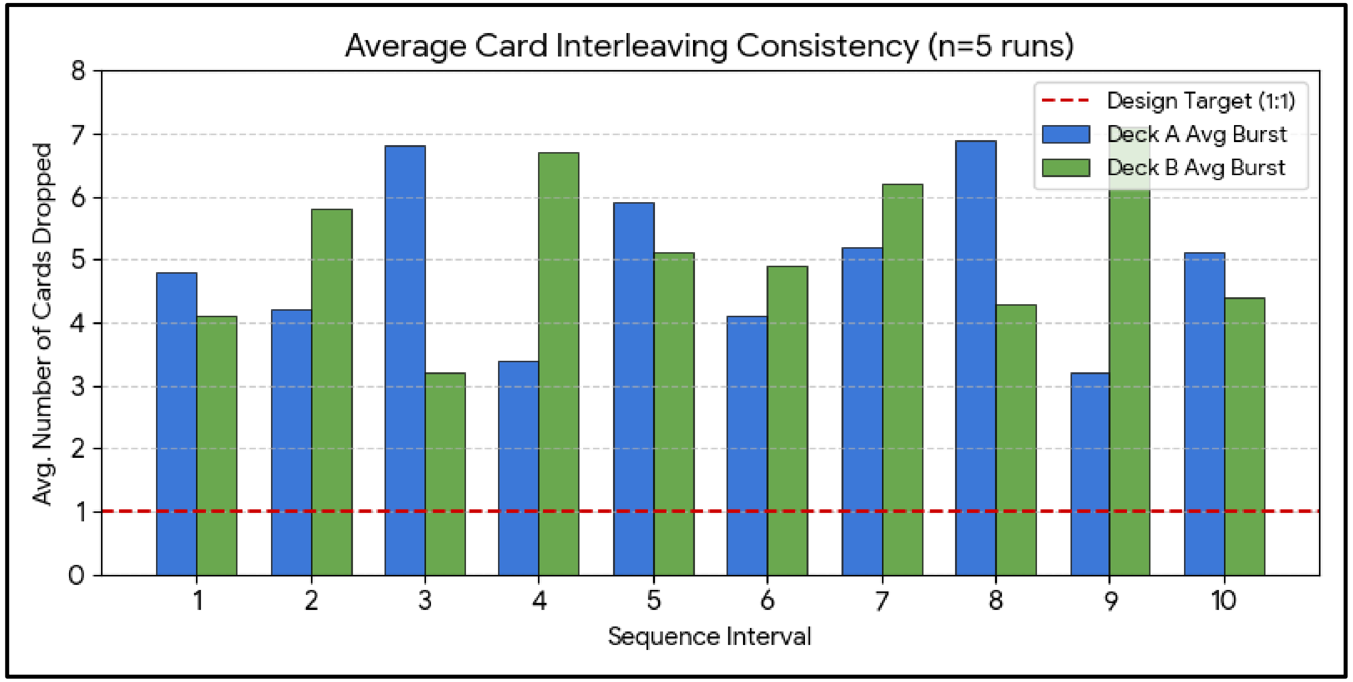
- Slot creation
- Friction between tray & slot
- Lack of card fall organization

