Magic: The Sortening

Team 24

FPR

Presentation & Demo

Zalman Lipschitz (MechE)

Prof. Mario Parente

Joseph Maloyan (CompE)

Malcolm Okaya (CompE)

Henry Powell (EE, MechE)

Liam Rees (MechE)

Problem statement

The secondhand market for Magic cards is very large, but dealing with cards in bulk requires significant time, manual effort, and money to sort and check for damage. Current automatic sorting solutions, which are themselves quite limited, can’t check for damage properly. Our solution is able to take a large stack of cards as an input, detect and grade each card’s condition, identify the cards, check for forgeries, and sort them into several configurable bins. This allows users to properly value and sort large quantities of cards.

System Specifications/Goals

Item	Description/spec
Card Throughput	>2 cards/min, >1000 card capacity
Card Condition Grading	Catch signs of damage. Provide a valid grading based on that damage.
Card ID	Solution must find the name of card as well as the set it comes from. >95% accuracy
Forgery Detection	Customer asked for 4 benchmarks to detect fakes. Solution must perform all 4 checks.
Card Damaging	No noticeable damage to cards.
Output Bins	Need at least 15 software-configurable output bins

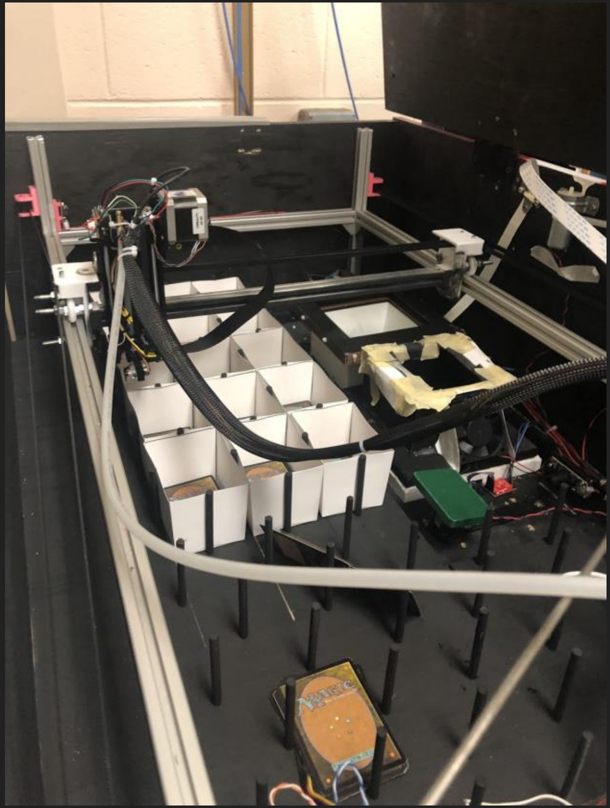
Recorded Demo 1 (Backup)

Recorded Demo 2 (Backup)

Final System Performance

Physical Specs

- 40” x 25.5” x 13.5” with lid closed, 40” x 25.5” x 37” with lid open
- ~60lb dry-weight (no cards loaded)
- 3500 card capacity
- 50-60 dB running sound level, depending on the stage.
- 120VAC input



Final System Performance

System Specification Compliance - Summary

Item	Description/spec	Met?
Card Throughput	>2 cards/min, >1000 card capacity	Process speed too slow. 3500 card capacity.
Card Condition Grading	Catch signs of damage. Provide a valid grading based on that damage.	Partially.
Card ID	Solution must find the name of card as well as the set it comes from. >95% accuracy	Yes. Not statistically tested.
Forgery Detection	Customer asked for 4 benchmarks to detect fakes. Solution must perform all 4 checks.	No, while the system is physically capable, the subsystems are not integrated.
Card Damaging	No noticeable damage to cards.	Partial. Some wear to cards
Output Bins	Need at least 15 software-configurable output bins	Yes.

Card Throughput Goal

Goal:
>2 cards/min, >1000 card capacity

Reason:
Requested by vendor. About what a human person is capable of.

Verification:
Basic demonstration of each. Fully process 2 cards in 1 minute.

Card Throughput Result

Result:
While we do have the capacity for 3500 cards, we are unable to meet the required speed. It takes ~70 seconds to complete a full processing cycle. This is largely a function of the computer’s processing speed and the camera focusing speed.

Verification:
Video Attached

Condition Grading Goal

Goal:

Find >95% of damage visible to human eye. 90% confidence.

Sides

Front

Back

Reason:

Key goal of this project is finding and evaluating damage to cards.

Verification:

Manually map all damage on digital image of card. Compare with computer-generated damage map. Do this for front/back/sides. 19/20 pieces of damage visually identified should also be identified by the computer. Do this 10 times.

Condition Grading Result - sides

Result: Our program is capable of finding damage on the edges of cards, and categorizing cards into 6 different levels of quality.

Verification:

- **False Positivity rate: Average Number of false positives per card, checked 10 cards**
  - Important as we do not want to mark clean marks as damage
  - ~0.1 False positives per card
- **Eye Test: How well do raw scores for damage agree with grades?**

Final System Performance

Condition Grading Result — Side Damage Detection Images

False Positive

86 -> 5

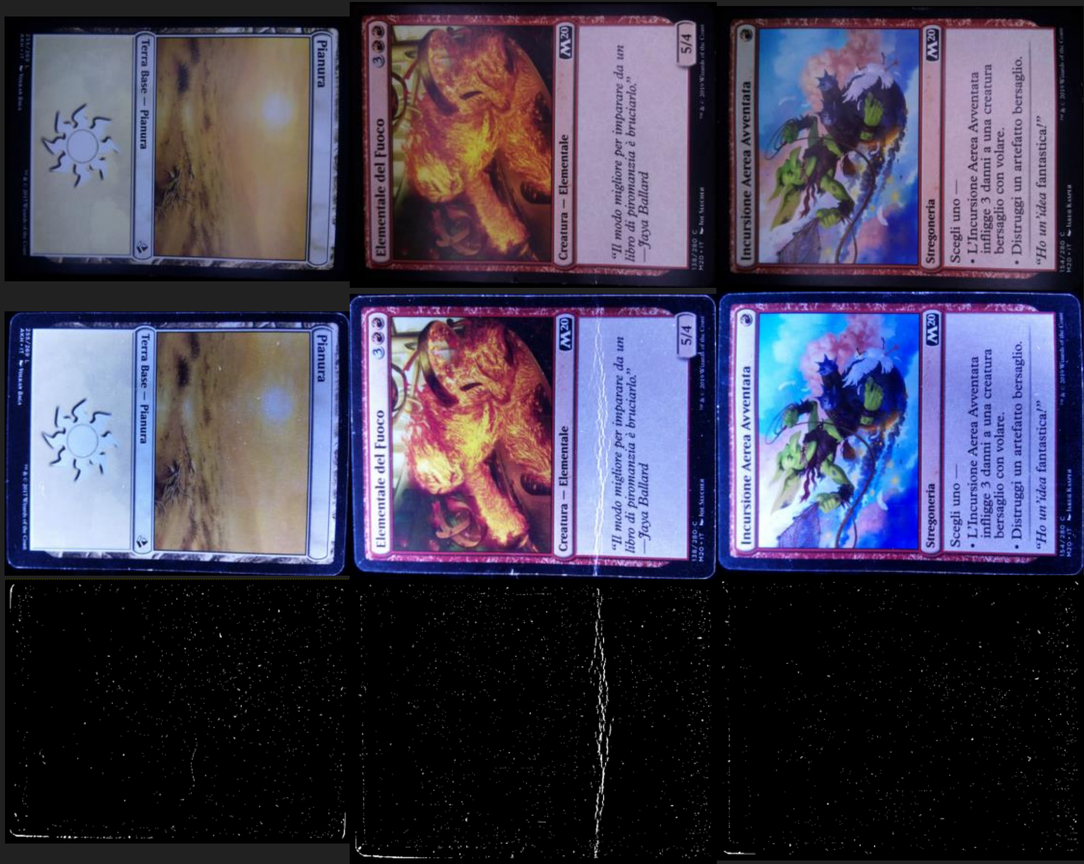
518 -> 4

60277 -> 1

Condition Grading Result - front

Result: We have the ability to find and map damage to the card surface, but we aren’t very confident in our ability to filter out dust and to generate a good grade from that map.