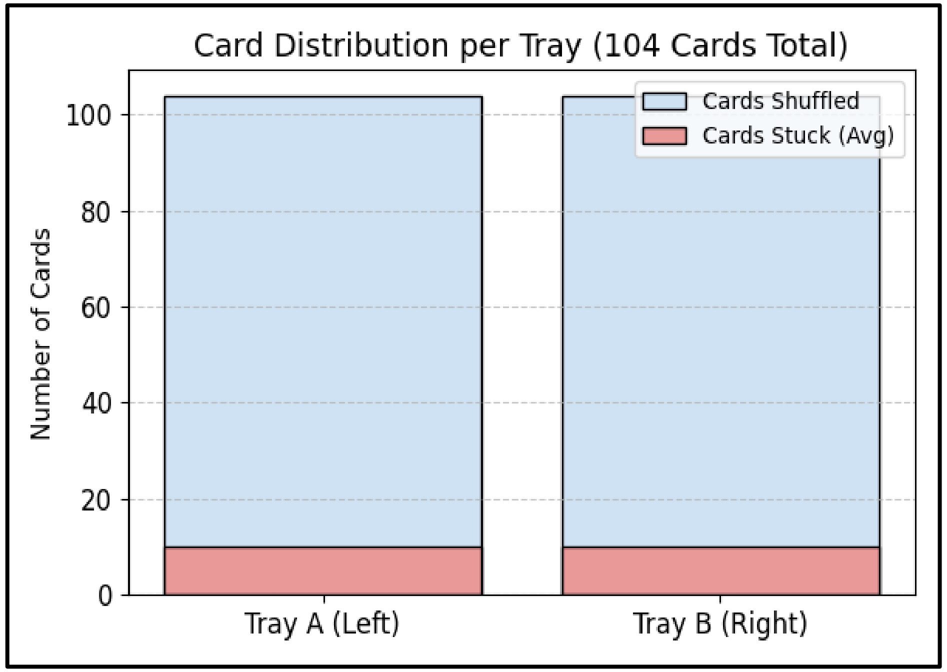
Efforts:

- 3D printed several slot/tray iterations
- Designed sloped edges to reduce friction
- Attempted funnel enclosure



System 2: Deck Shuffling Mechanism Results





Average Deck Shuffle Burst Range (cards): 4.8 – 5.2

Successful Feed Rate: ~86%

System 3: Cut Card Penetration

Overview:

User turns rotary encoder to preferred level

Cut card slot follows accordingly, allowing for any depth insertion

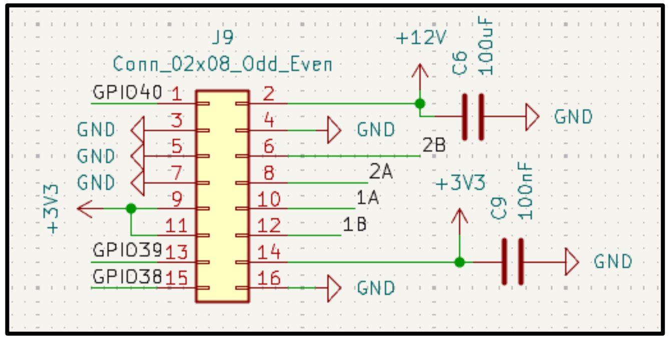
Requirements	Verification
• Rotary Encoder input must be able to map deck depth from 10 – 90% at increments of 4% to ensure user input is properly mapped	• Rotate encoder by set detents • Measure the height of the shuffled decks with a ruler, multiply this measurement by 0.1. Ensure that the cut card slot is above this measurement + bottom of deck and below the top of decks – measurement
• Limit switches must trigger within reasonable time when contact is made from lowering of the pulley system (3.3V to 0V within 1mm of physical contact)	• Use multimeter to monitor signal pin of limit switch • Lower the plate until the limit switch clicks • Verify that there is a voltage drop • Measure gap between plate and ground base to confirm tolerable distance
• Slot travel along the vertical axis must remain free from jams and the insertion window should remain relatively flat	• Command plate to move to highest position • Use ruler to measure height from bottom on left and right sides • One side of the slot should not be more than 0.5cm higher than the other. Any impediment to movement along the vertical axis should be considered a jam.

Key Challenges & Verification

- Tuning the rotary encoder rotation to the corresponding motor spin
- Fabricating pulley system and incorporating it with design

Solutions:

- Developed software solutions to work around 'ticking' logs of rotary encoder
- Ensured complete vertical alignment in pulley design and assembly



System 3: Cut Card Penetration Results

PREDICTED (DESIGN DOC)

Deck height (6 decks)	94.0 mm
Mechanical slippage	1.50 mm
Rail stitching effect	~1.00 mm
Encoder error (4% × 94 mm)	3.76 mm
Total estimated error	6.26 mm
6.26 ÷ 94.0	= 5.5%

Acceptable threshold: ≤ 5% (margin: +0.5%)

ACTUAL (MEASURED)

Usable travel (10%-90%)	80%
80% × 94.0 mm	= 75.2 mm
Encoder ticks (10%→90%)	17 ticks
75.2 mm ÷ 17 ticks	= 4.42 mm/tick
80% ÷ 17 ticks	= 4.71%/tick
4.42 mm ÷ 94.0 mm	= 4.71%/tick

Resolution finer than predicted error → 1 tick corrects any drift

Metric	Predicted	Actual	Result
Max linear error	6.26 mm	4.42 mm/tick	✓ Within budget
% error / resolution	5.5%	4.71%/tick	✓ Aligns
Encoder corrects predicted drift?	4.71% < 5.5%	Yes (1 tick)	✓ Verified

System 4: Deck Tray Extension

Overview:

IR sensor pushes out tray once all cards have funneled down

Retracts once cards have been retrieved by user

Requirements	Verification
• Optical sensor must detect deck removal to allow for repeated shuffling processes (maintain logic HIGH vs logic LOW signals at appropriate settings)	• Use multimeter to measure sensor output voltage when cards are present (should read > 2.4V • Use multimeter to measure sensor output when cards are removed (should read < 0.8V) • Ensure timing difference between removal and voltage drop is below 100ms
• Tray needs to extend once shuffling operation is concluded.	• Visually ensure that the cards have cleared the feed trays before the bottom tray extends with the collected deck. • Time the gap between the cards clearing the feed trays and the collection tray extending to find out if there is the proper amount of delay. • Confirm time taken is not tedious
• Tray needs to fully extend out of the body to allow for removal of the cards. (to full extent of linear actuator)	• Measure the extension of the tray with a ruler and verify that it is extended out far enough for easy removal of the shuffled deck. • This will be done visually and through use of human interaction

Key Challenges & Verification

- Ensuring IR sensor wouldn’t pick up internal walls
- Ensuring tray moves out smoothly

Solutions:

- 3D-printed a mount that used a pipe cleaner to connect allowing flexibility in movement
- Software solution to avoid IR conflicts & improper behavior

System 4: Deck Tray Extension Results

Systems 5&6: Power + Control

Overview:

Allow proper power delivery to all electronic components (12V / 5V / 3.3V)

MCU should receive and send data to all subsystem through GPIO commands

Requirements	Verification
• Voltage outputs are of correct magnitude and within a reasonable level of tolerance o Ex. 3.3V rail must maintain voltage of within 5% of 3.3V under a max load of 500 mA to power the ESP32 and other sensor lines	• Use multimeter to measure output voltage at various points along the line • Ensure readings are within tolerance threshold
• The 5V rails for the servo motors must maintain proper powr distribution when all 4 motors are in rotation	• Connect oscilloscope to 5V power line • Activate all 4 servos simultaneously (shuffling cycle) • Monitor voltage to see if there is no voltage dip • Ensures that power dist. Is able to deliver at peak distribution for motors
• Traces routed from this subsystem are required to adhere to the correct minimum thickness to allow for proper power distribution	• Examine PCB design to ensure traces are of correct thickness. Measure voltage inputs at sensors and components to ensure they have met requirement.

Requirements	Verification
• Commands from input buttons and dials are correctly received and able to communicate across all subsystems	• Using a multimeter, measure the output of the buttons when they are pressed and not pressed and ensure reading is corresponds to action
• Must be able to continuously operate throughout multiple shuffling cycles to ensure proper operation (State Lockout)	• When shuffle cycle is in operation, pressing the buttons should have no impact on the machine • Ensure that no behavior is changed when a current cycle is in mode
• Read and store pulses from rotary encoder accurately	• Open digital monitor and test rotary encoder movements • Measure degree of turn and confirm it aligns with pulses

Key Challenges & Verification

Problems:

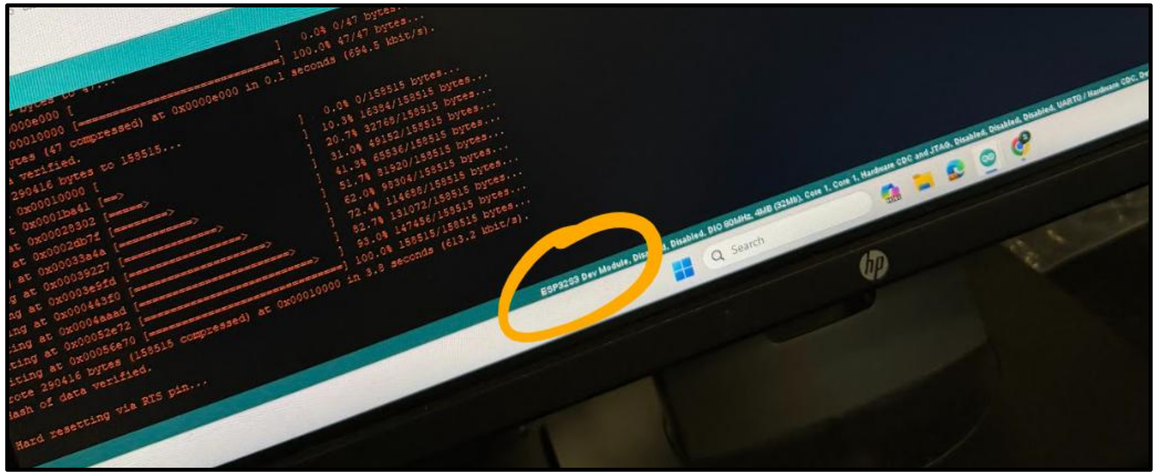
- Breadboard had issue with the 12V battery + buck converter setup (instantaneous current spike)
- PCB needs separate power lines for 12V, 5V, and 3.3V
- MCU must communicate to 25+ ports and handle real-time interrupts

Solutions:

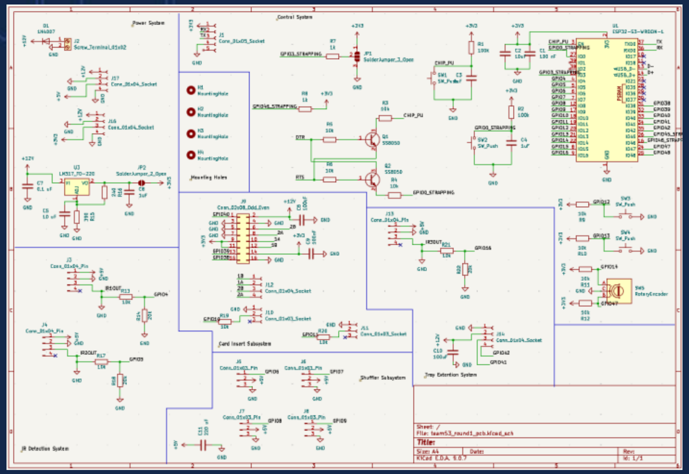
- Utilized 2 buck converters to step down voltage from 12V to 5V and 12V to 3.3V
- Processed software development to handle real-time user input and log it to serial monitor
- Routed PCB traces to allow for proper communication to all sensors, motors, and drivers