Verification:
Visual comparison with 2 pre-graded cards. Able to find the large bend and more major damage.



Condition Grading Result - back

Result: The evaluator for the back-side of cards is built and functional, but we were having issues with image quality and integration. It is not operating in the final machine.

Verification:

N/A

Card ID Goal

Goal:
Correctly identify cards 95% of the time. 95% certainty.

Reason:
Second key goal of this project is sorting cards correctly.

Verification:
Run 20 cards through identification. No more than 1 card misidentified. Do this 20 times. Unidentifiable cards that are correctly judged as such don’t count.

Card ID Result

Result:
We managed to identify 18/20 cards in our latest test of the card ID system.
Setup is not robust enough for the full 95% confidence test.

Verification:
18/20 = 85% of cards correctly identified.

Forgery Detection Goal

Goal:
Weight check
Glossiness check
Printing verification check 1 (print pattern)
Printing verification check 2 (dots)

Reason:
Third key goal of this project is finding forgeries. Vendor mentioned each of these checks.

Verification:
90% of legitimate cards pass all 4 tests. 90% of fake cards fail at least one. 90% confidence: Run through 10 of each, 10 times. No more than 1 false positive/negative each run.

Forgery Detection Result — Weight

Result:
The load cell is integrated into the back-side evaluator. Because the back-side evaluator is not enabled, the weight detection is not working.

Verification:
N/A

Forgery Detection Result — Glossiness

Result:
The glossiness evaluation was abandoned.

Verification:
N/A

Forgery Detection Result — print pattern

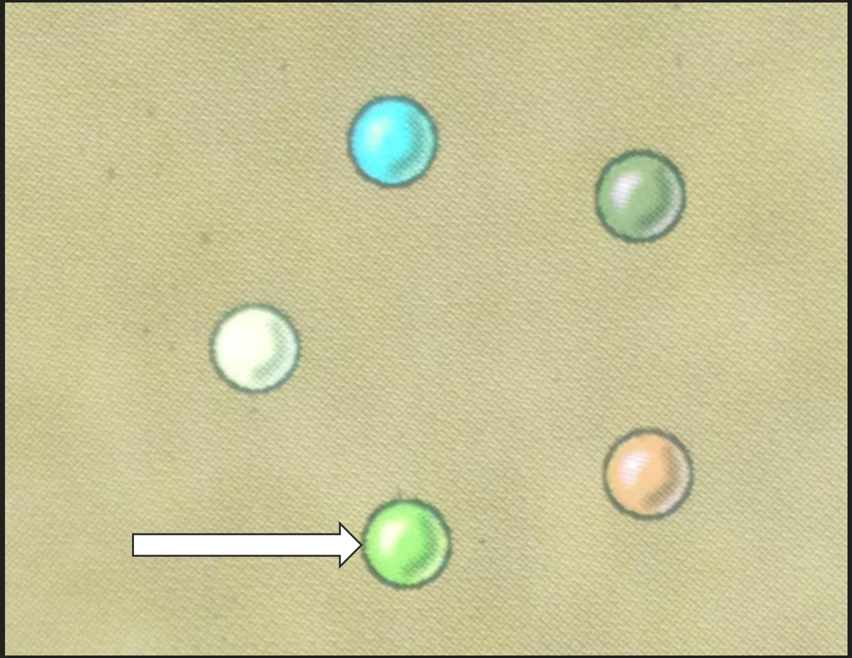
Result:
While we believe we have built a system that might be capable of forgery detection, it isn’t complete on the final machine.

Verification:
Print pattern is very clear with our camera setup

Forgery Detection Result — dots

Result:
The back-side evaluator is built and functional, but is not enabled on the final machine. Aforementioned camera quality issues make it very hard to see the dots

Verification:
See image



Card Damaging Goal

Goal:
Cause minimal damage to cards through processing

Reason:
Want to have client trust. Damaging the cards hurts their value.

Verification:
Run 1 card through 5 times. Damage detection software should not detect more than 5% additional damage. Visual inspection should also show no damage.

Demonstrate fail-safe by unplugging mid-operation and showing no fires.

Final System Performance

Card Damaging Result

Result:

Minor wear to cards that are run through the system

Verification:

Below card was run through the system 5 times, 1 new piece of damage detected by the end, grade dropped from a 5 to a 4

Output Bins Goals

Goal:
At least 15 configurable input/output bins.

Reason:
Distributor identified maximizing number of possible outputs as very desirable.

Verification:
Demonstrate system is capable of accessing each bin. Demonstrate that they may be configured.

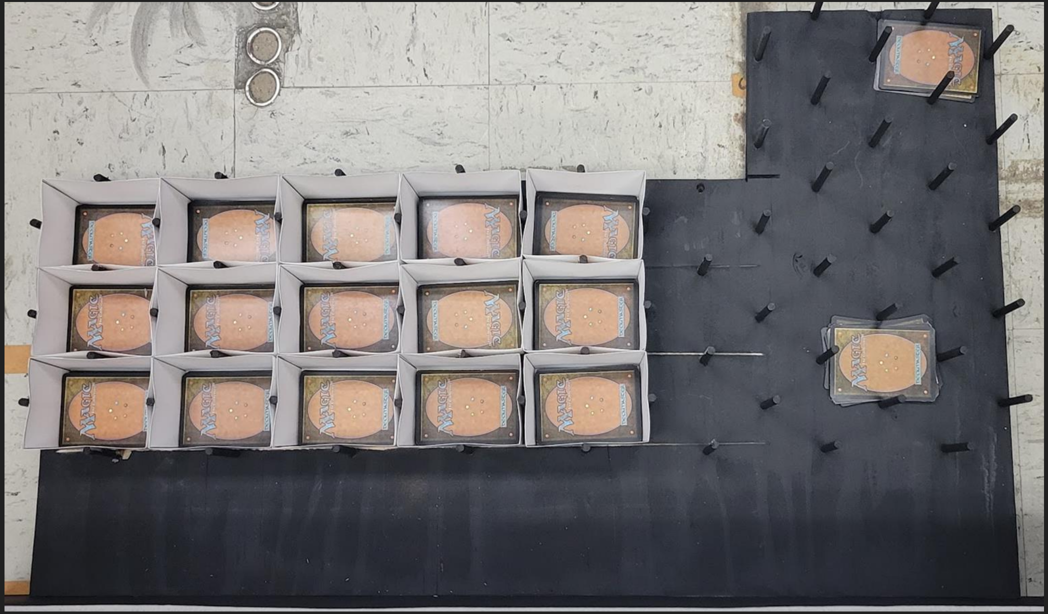
Output Bins Result

Result:

We have well over 15 functional input/output bins, though the interface to configure them is not fully functional.

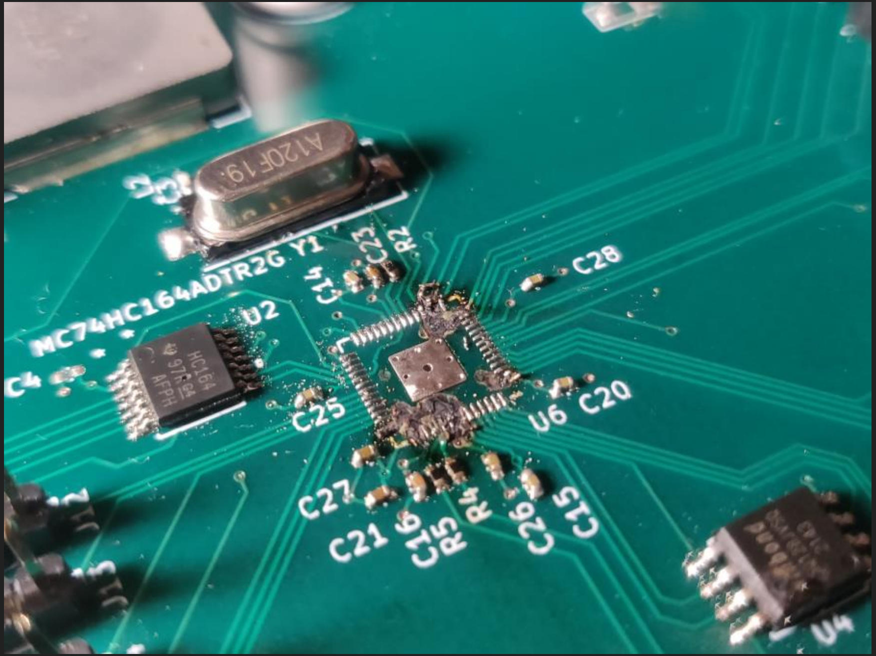
Verification:

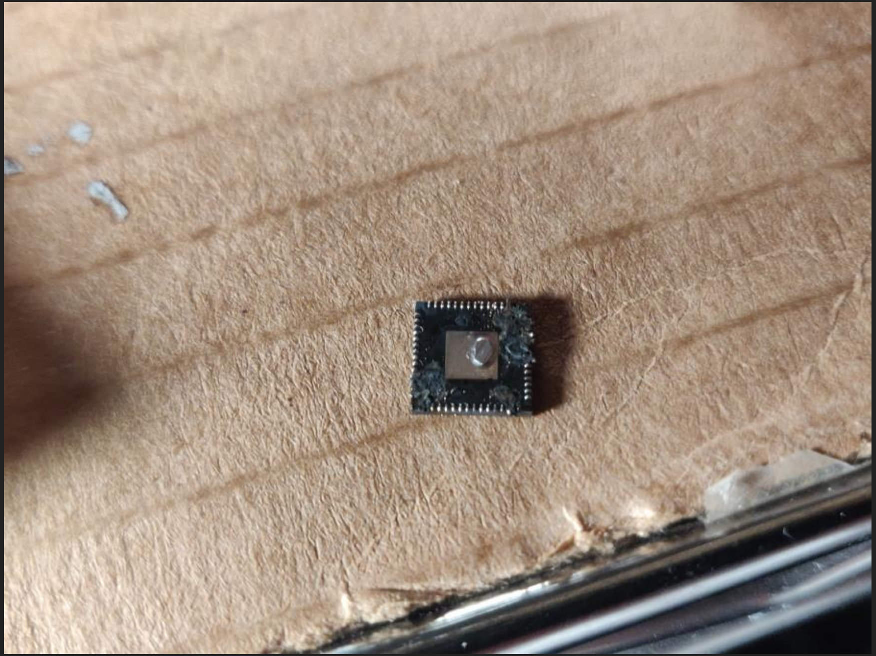
Count them! 28 total



PCB Performance

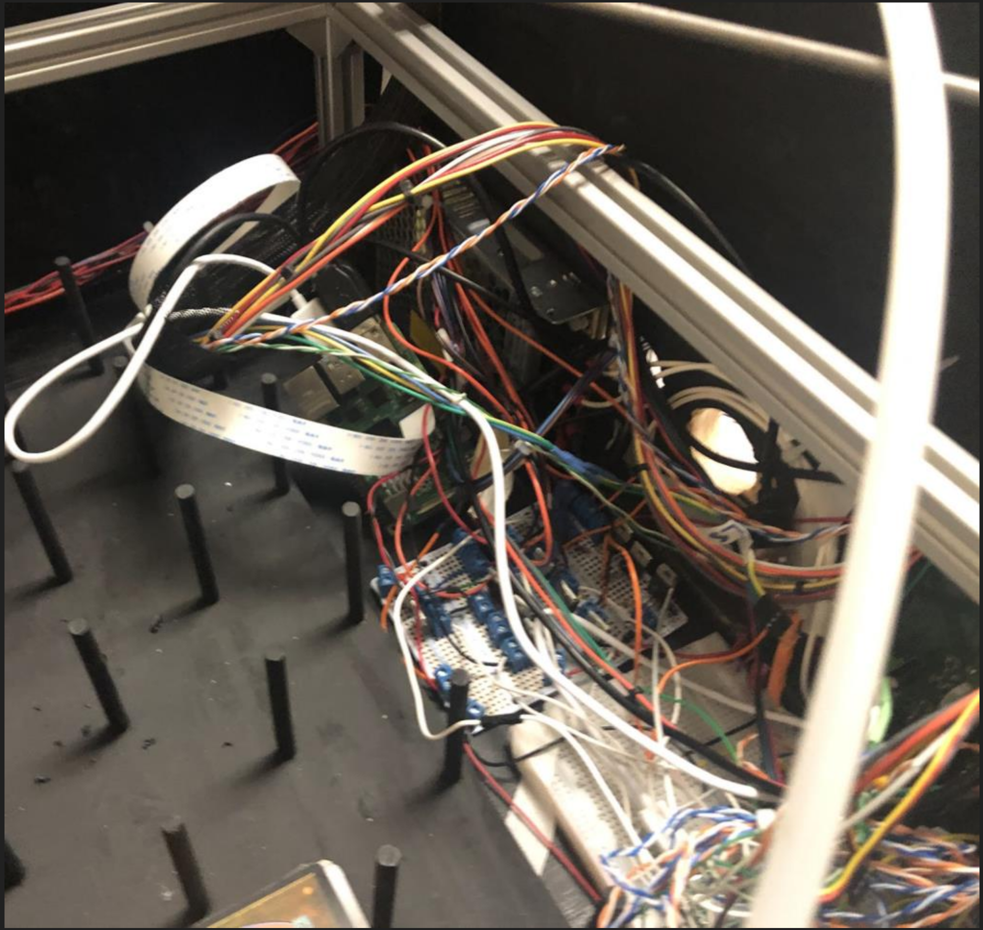
State of the Design — Tragedy





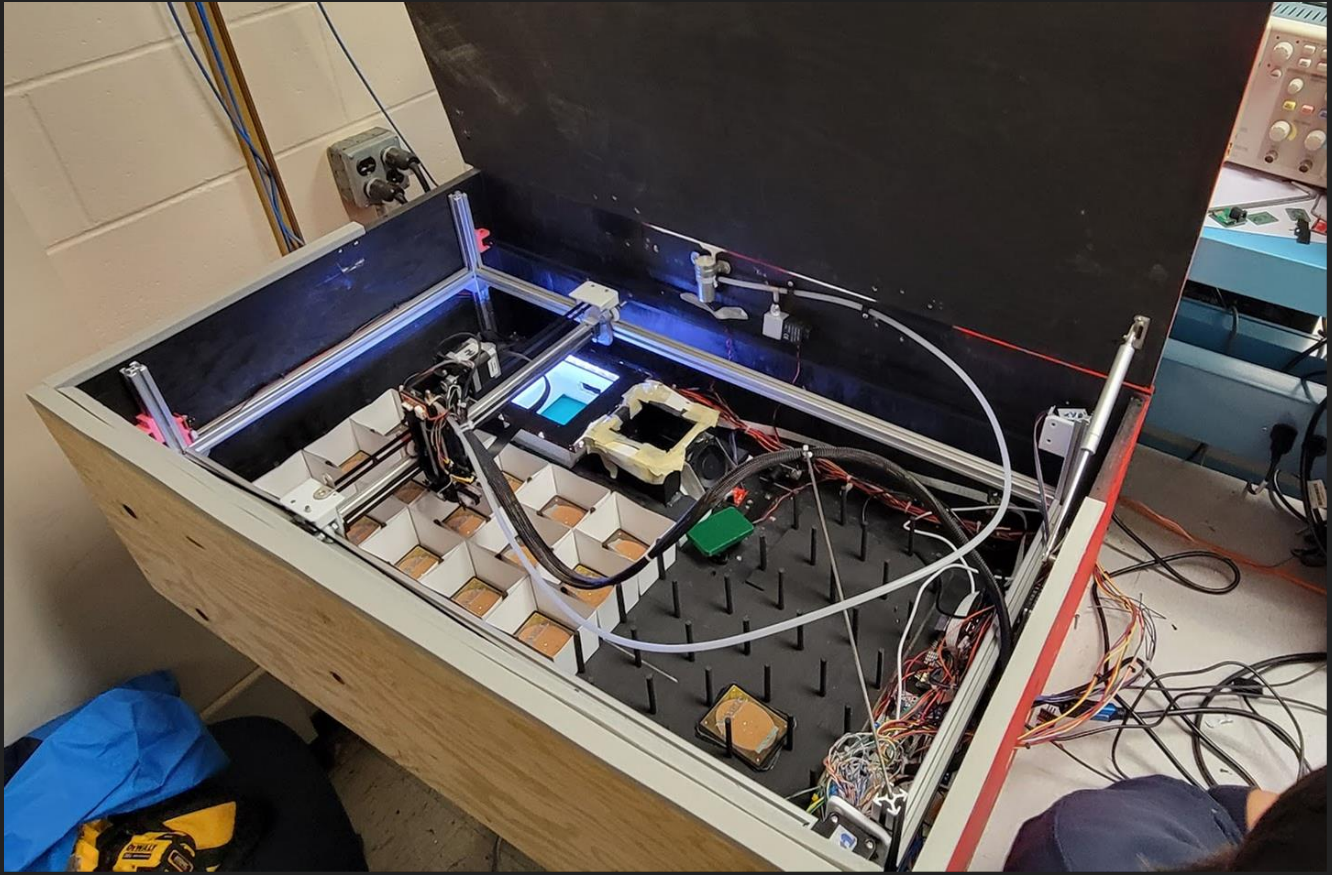
Originally, The PCB had been planned to run the RP2040 microcontroller, and all of the components relating to the power supply. The former do not work, but the latter are still working and functioning correctly.

For the purposes of our FPR demonstration, the functions that would have been handled by the part of the PCB that was destroyed are being done on an rpi pico attached to a breadboard.



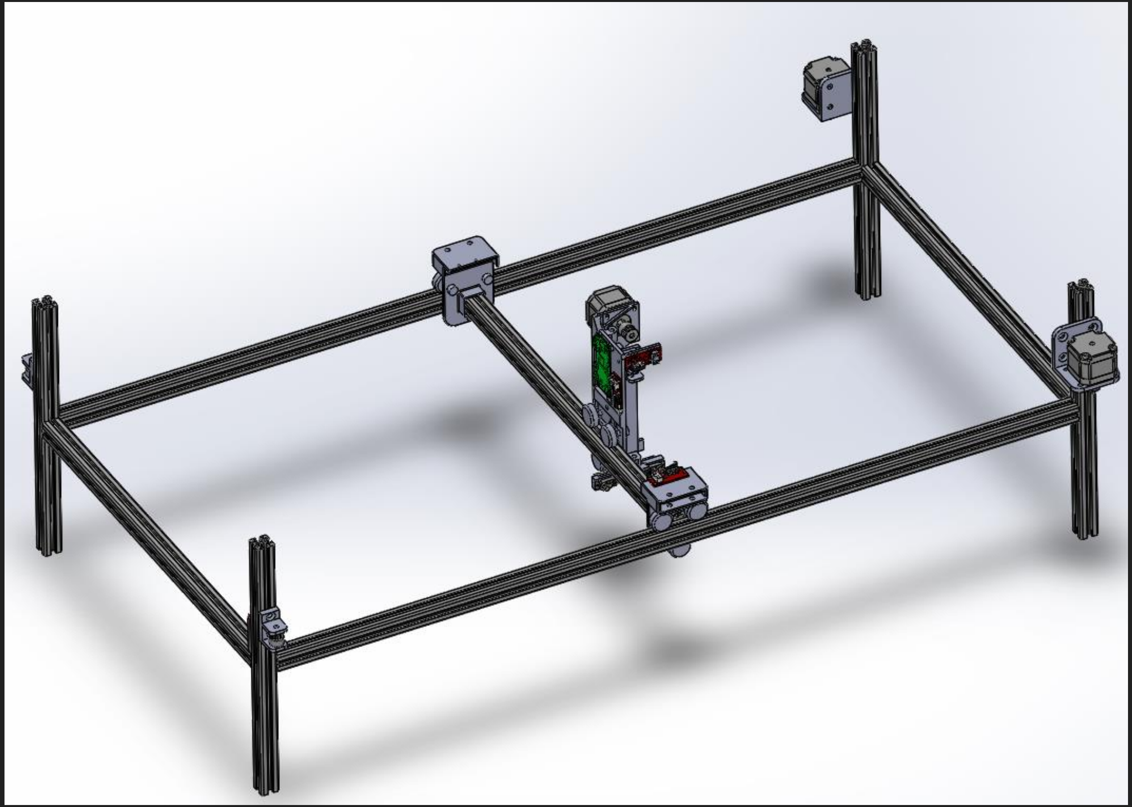
Documentation

Final System — General

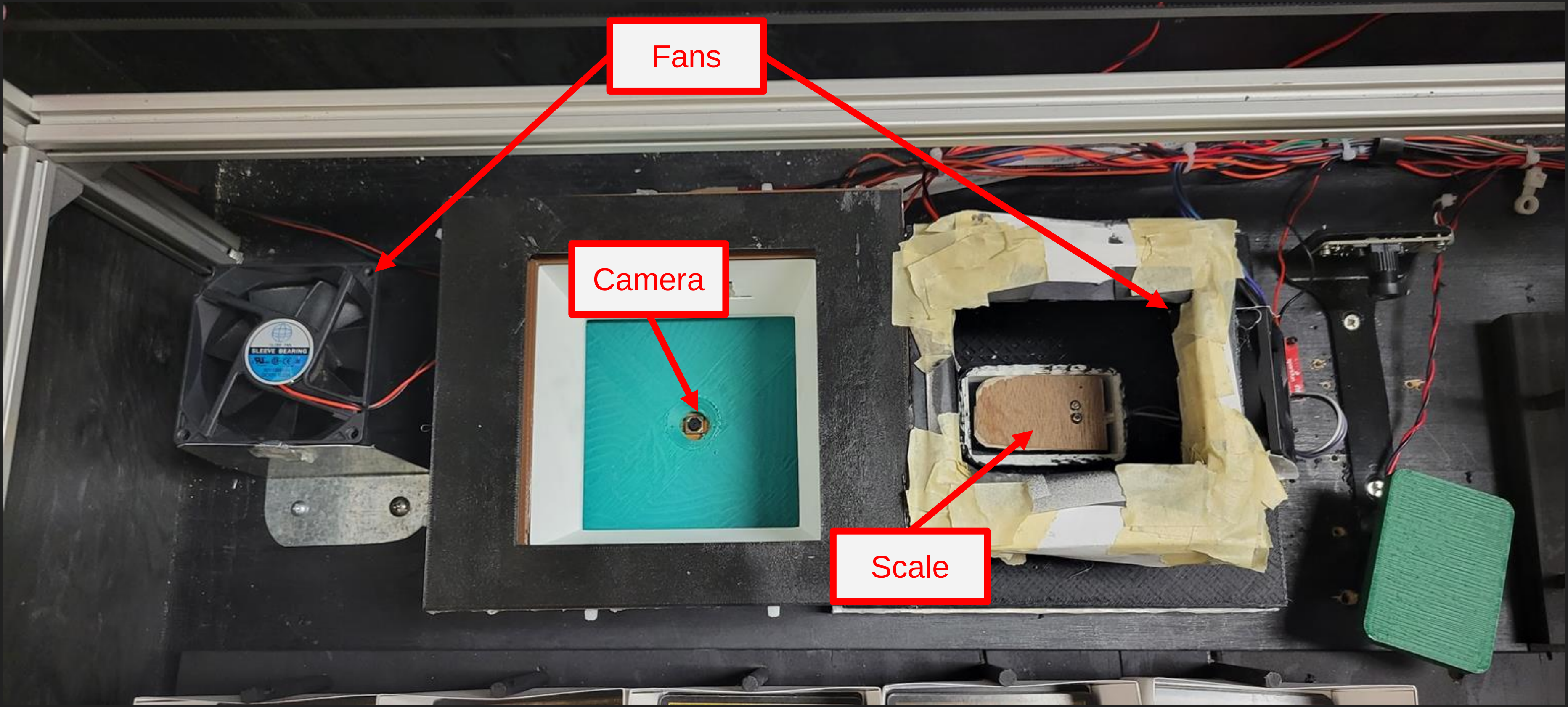


Motion System

- ±.15mm tolerance for repeatability in the XY plane
- ±.05mm tolerance for repeatability in the Z axis
- Backlash and frame deformation are primary causes
- 95cm x 40cm x 14cm usable volume

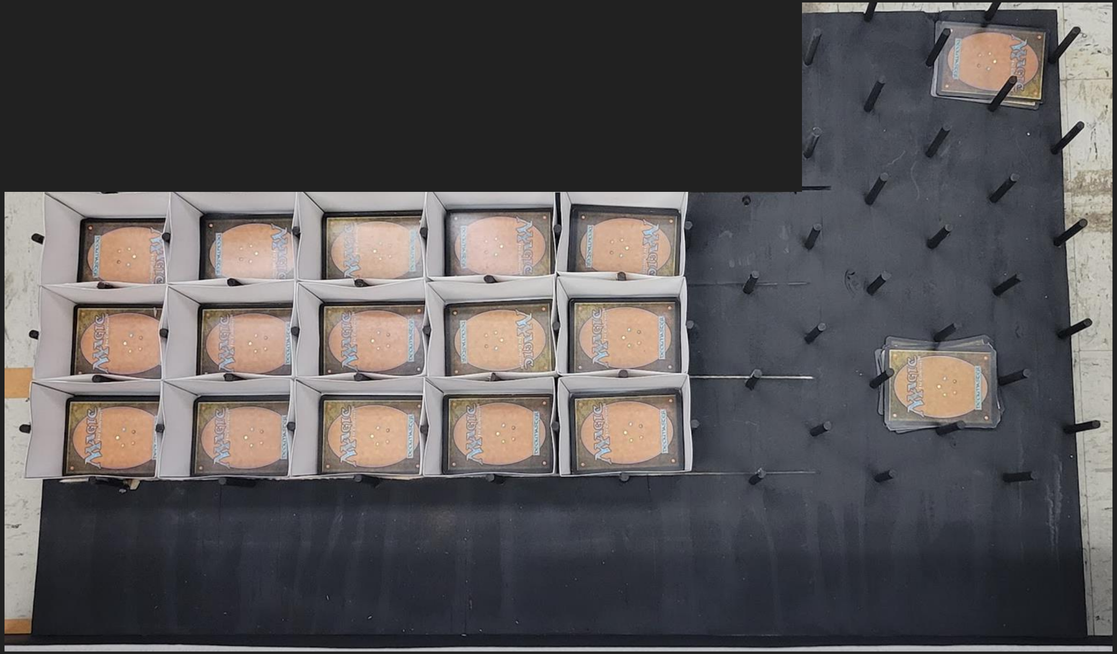


Evaluator

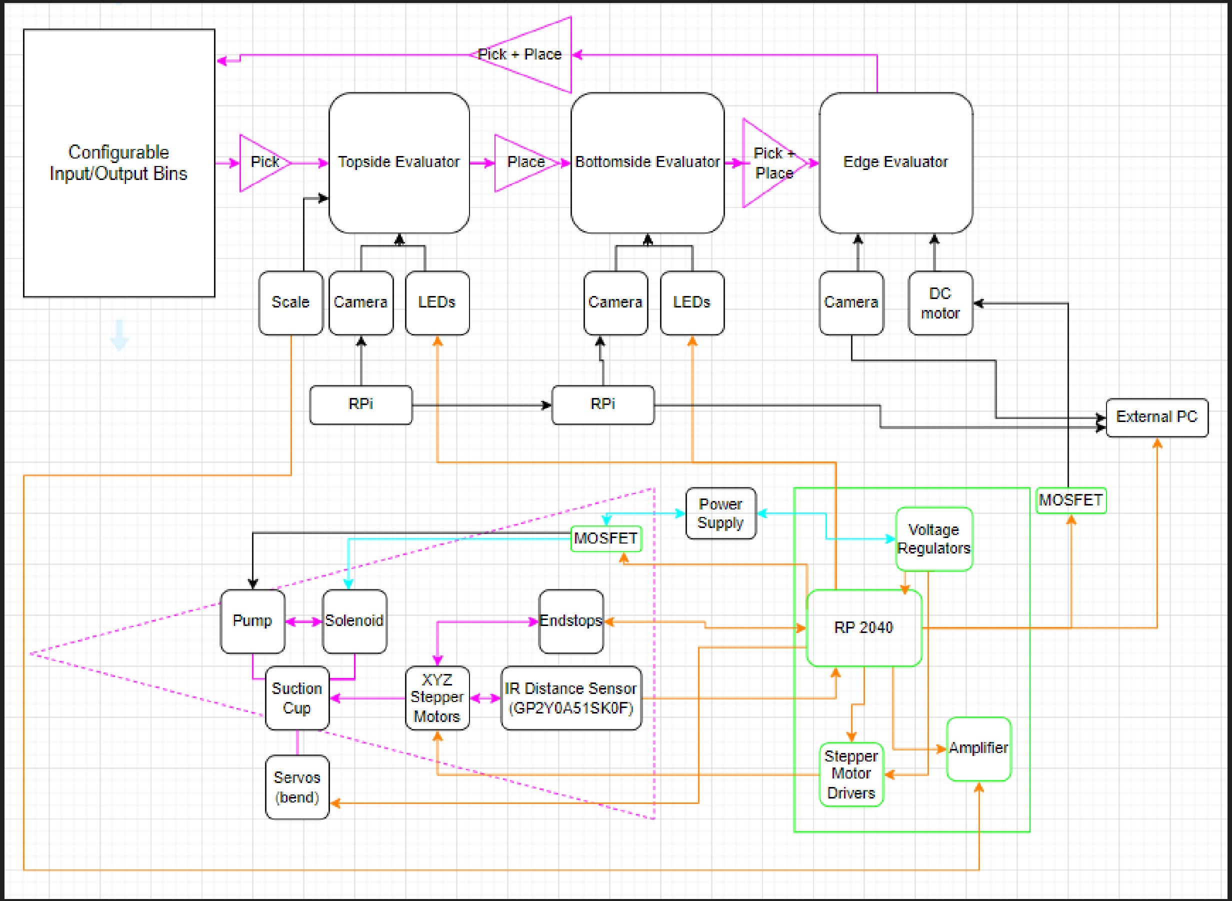


Card I/O Tray

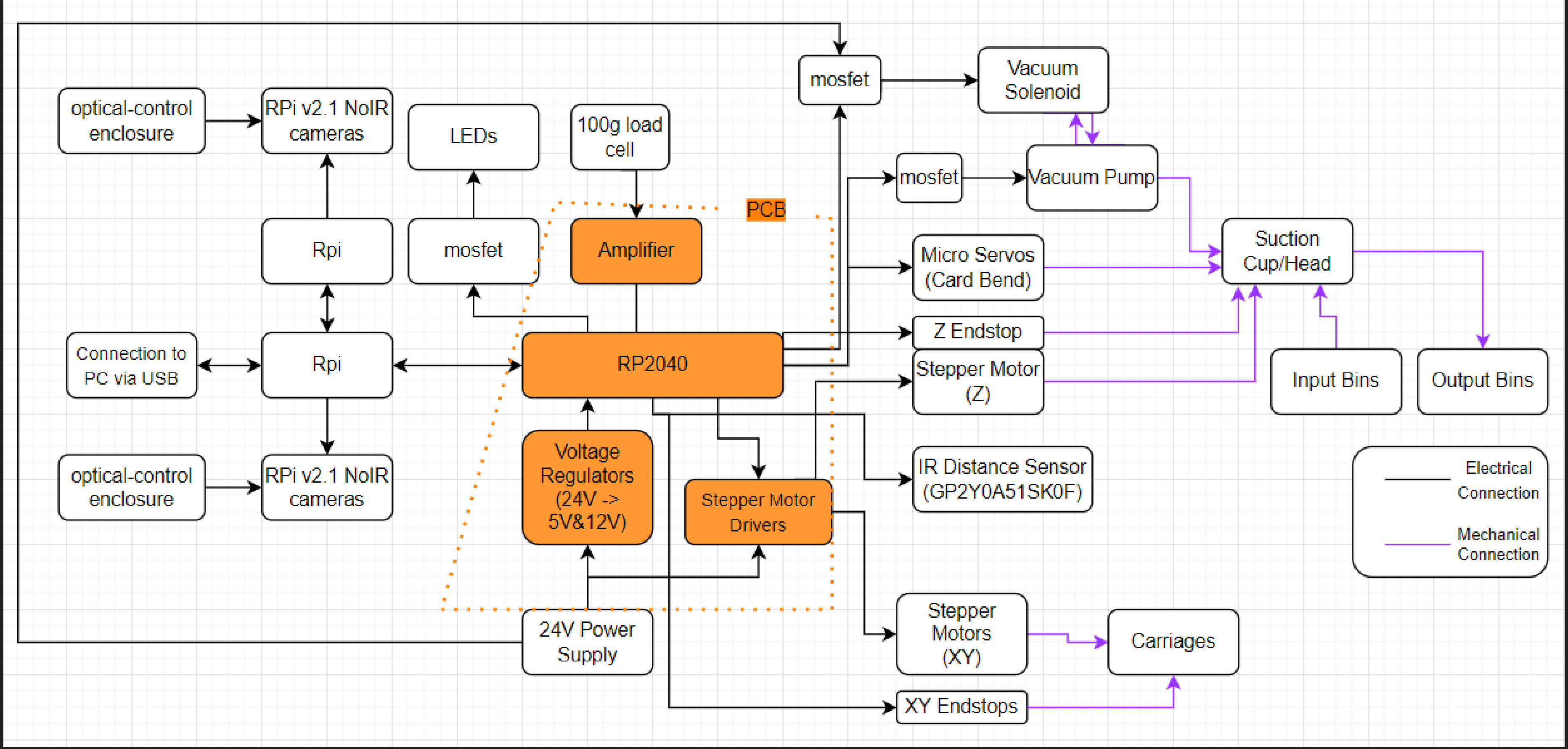
- 28 user-configurable I/O bays
- 3500 card capacity
- Paper bins prevent damage to sides of cards and line them up for processing



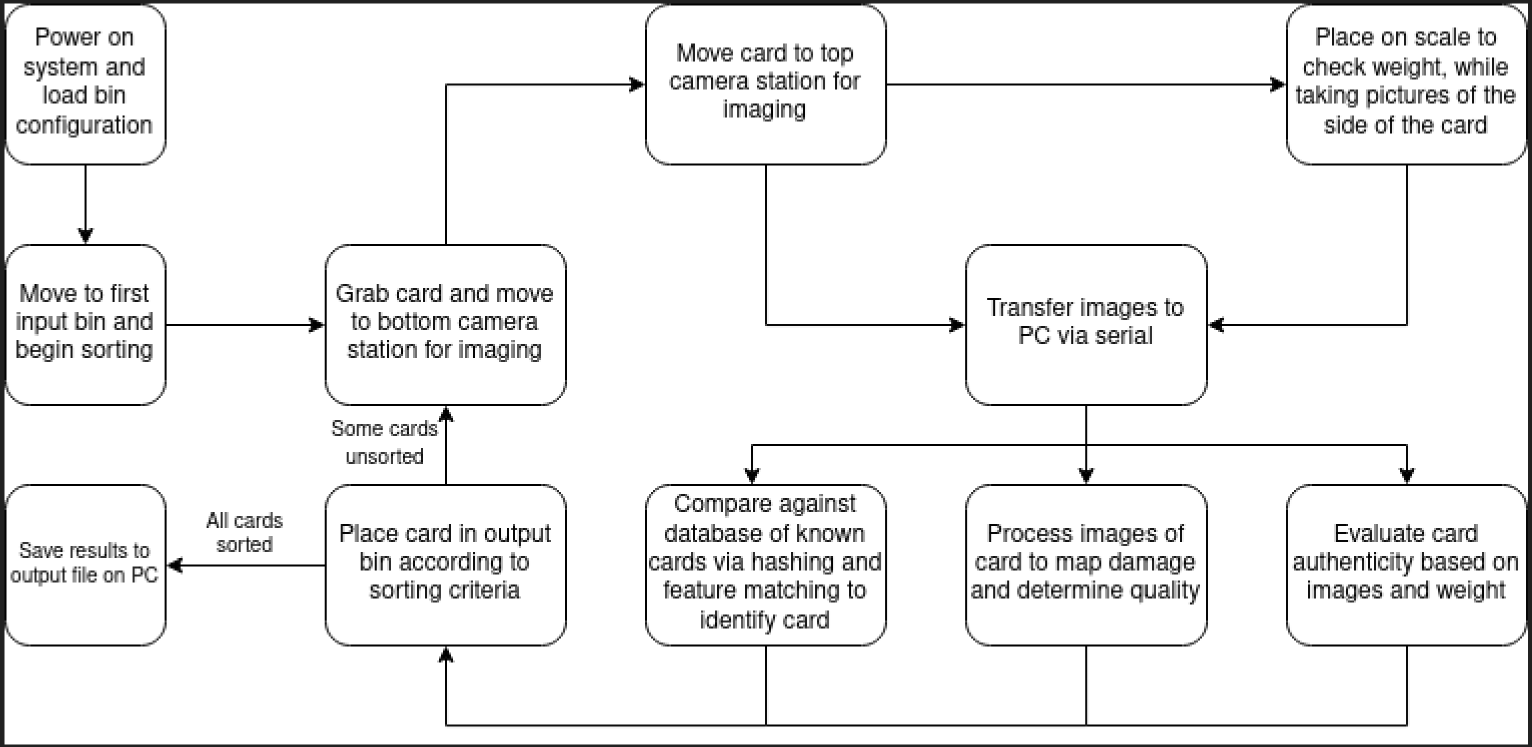
Hardware Diagram + Path of a Card



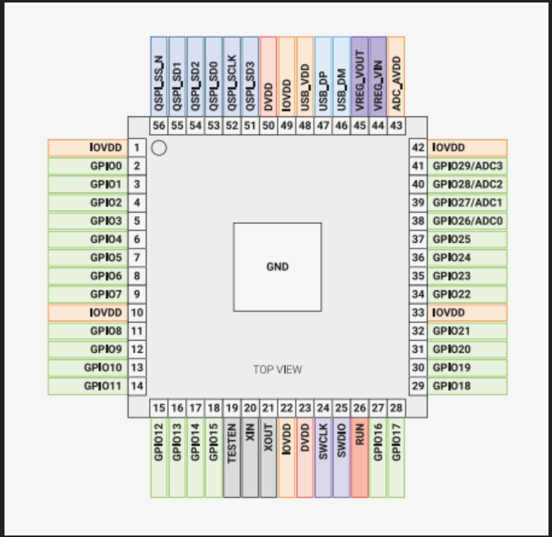
Hardware Diagram



Software Diagram

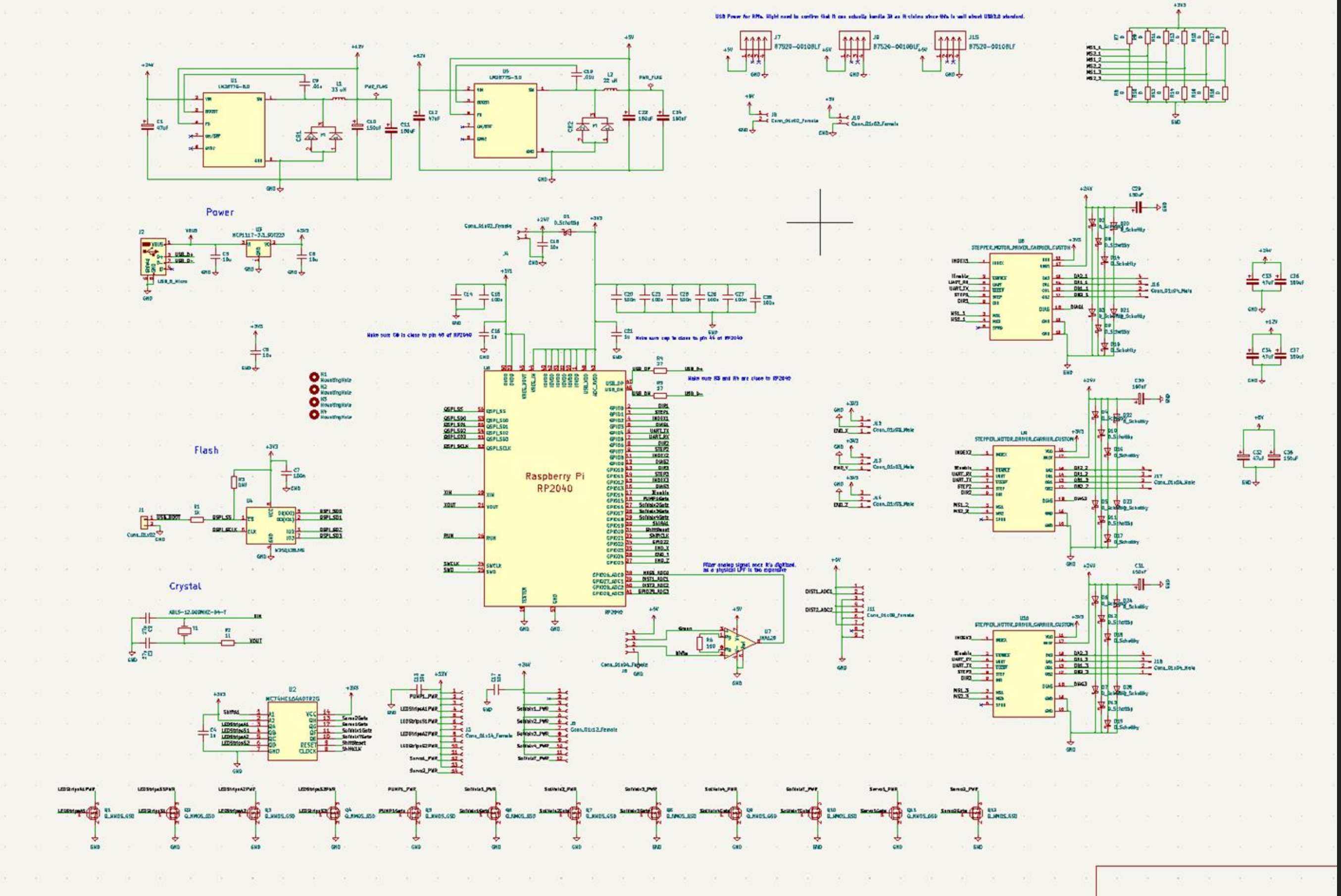


The PCB



- RP2040 microcontroller
- Voltage regulation from 24V to 5V and 12V
- Interface with sensors for system
  - IR distance sensor for z-axis of gantry
  - endstops for all 3 axes of gantry
  - load cell
  - current sensor on pump to check if card is dropped accidentally (spike)
- Motor drivers
- Current status is not fully functional so breadboard is used.

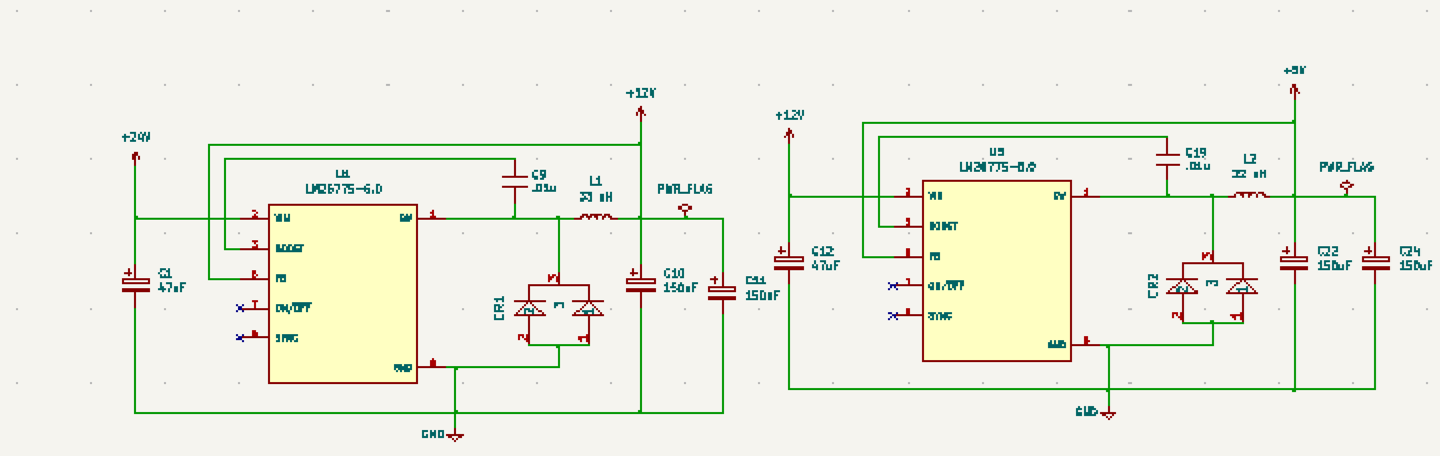
Main PCB Full Image



The main PCB had around 120 components.

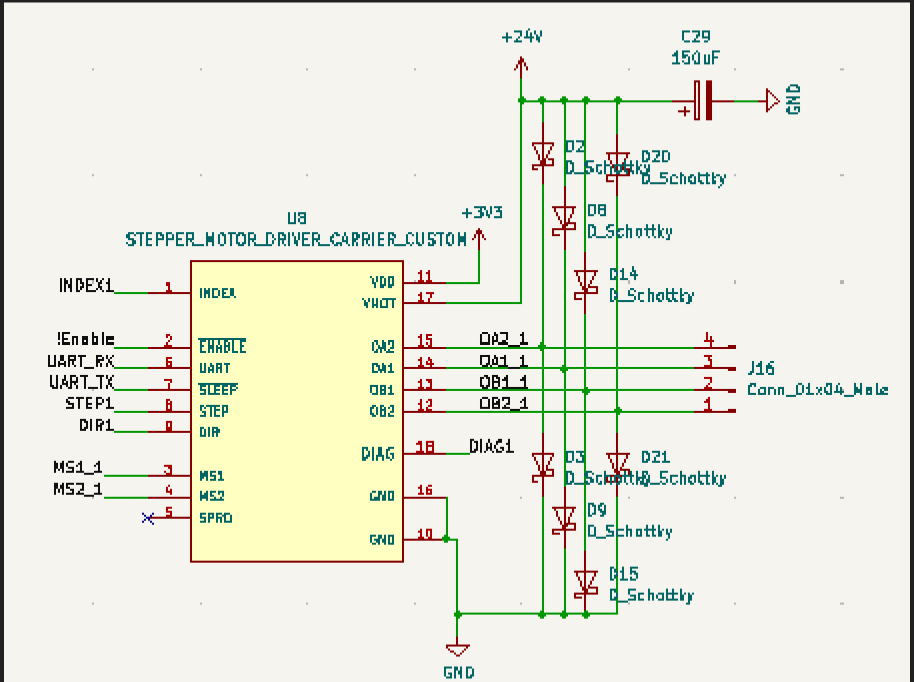
We will go over some of the subsections in the following slides.

Voltage regulation for power



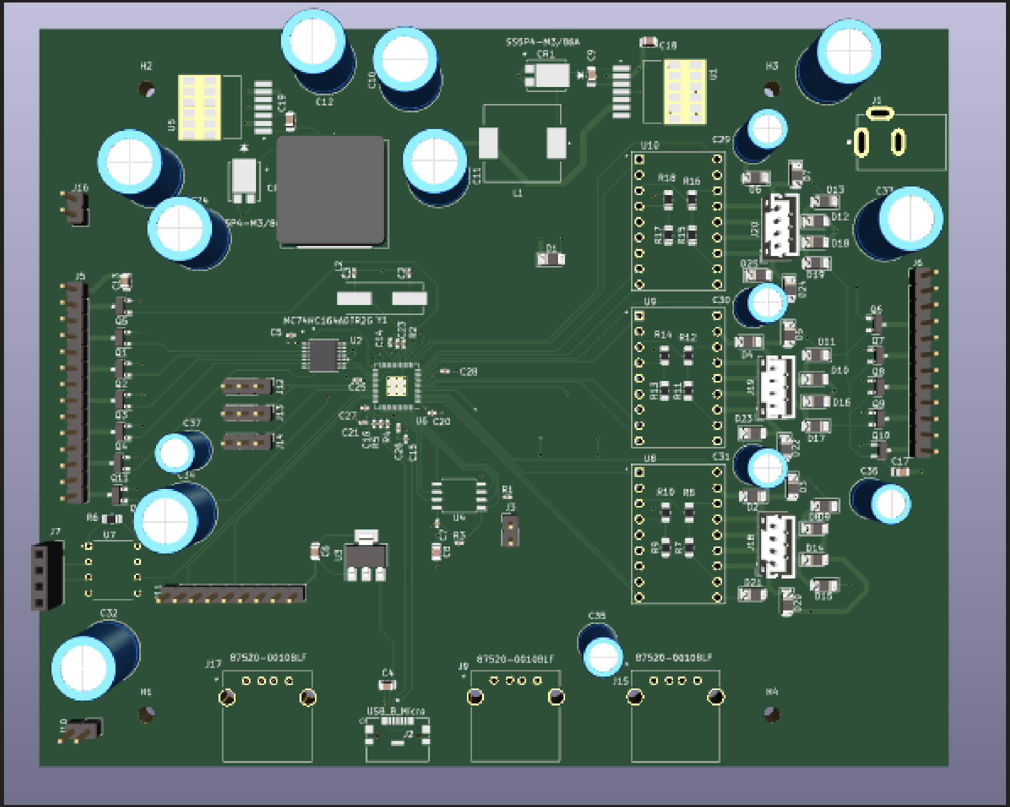
Large capacitors and inductors are required to manage the high power throughput. This part works, and means we only need a 24V power supply and a computer with USB C to power and communicate with the system

Stepper Motor Driver Carrier on Main PCB



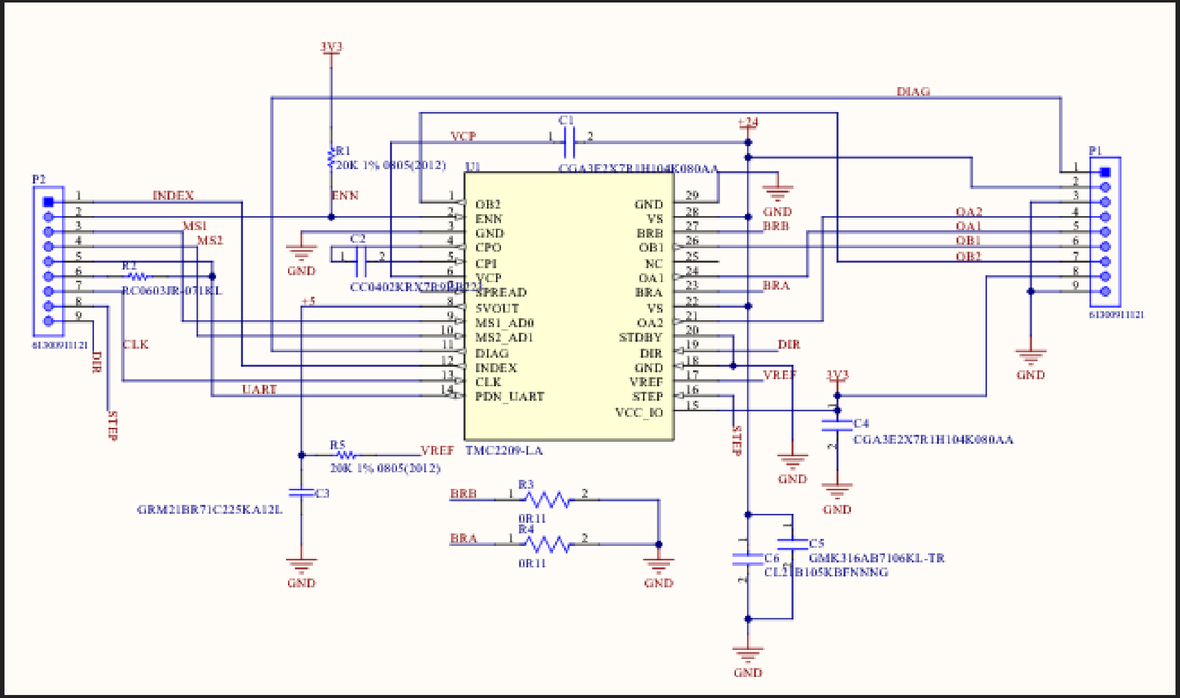
The diodes ensure that back EMF doesn't harm the motor driver or other components, however they were accidentally reversed in the original design (seen left). This and other errors were caught too late for a rev 2.

PCB Layout



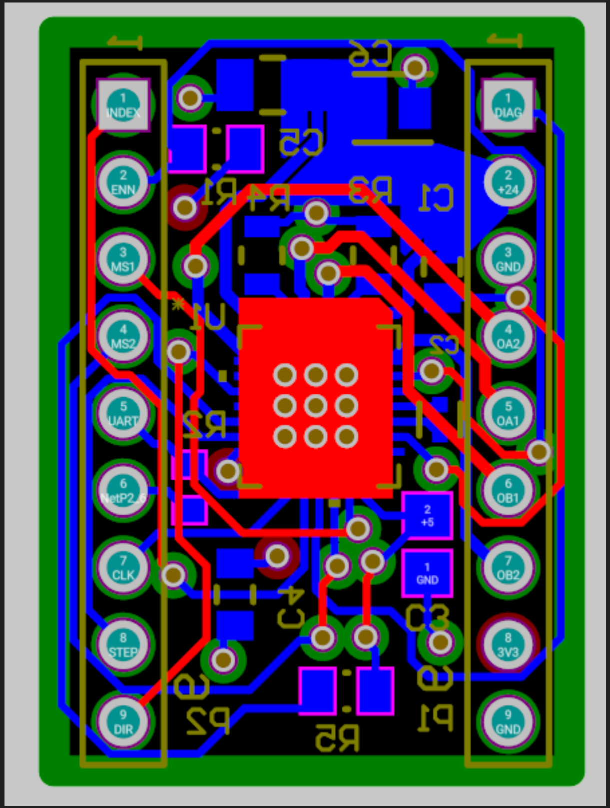
Due to breaking it, much of the design is not actually put to use, but it still provides a number of helpful pins for the rest of the project to function on.

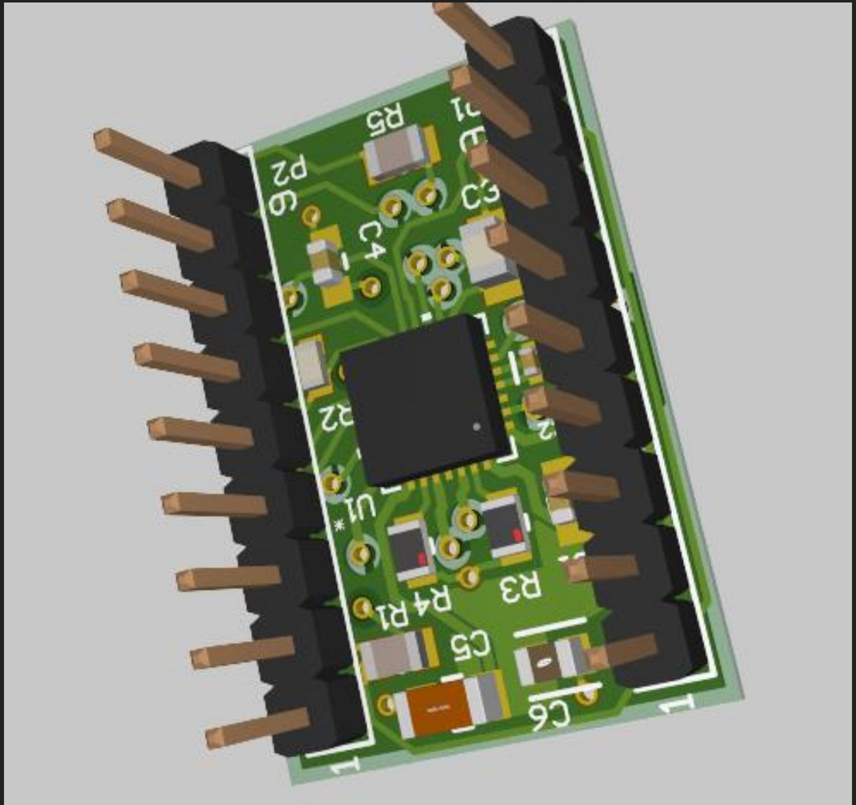
Stepper Motor Driver Carrier PCB



The new version was tested but could not control the motors.

PCB Layout for rev 2 - Unsuccessful





Thanks!
Questions?