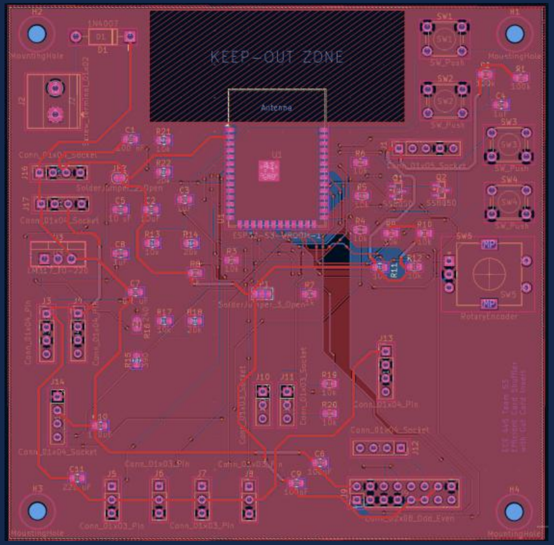
System 5&6: Power + Control Results

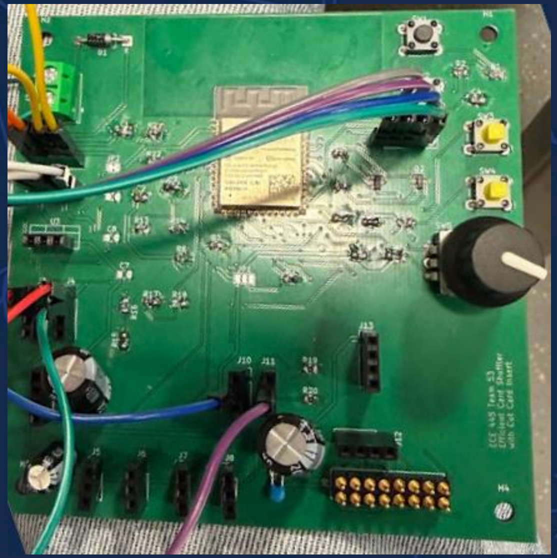
```
// --- STATE MACHINE ---
enum SystemState {
  IDLE,
  SHUFFLING,
  WAITING_CARDS_EMPTY,
  ACTUATING_OUT,
  ACTUATING_IN
};
SystemState currentState = IDLE;
```



PCB Implementation



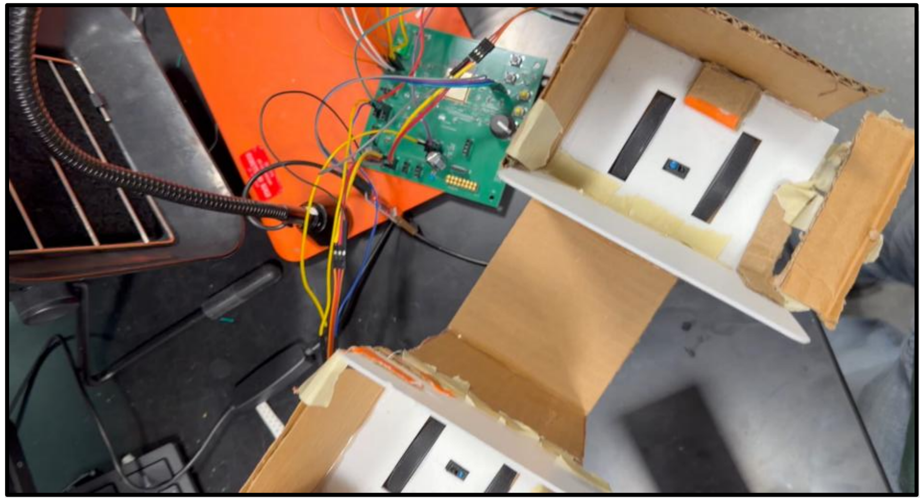


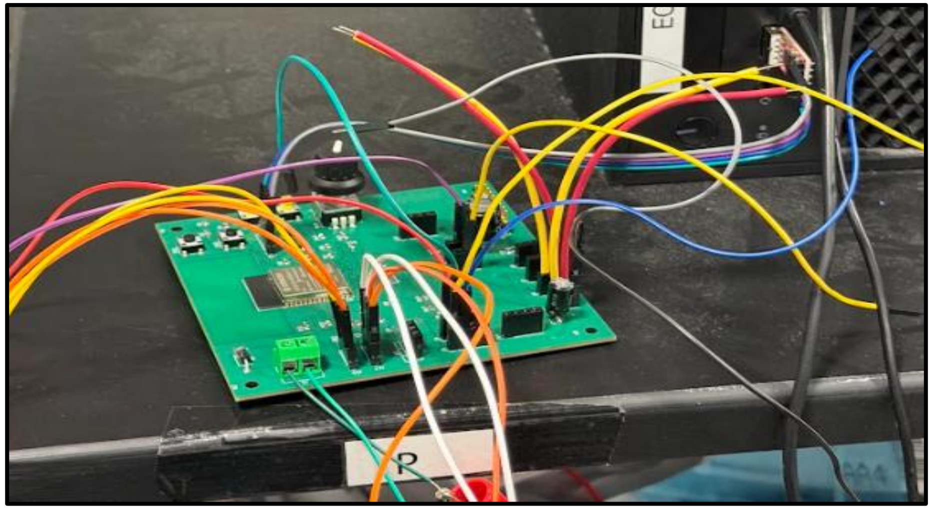


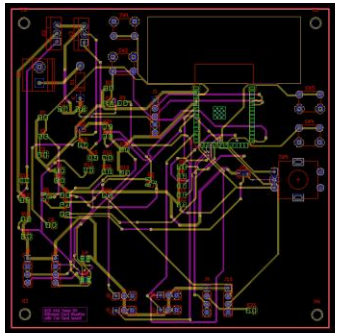
PCB Functionality + Failure

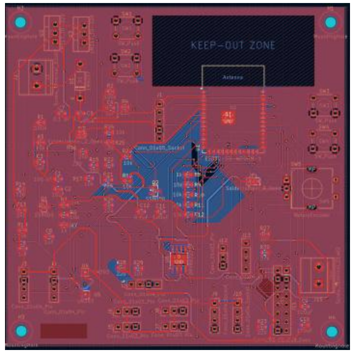
- 3 PCB designs
  - Physically implemented Round 4 design with 2 boards
  - Round 4 v1 integrated with subsystems 1, 2, 5, & 6

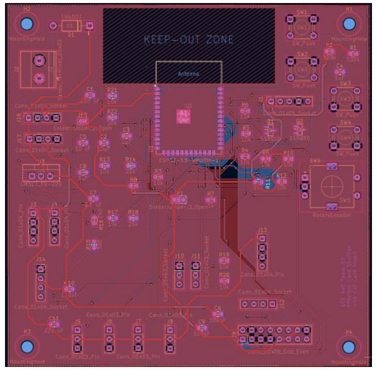
  - MCU internal short-circuit issues
  - Programming issues with v2







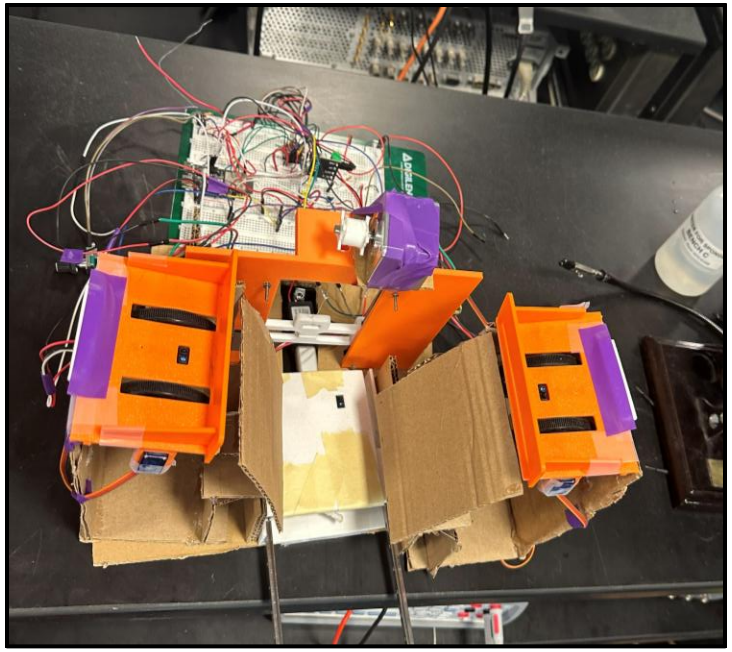




Conclusion

What we Learned

- Independent testing of PCB components
- PCB design practices
- Integration of parts sequentially into a physical design
- Importance of part care and management



Do differently / Further Work?

- **General**
  - Start physical build earlier
  - Make a test PCB and a final PCB
- **Specific**
  - Build a tighter funnel channel to receive cards after they exit shuffle trays
  - Use a stepper motor with smaller steps to allow for even better deck penetration adjustments
  - Put a ramp on pre-shuffle trays to eliminate last few cards from hanging
  - 3D print all parts and create professionally packaged look, refine part iterations

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN