LEBBVI

Laser Etching Books in Braille for Visually Impaired

Final Presentation

Group 9

John Childs - EE

Antonio Dufford - CPE

Vincent Pagliuca - PSE

Raul Perez - CPE

The Team

Vincent Pagliuca

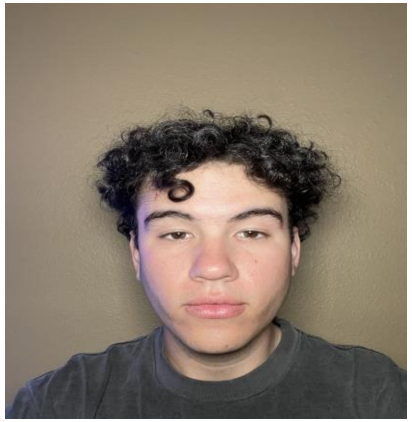
Photonics Engineer

John Childs

Electrical Engineer

Antonio Duford

Computer Engineer



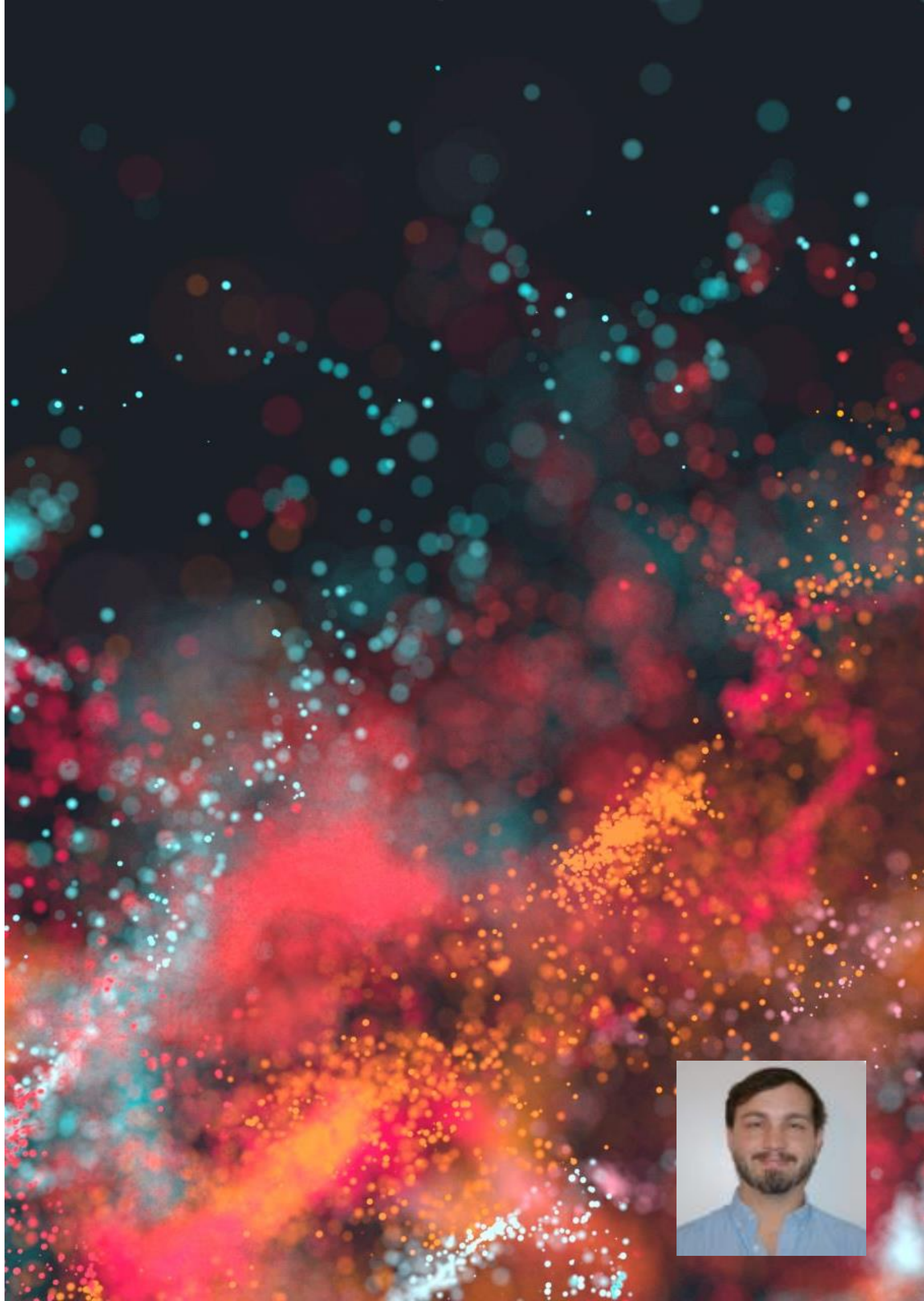
Raul Perez

Computer Engineer (VLSI)

Motivation and Background

- Our motivation behind developing a machine that can etch Braille patterns into paper using a laser stems from the urgent need to make written information more accessible to individuals with visual impairments.

- Our laser-based Braille etching system aims to bridge this gap by enabling fast, accurate, and customizable transcription of written content into tactile Braille directly onto paper, eliminating the need for bulky or expensive embossing equipment.

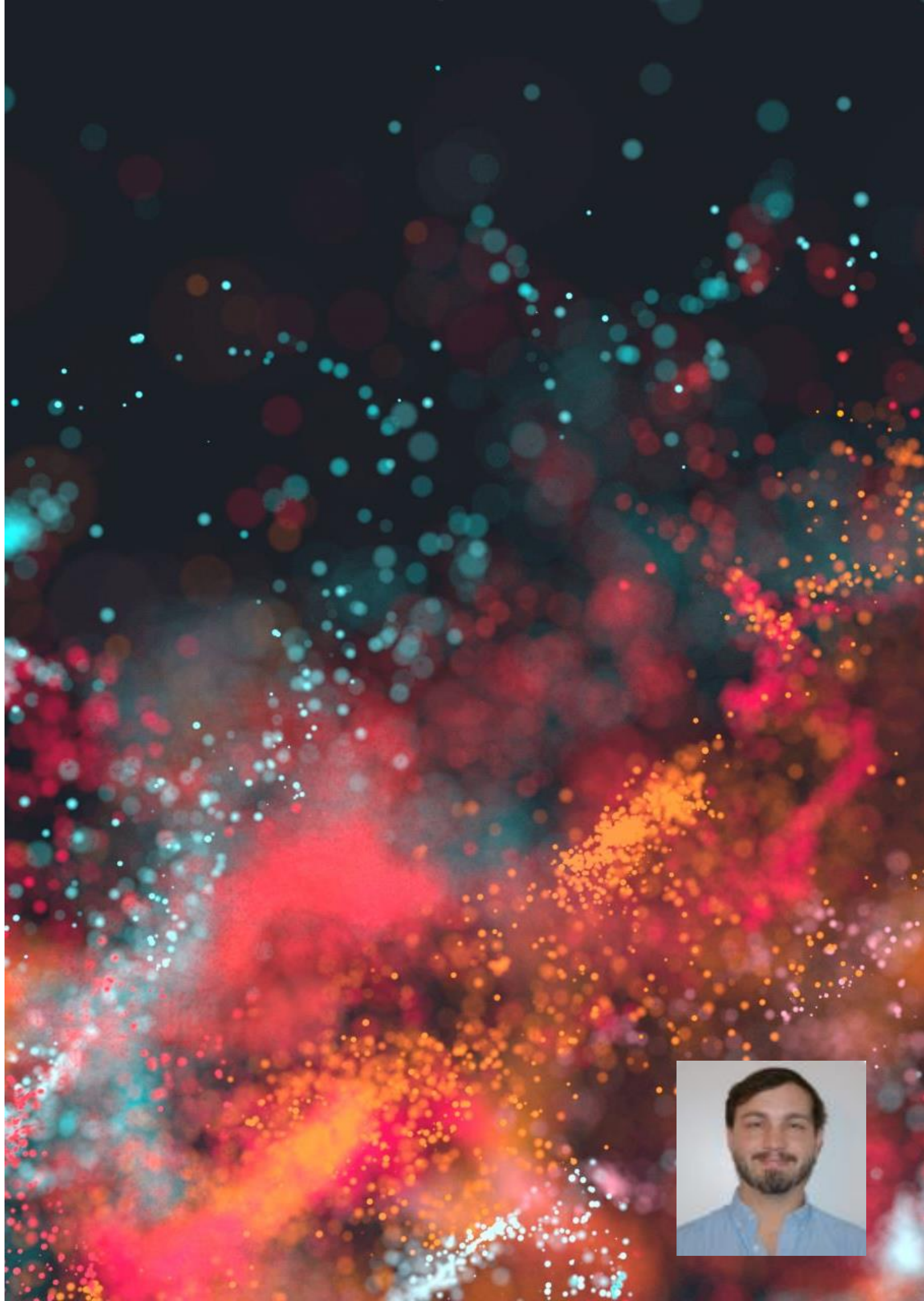


Comparable Techs

- Thermal-Emboss Braille Printers
  - Most desktop units use a heavy embossing head and motorized platen. They cost \$2000–\$5000, consume 20 W, and print at around 15 chars/sec. They require special thick paper and frequent maintenance.

Comparable Techs

- Piezo-Actuated Dot-Formers
    - Emerging designs (e.g. Tactonix BrailleCell) use piezo stacks to push pins through paper. They offer real-time refreshable panels but remain $1000+ per cell and limited to small displays.



Creality Falcon Engraver

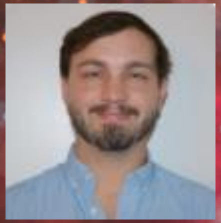
•The creality falcon Engraver is a very similar final product to what we have in mind for this project. This product is your average engraving machine, as it has a 10 watt laser and boasts a 0.1 mm precision and spot size compression to 0.12*0.06 mm from 0.32*0.14 mm.

Goals and Objectives

- Basic Goals
  - Laser Ablation: Achieve controlled, repeatable holes in standard 24 lb bond paper at 100mW laser power.
  - Print Rate: At least 10 characters per minute..
  - Etchings should be clear and easily distinguishable, meaning the beam's accuracy must be high
  - Using PWM the laser should be set at such an output that there is no risk of combustion

Goals and Objectives

- Advanced Goals
  - Multiple Language Support
  - User Interface
  - The laser source should be able to be finely tuned to control depth of etching using PWM.
  - The rail should be able to move the laser output quickly without feeling causing sudden movements in the whole system.



Goals and Objectives

- Stretch Goals
  - Wireless Printing
  - App Support
  - Dot Size variation

Goals and Objectives

- Objectives
  - Produce a stable gantry system that ensures full control of the laser and little to no kickback from the motors
  - Use different duty cycles with PWM to achieve different dot sizes, 1.3mm, 1.5mm, 1.7mm
  - Validate tactile clarity through iterative testing with Braille standards

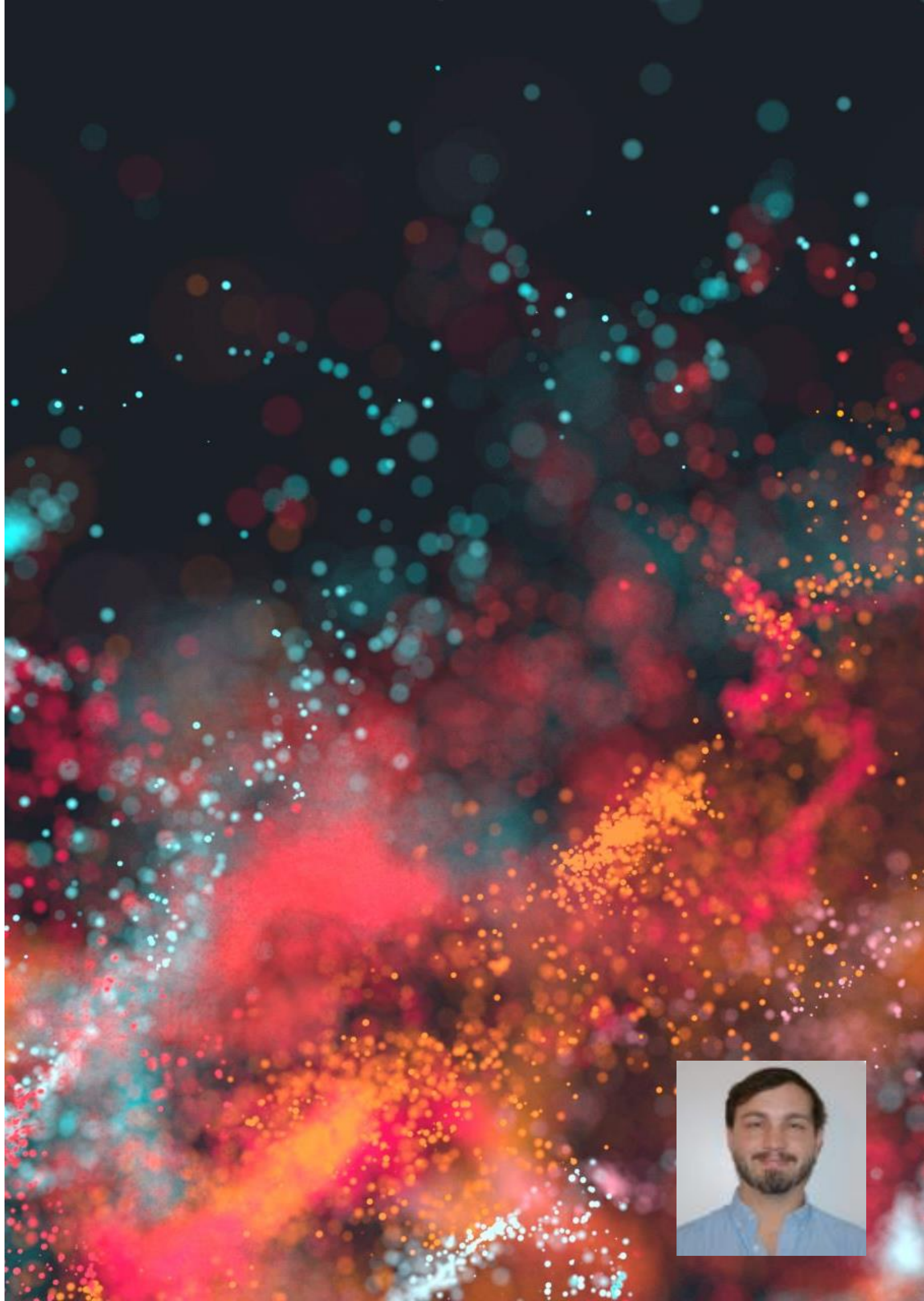
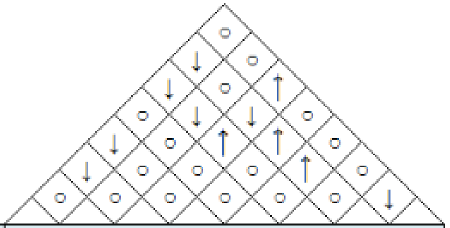


Table of Engineering Specifications

Specification	Value
Laser Ablation	< 0.25mm
Print Rate	≥ 3 char/min
Printed Types of Braille	2 (Uncontracted+Contracted)
Dot Size (Diameter of Base)	≤ 2mm (Typical 1.4-1.6mm)
Battery Life (Full Load)	≥ 2 hours
Weight of Device	≤ 6800 grams
Translation Accuracy	≥ 95%
Cost	≤ $600

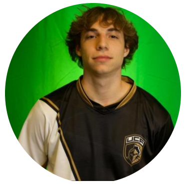
Correlations	
Strongly Positive	↑↑
Positive	↑
No Correlation	○
Negative	↓
Strong Negative	↓↓

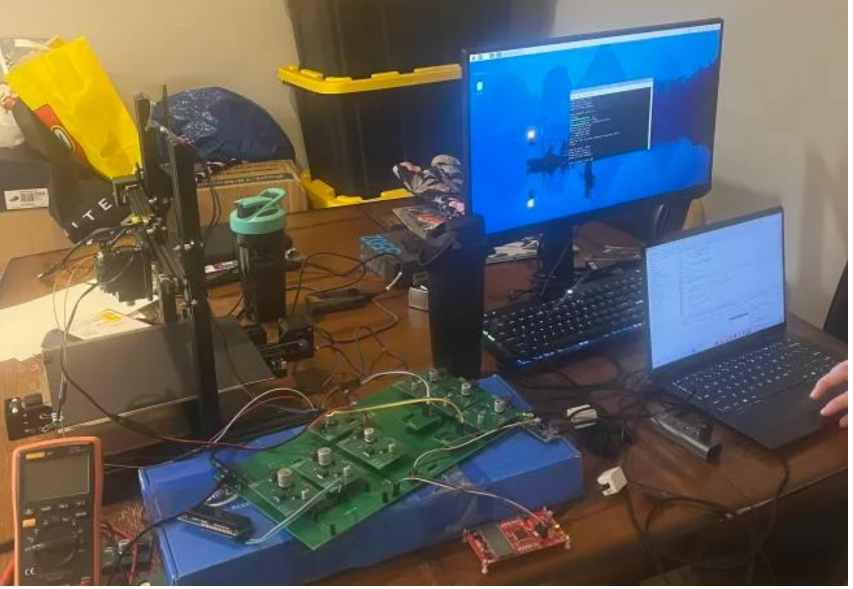
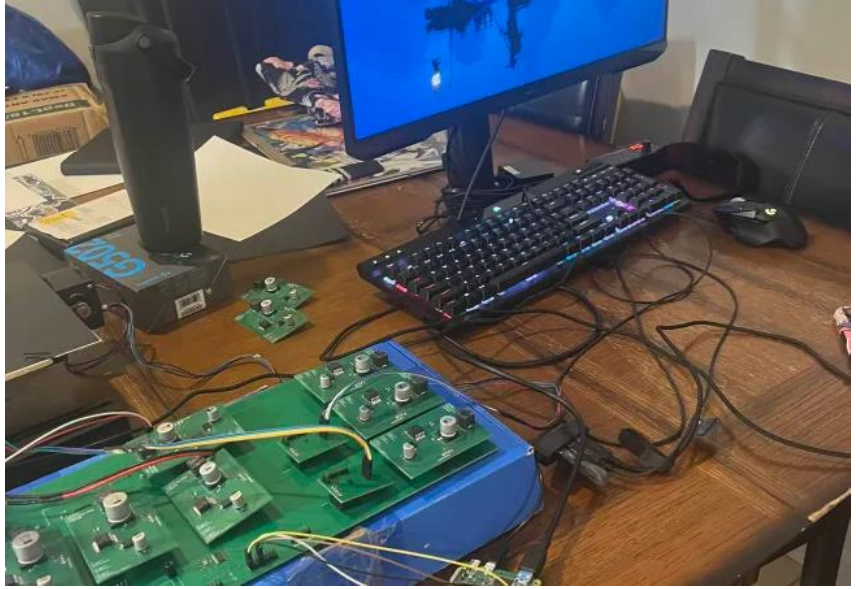
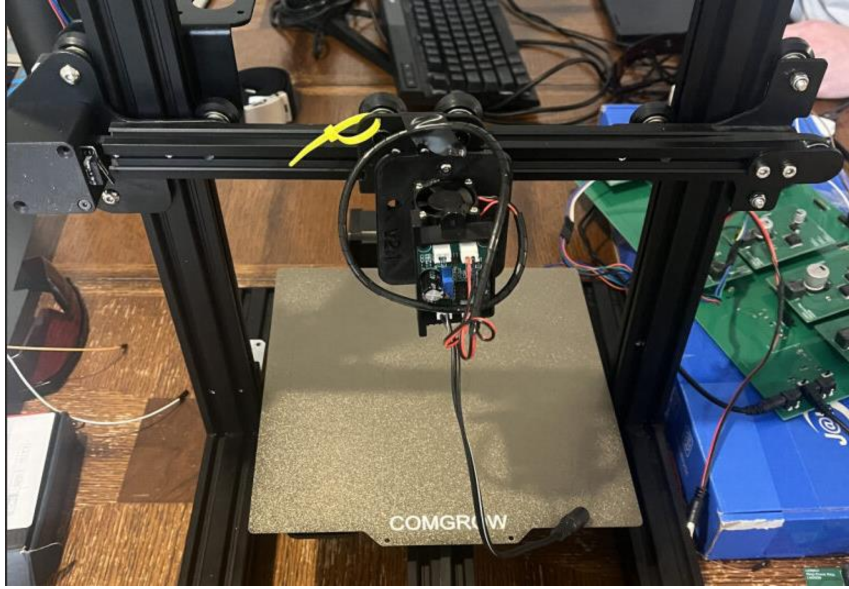
Direction of Desirability	
Maximize	+
Minimize	-



Column #		1	2	3	4	5	6	7	8
Marketing Requirements (Explicit and Implicit) / Engineering Requirements	Direction of Desirability	Print Rate	Printed Types of Braille	Dot Size	Battery Life (Full Load)	Weight of Device	Translation Accuracy	Cost	Laser Ablation
Direction of Desirability		+	+	−	+	−	+	−	−
Size of Device	−	○	○	○	↓	↑↑	○	↑	○
Affordability	+	○	○	↓	↓↓	↑	○	↑↑	↓
Laser Accuracy	+	↓	○	↑	○	○	○	↑	↑
Battery Life	+	↓	○	○	↑↑	○	○	↓	○
Ease of Use	+	○	↑	↑	↑	↑↑	↑	○	↑
Maintenance	−	↓	○	↓	↑	○	○	○	↓
Reliability	+	↓	↓	↓	↑	○	↑↑	○	↑
Safety	+	↓	○	↓	↓	○	○	○	↓↓
Target		>= 10 char/min	2 (Uncontracted + Contracted)	<= 2mm	>= 2 hours	<= 6800 grams	>= 85%	< $600	< Thickness of Paper

House of Quality



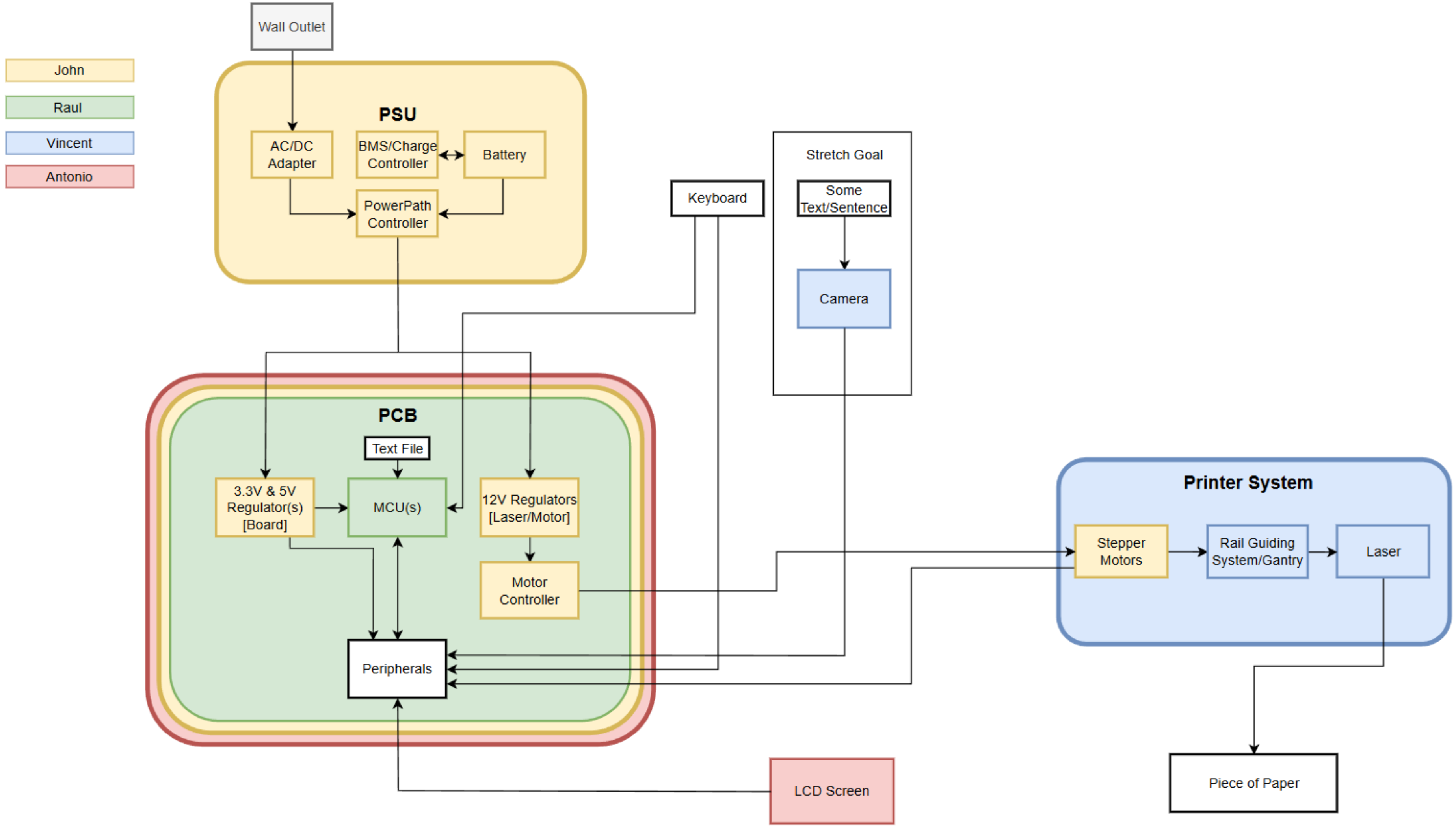




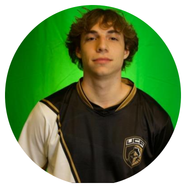
Photos of Final Product

- Final assembly of LEBBVI
  - NEMA17 motors, laser, and paper tray mounted.
  - PCB to the side to power and control the motors, laser, Raspberry Pi Zero 2, and the MSP430FR2355

Hardware Comparison and Part Selection

Hardware Block Diagram





Laser Selection (CW)

Specs	Thulium-Doped Fiber Lasers	CO2 Gas Laser	Laser Diode
Output Wavelength (nm)	1900	10600	405
Output Power	>30mW	20W	100mW
Price	$6000	$4500	$50

- This diode was a perfect balance between high power and budget cost. This diode operates at 405nm and outputs about 100mW, and also comes with an adjustable focus which allows us to change the dot size and therefore intensity.

Comparable Parts MSP

Feature	MSP430FR6989	MSP430FR2355 ☆	MSP430G2553
Flash/FRAM	128 KB FRAM	32 KB FRAM	16 KB Flash
RAM	2 KB	4 KB	512 B
Clock Speed	16 MHz	16 MHz	16 MHz
Peripherals	LCD driver, ADC12, DMA, multiple timers	12-bit ADC, timers, low-power modes	Basic ADC10, limited timers
Power Consumption	Ultra-low power modes	Ultra-low power modes	Low, but less capable
Package	100-pin	38-pin QFN/TSSOP	20-pin DIP
Cost	$18	$6	$5

- MSP430FR2355 was chosen due to it being easier to solder onto the main board.

Comparable Parts
Raspberry Pi

Feature	Raspberry Pi Zero WH ☆	Raspberry Pi 3B+	Raspberry Pi 4B
CPU	1 GHz single-core ARM	1.4 GHz quad-core ARM	1.5 GHz quad-core ARM
RAM	512 MB	1 GB	2–8 GB
Wireless	Wi-Fi + Bluetooth	Wi-Fi + Bluetooth	Wi-Fi + Bluetooth
GPIO Pins	40	40	40
Size	(65 x 30 mm)	(85 x 56 mm)	(85 x 56 mm)
Power	5V @ ~150 mA	5V @ ~700 mA	5V @ ~1.2 A
Cost	$25	$35	$45–75

- The Raspberry Pi Zero WH was chosen as it had all the features needed for stretch goals while still being both a small form factor and decently inexpensive.

PowerPath Controller Technology Selection

PowerPath Type	Ideal Diode-Based	Power Mux	Dedicated ICs
PowerPath Eff.	High	High	Moderate
Control Method	Voltage Difference	Comparator	Internal Logic
Voltage [V]	< +/- 100	0.7 – 22	3 – 70
Batt. Charging	No	No	Yes
Regulation Control	No	No	IC Dependent

- An Ideal Diode-Based PowerPath technology was chosen because they are simple to integrate into a circuit.

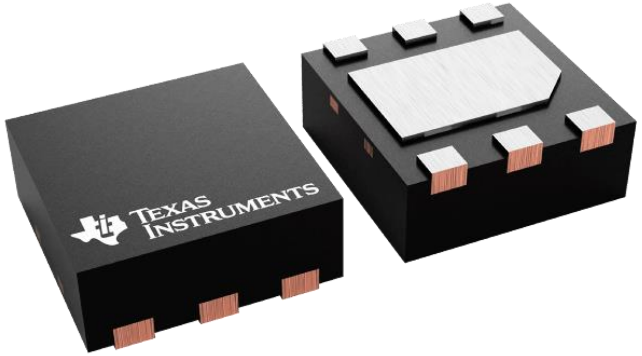
Ideal Diode PowerPath Controller Part Selection

	LTC4412	LTC4413	LTC4416
Input Voltage from AC/DC Adapter [V]	3 – 28	0 – 5.5	3.6 – 36
Input Voltage from Battery [V]	2.5 – 28	2.5 – 5.5	3.6 – 36
Channels	1	2	2
Max Output Current [A]	Limited by FET	2.6 per channel	Limited by FET
Max Quiescent Current [uA] (High Supply)	26	40	130
Operating Temperature [C]	-55 – 150	-40 – 85	-40 – 125

- The LTC4412 was chosen due to its wide input voltage range and the simple "plug and play" ability when paired with an AC/DC wall adapter and battery pack input source.

Undervoltage Protection Circuit

Voltage Supervisory Chip	MCP100	MAX809	TPS3808G01
Nominal Supply Voltage [V]	1 - 5.5	-0.3 - 6	-0.3 - 7
REST Pin Current [A]	3m	20m	-5m - 5m
RESET Pin	Push-Pull	Push-Pull	Open-Drain
Threshold Voltage (V)	1	1	Adjustable
Temperature Range [C]	-40 - 85	-40 - 105	-40 - 125



- The reason we chose the TPS3808G01 is because it provides the needed flexibility through the adjustable threshold voltage via the equation: $V_{IT}' = (1 + \frac{R_1}{R_2})\ 0.405$
- The flexibility is needed because the battery pack will not always sit at a constant voltage and can fall below the needed supply voltage needed to power our system.

Motor Technology Selection

Motor	Brushless	Servo	Stepper
Step Angle	Continuous	Continuous	0.9° - 1.8°
Speed [RPM]	< 10,000+	< 5000+	< 1000
Efficiency [%]	85 – 95	80 – 90	20 – 40
Torque [Nm]	0.1 – 5+	0.5 – 10+	0.2 – 3
Ease of Use	Medium	Complex	Simple

- Stepper motors were our motor of choice because the purpose of our project is to move in very minuscule steps (one step at a time) and not continuously.

Motor Selection

Motor	25BY4801	20M Series	NEMA 17 ☆
Steps Angle [Deg.]	7.5	18	1.8
Holding Torque [N.m]	0.01	0.07	0.59
Rated Current [A]	500m/Phase	100/Phase	2/Phase
Phases	4	2	2
Cost	Low	High	Low

- The NEMA17 stepper motor was chosen due to how common it is used (therefore a lot of online support), the cost, high step precision when paired with the DRV8825 due to micro stepping, and the precise step angle.

Motor Controller Part Selection

Motor Controller	TMC2209	TMC2225	DRV8825 ☆
Voltage Range [V]	4.75 – 29	4.75 – 36	8.2 – 45
Maximum Drive Current [A]	2.8	2	2.5
Microstepping	Up to 1/256	Up to 1/256	Up to 1/32
Interface Type	STEP, DIR, UART	STEP, DIR, UART	STEP, DIR
Operating Temperature [C]	-50 – 150	-50 – 150	-40 – 150

- The DRV8825 motor controller was chosen due to its widespread commonality with the NEMA17 motor as well as how simple it is to implement with our MCUs.

Power Distribution Table

COMPONENT	SUPPLY VOLTAGE [V]	ESTIMATED CURRENT DRAW [MA]	NOTES
MSP430FR2355 MCU	3.3	< 50	
Raspberry Pi Zero 2	5	< 1500	HDMI, USB Hub, Computing
DRV8825s	12	< 1500	Estimated peak current from motors, not continuous
TPS3808	3.3	< 5	
LTC4412	13.5 - 17	< 0.05	
I2C LCD1602	5	< 1.2	
Laser	12	< 42	500mW Laser
NEMA17 Steppers	12	< 750/phase	Limited to Vref (0.75V), 2 Phase Motors

- Total peak current to ever be drawn from the PSU would be 3-4.5A. We did not encounter any case where we drew more than a continuous 3A during testing.
- Current draw from each respective LM2596-ADJ regulator is directly tied to the component that it is powering

Switching Regulator Selection

Switching Regulator	Buck	Boost	Buck-Boost
Input Voltage [V]	1 – 100	0.5 – 40	2 – 40
Polarity	Positive	Positive	Negative
Output Voltage [V]	0.8 – Vin	Vin – 100+	-(0.8 – 100)
Efficiency [%]	85 – 95	75 – 90	70 – 85
Ripple Voltage [mV]	< 50	50 – 100	50 - 200

- Buck converter regulators are chosen as we will not be boosting any of our voltages.

Buck Regulator Part Selection

Buck Converter	LM2596	TPS5430	LT8609
Input Voltage [V]	4.5 – 40	5.5 – 36	3 – 42
Output Load Current [A]	3	3 (4 peak)	3
Efficiency [%]	Up to 90	Up to 95	Up to 95+
Switching Frequency [Hz]	150k	500k	200k – 2.2M
Operating Junction Temperature [C]	-65 – 150	-40 – 125	-40 – 125

- The LM2596 was chosen due to its high input voltage range, large output load current, high efficiency, and the ability to adjust the output voltage based on 4 components using the same design.

AC/DC Power Supplys

AC/DC PSU	SHNITPWR Adjustable ☆	Benchtop PSU	ATX Power Supply
Output Voltage [V]	3 - 24	0 – 30	12
Current Limit [A]	3	10	~35~
Maximum Power [W]	75	300	~500+
Cost	Low	High	Very high

- We choose the adjustable power supply brick because we simply do not need the capabilities of other power supplies, such as a higher current limit/maximum power.

- Adjustable AC/DC Power supply allows us to set our wall adapter input slightly larger than our battery output voltage as well as it being cheaper than most options.

Battery Type Selection

Battery Type	Alkaline	Li-Po	Li-Ion
Nominal Voltage [V]	1.5	3.6 – 3.7	3.6 – 3.7
Operating Range [V]	0.9 – 1.6	2.75 – 4.2	2.5 – 4.2
Capacity [mAh]	500 – 3000	500 – 3000	800 – 5000
Cycle Life [Cycles]	N/A	200 – 500	300 – 500
Rechargeable	No	Yes	Yes

- Li-Ion batteries are chosen because of their high capacity, rechargeability, and safety.

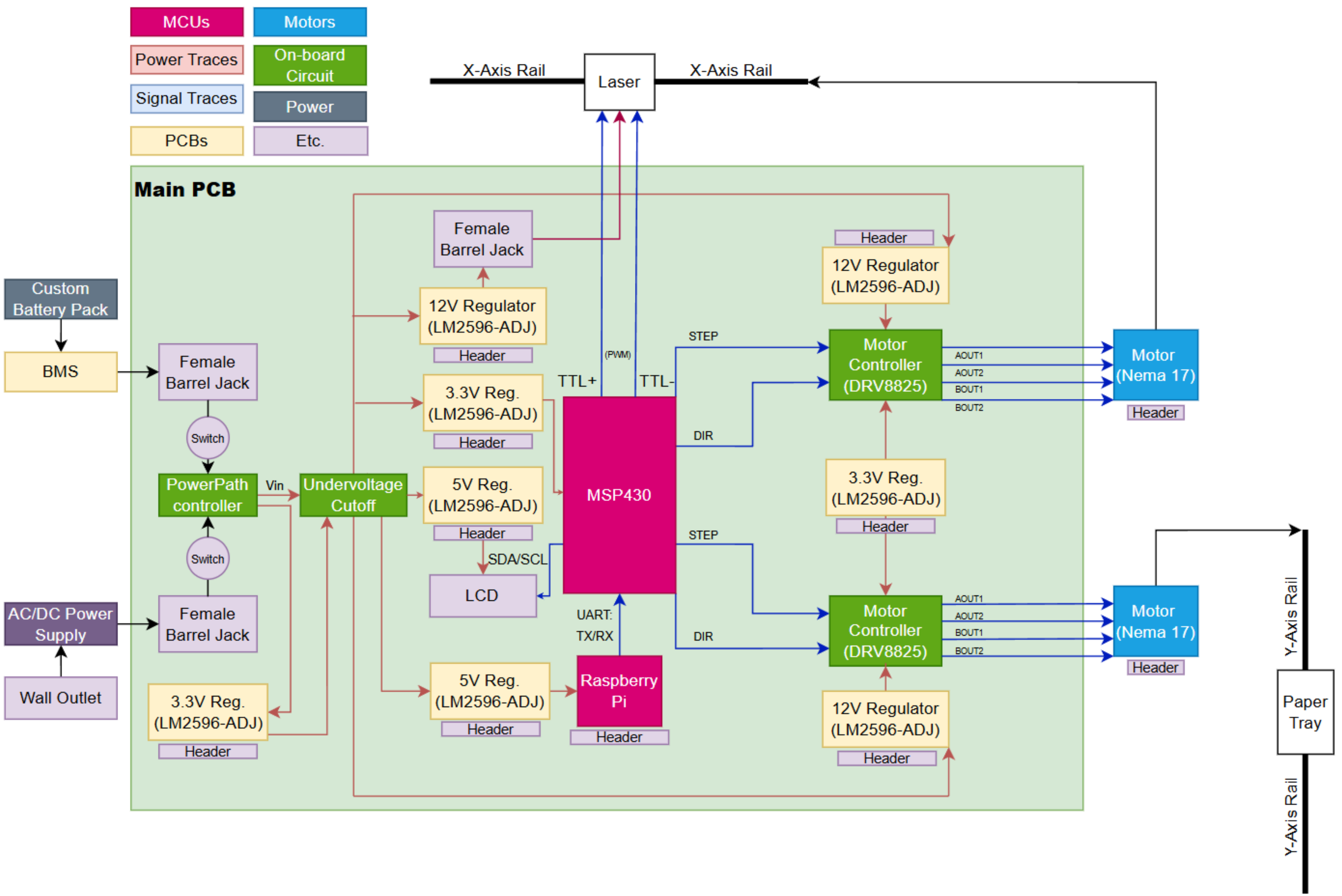
Battery Part Selection

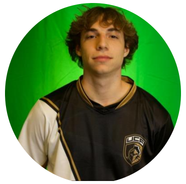
Battery (Hybrid Cell)	18650	21700
Nominal Voltage [V]	3.6 – 3.7	3.6 – 3.7
Operating Range [V]	2.5 – 4.2	2.5 – 4.2
Capacity [mAh]	2800 – 3100	4000 – 4500
CDR [A]	15 – 20	15 – 25
Cost/Cell [$]	6 – 10	8 – 14

- 18650 Batteries chosen as they are more affordable
- Will be in a 4S2P configuration.

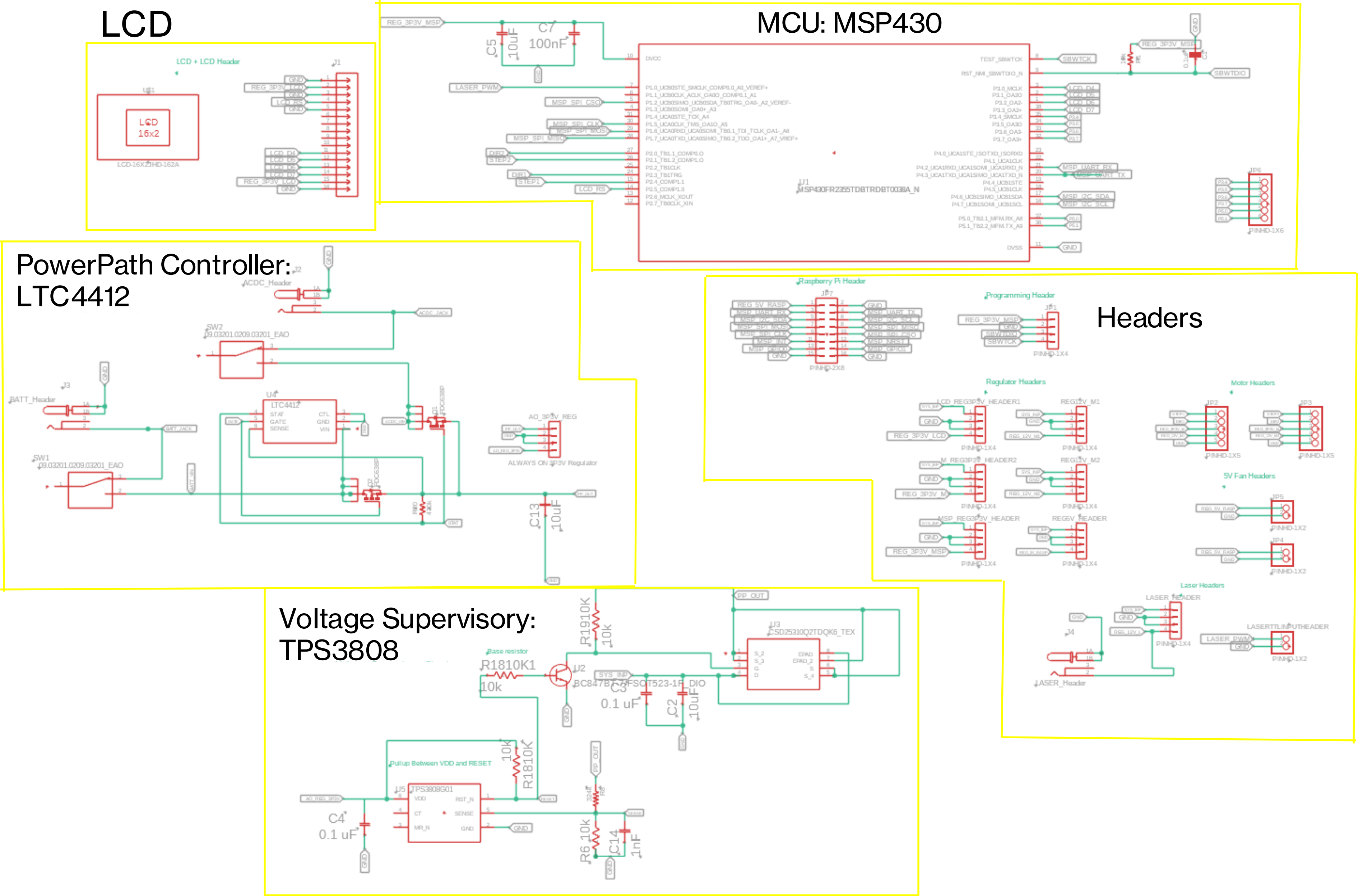
Hardware Design

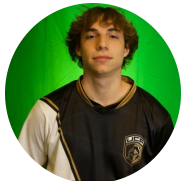
High-Level Design Block Diagram





Overall Schematic

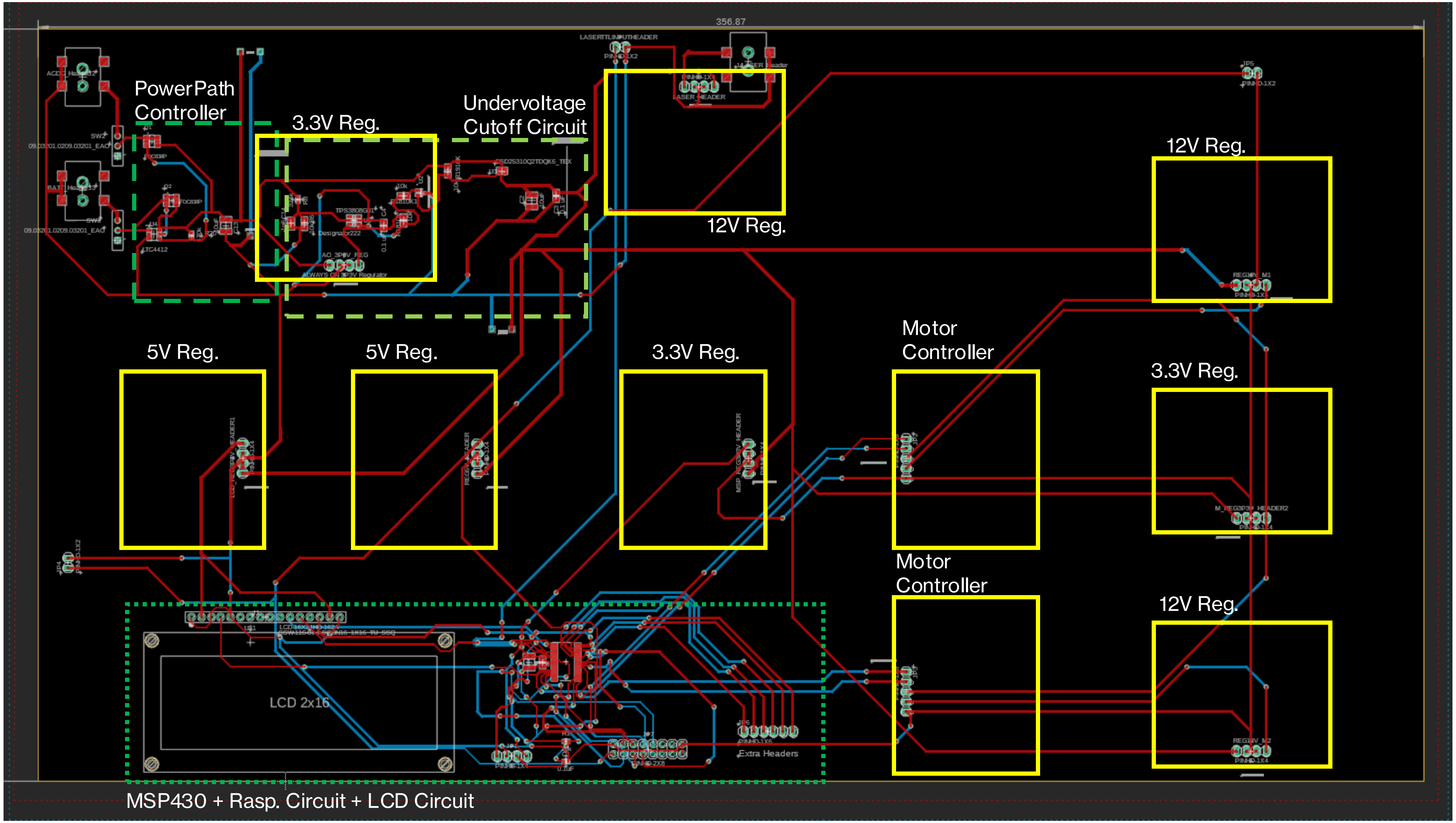




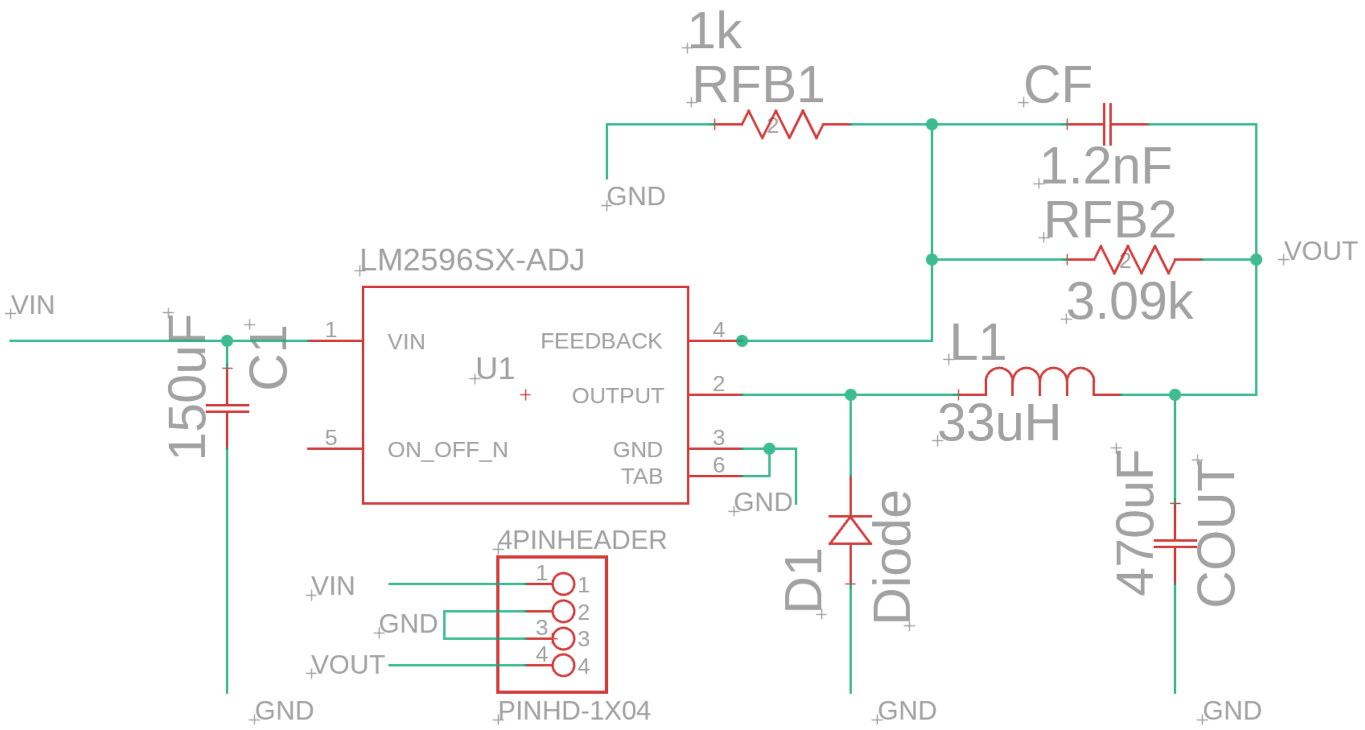
PCB Design

Main PCB Design

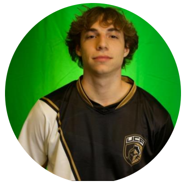
Built from "Overall Schematic"



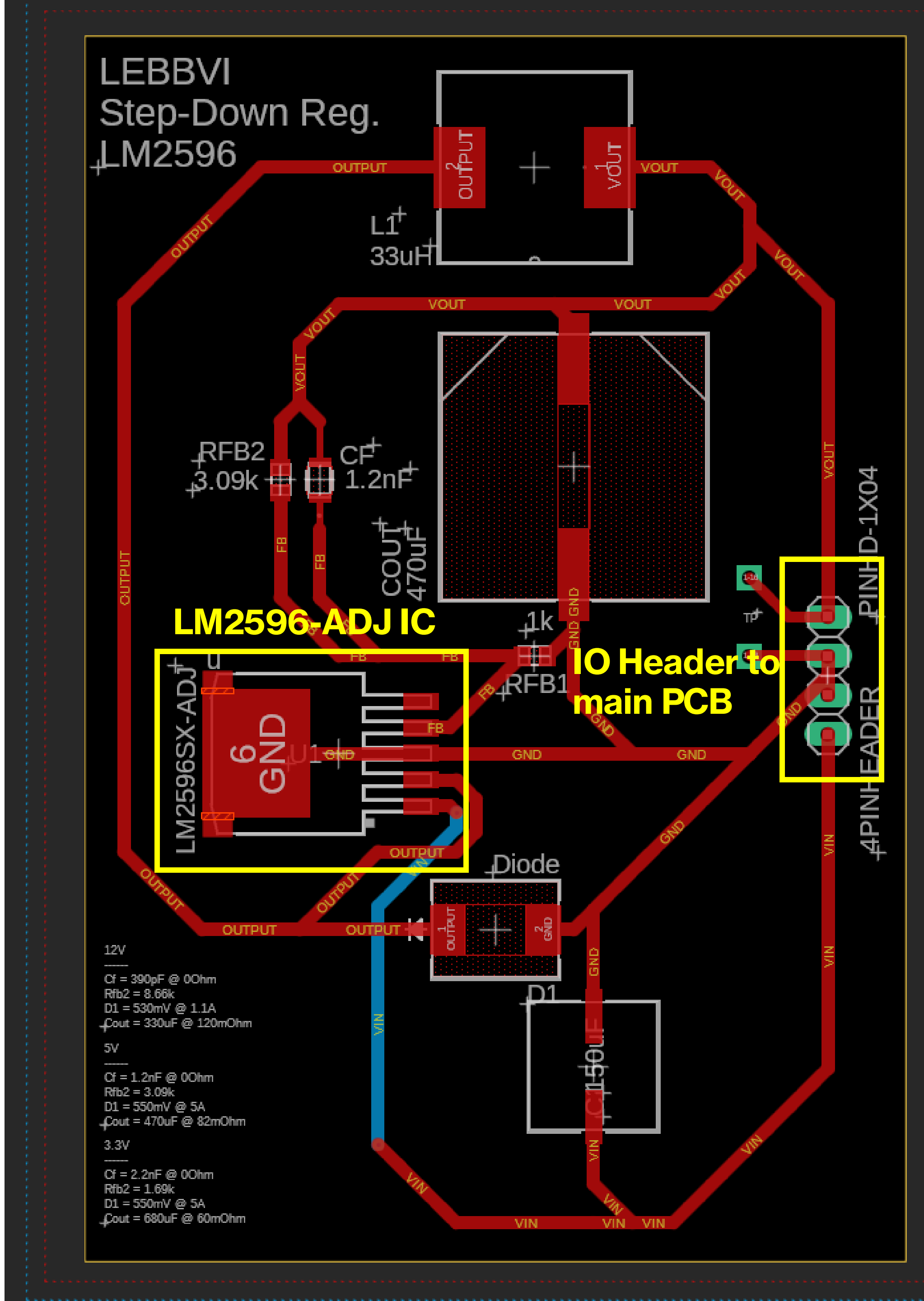
Voltage Regulator Schematic: LM2596T-ADJ

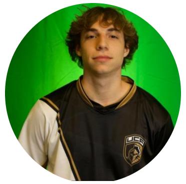


- Same design for each regulator: 3.3V, 5V, and 12V
- The components that change between each regulator:
    - CF, RFB2, D1, and COUT

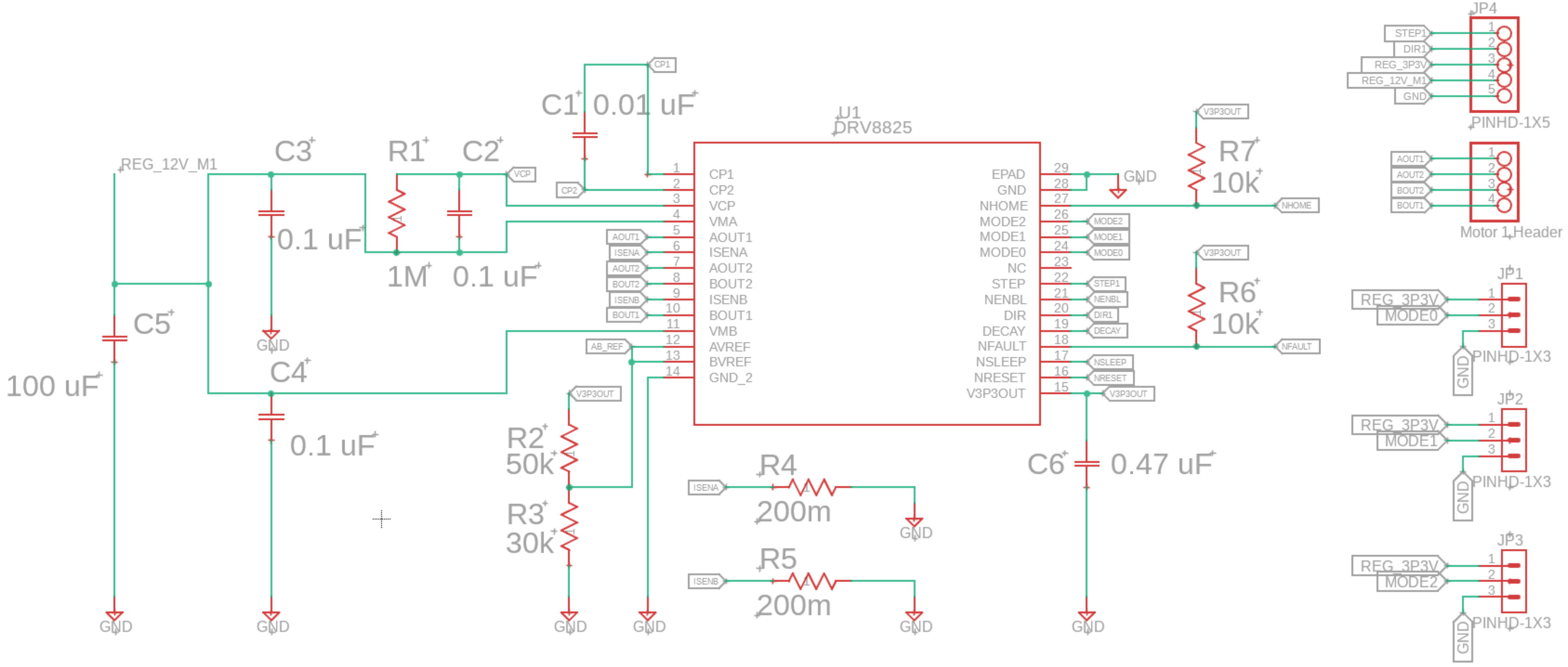


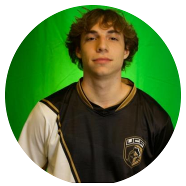
LM2596T-ADJ PCB Design



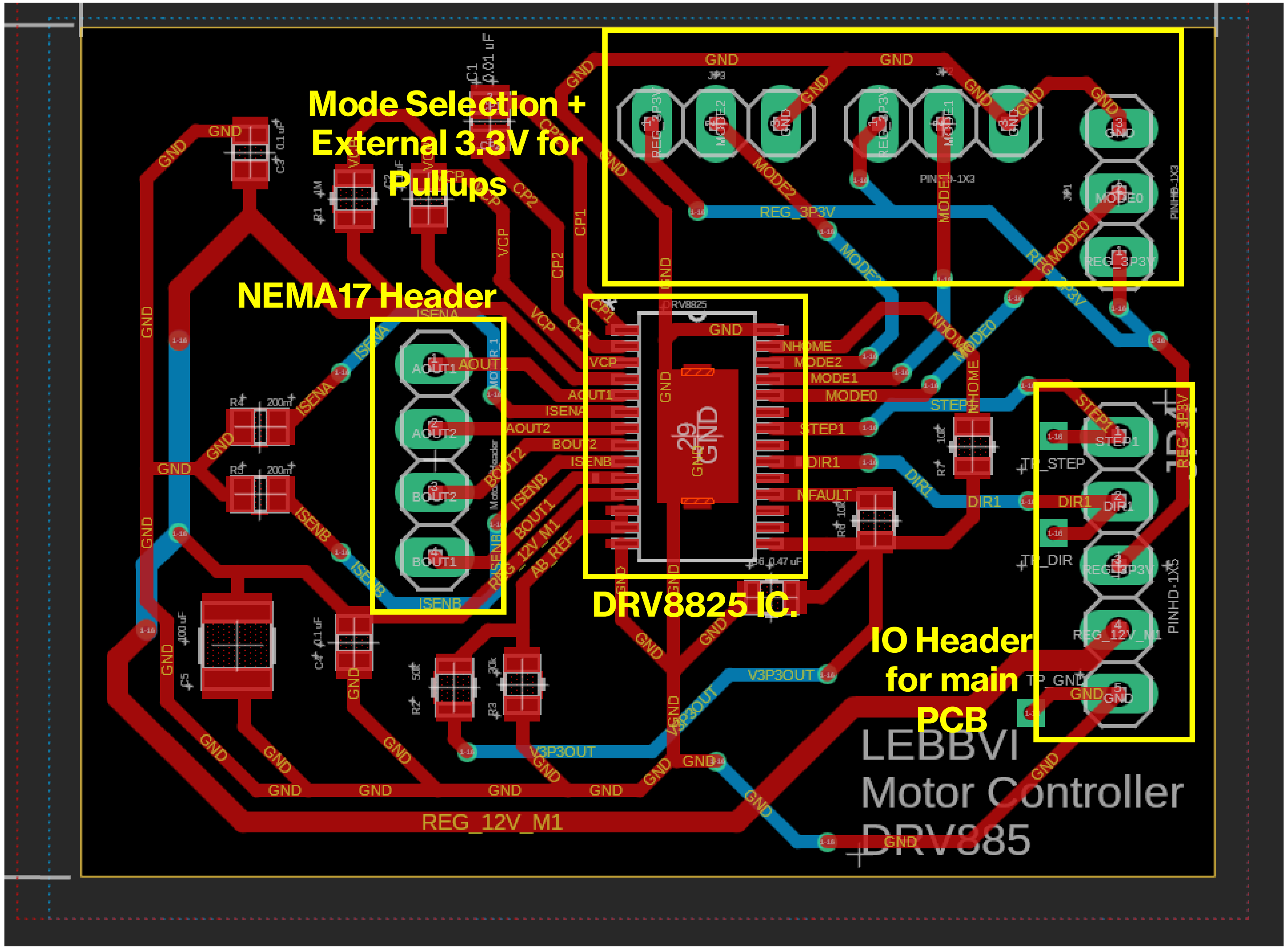


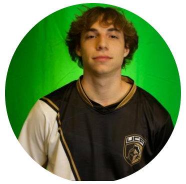
Motor Controller Schematic: DRV8825



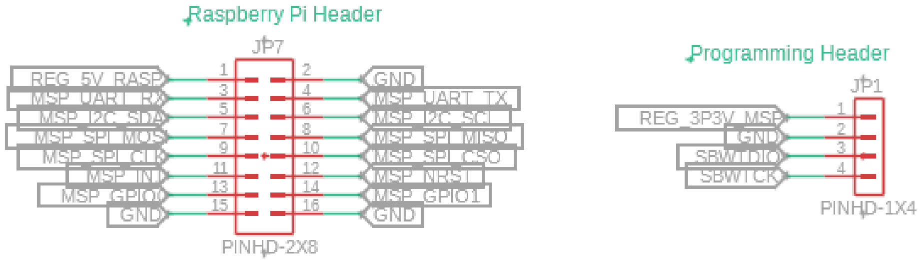


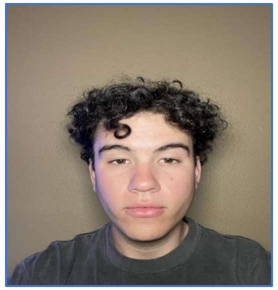
DRV8825 PCB Design





MSP + Headers Schematics

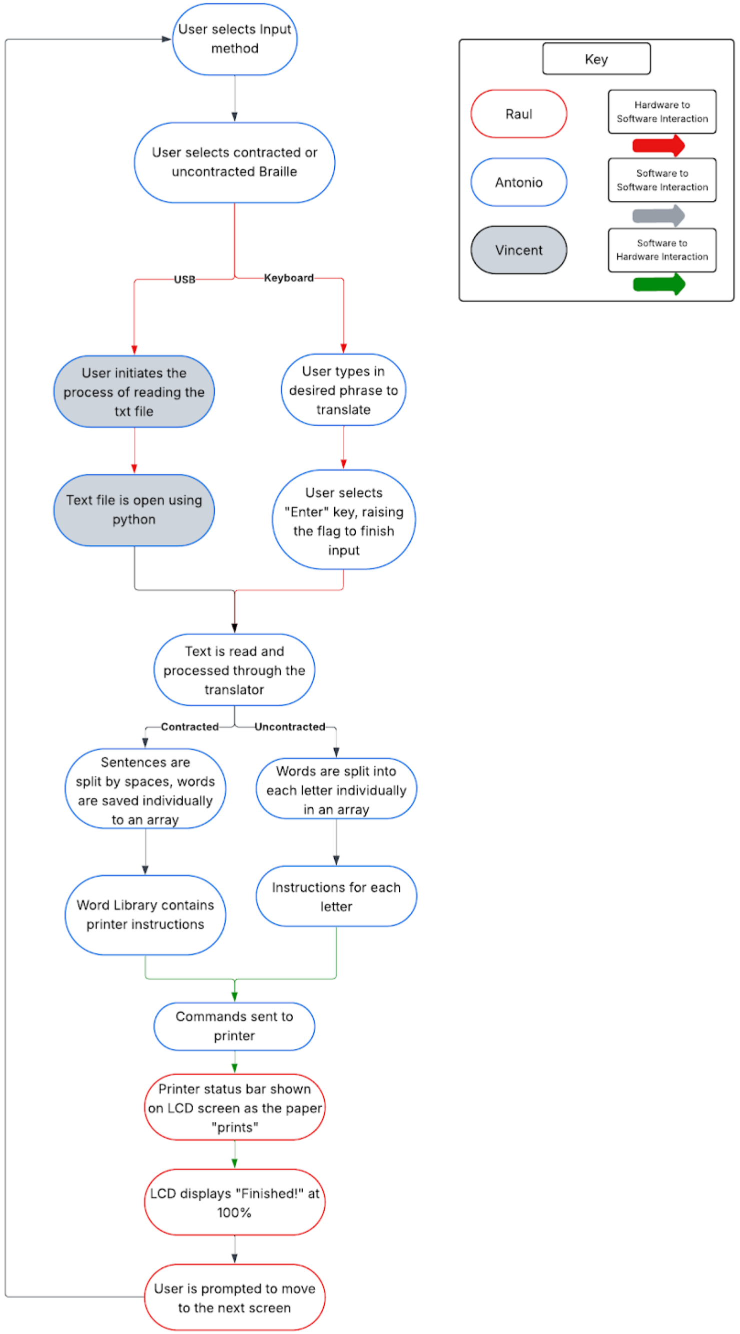




Software Design

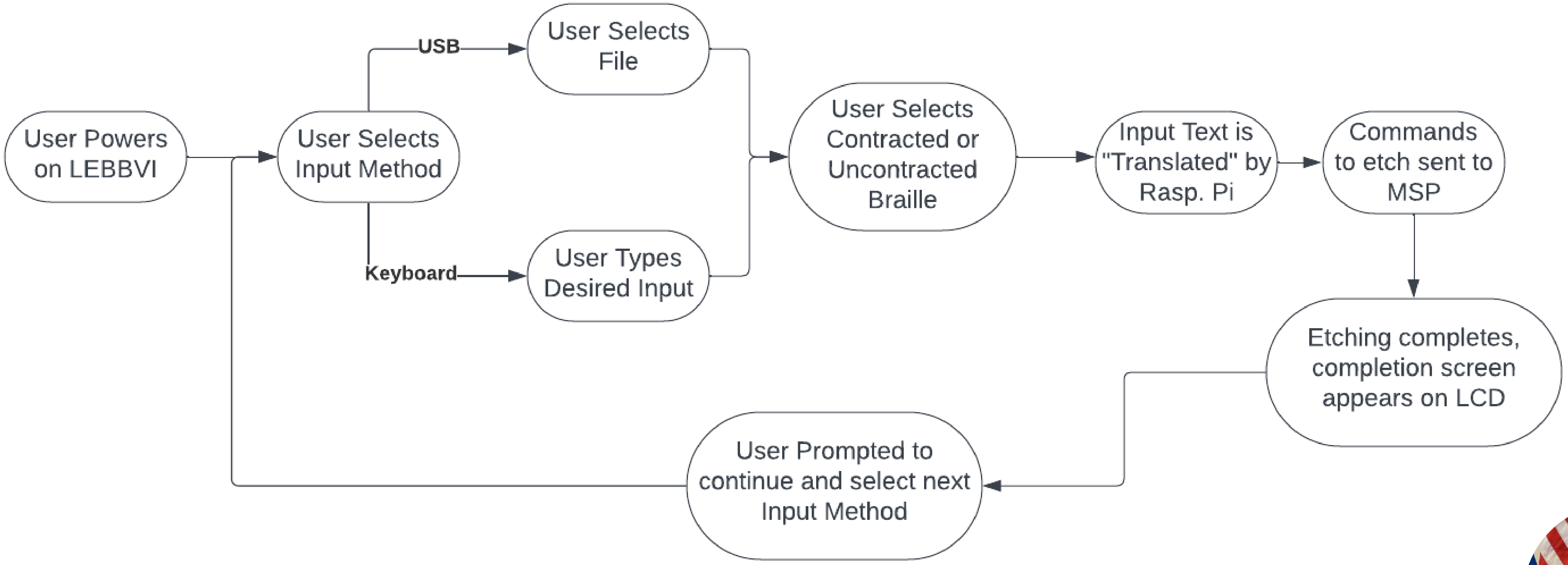
Software Selection and Comparison

- High-level and Low-level operations
  - Python used on RaspberryPi
  - C used on MSP
- Python selected for ease of use
- C selected for prior use



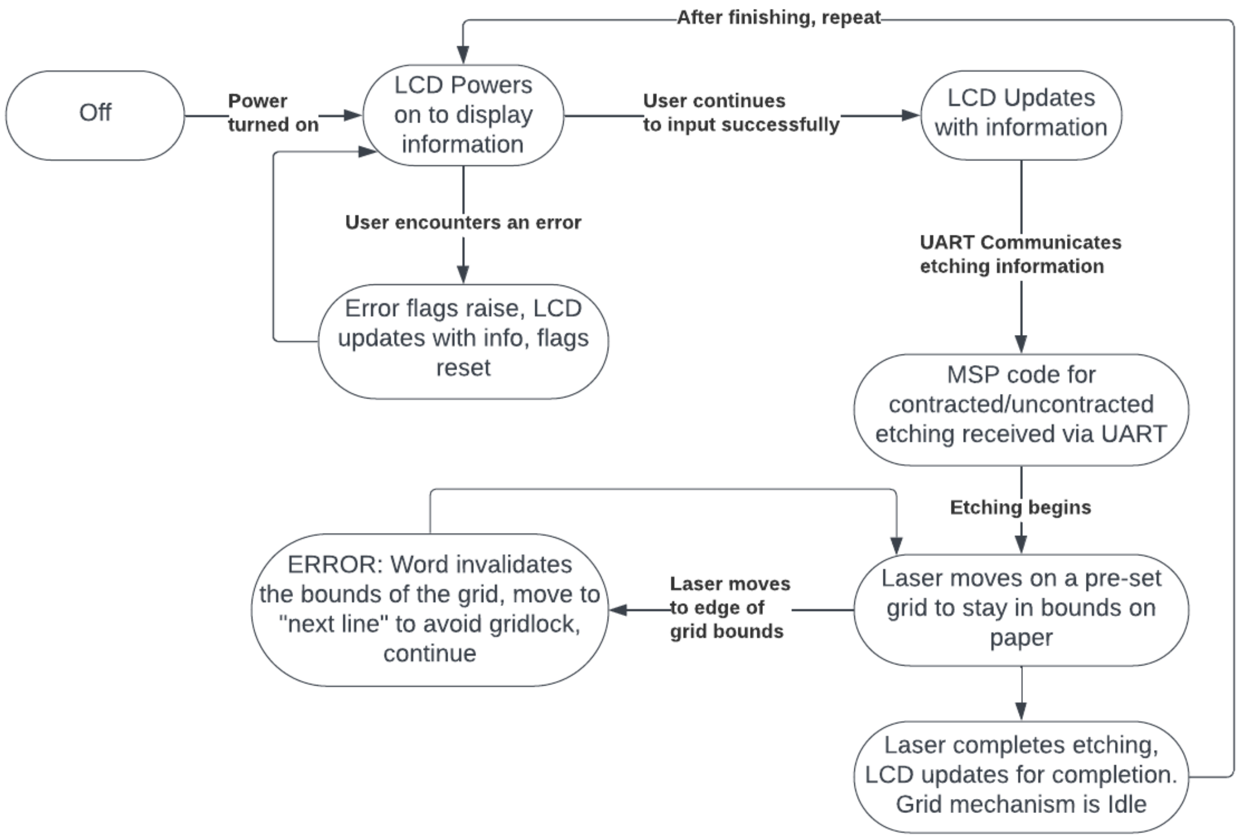
Software Flowchart

Use Case Diagram



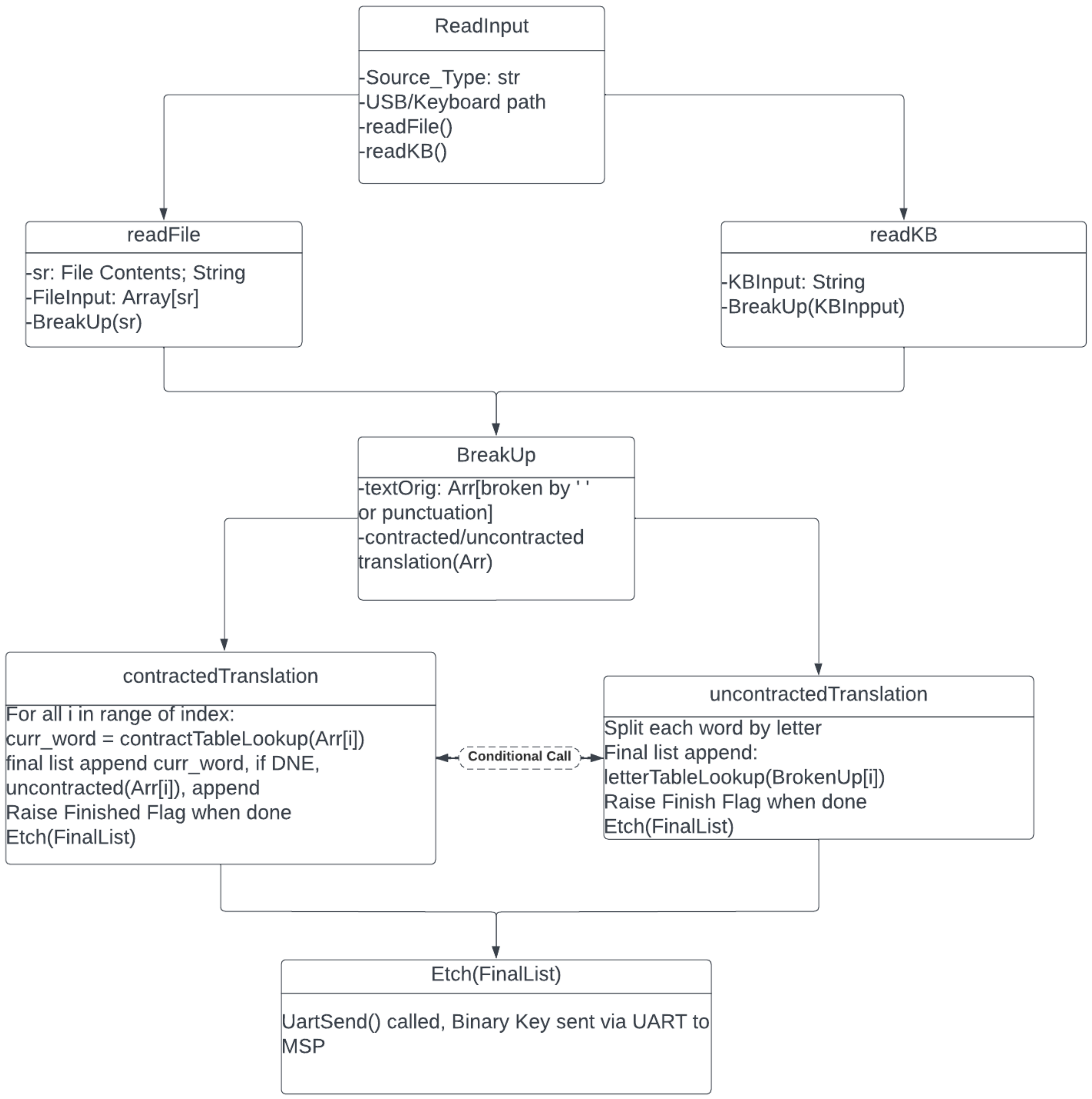
State Diagram

State Diagram

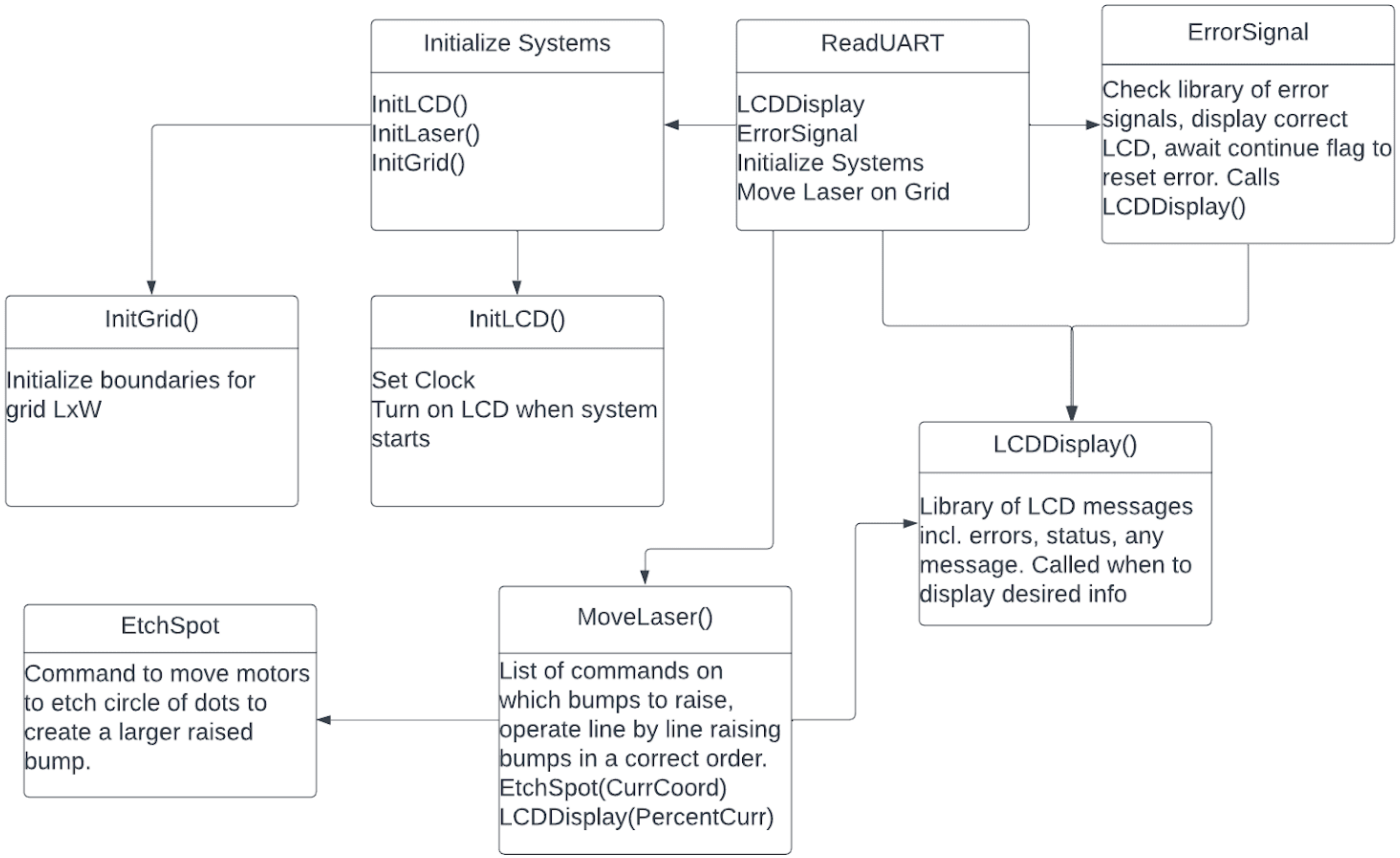


- MSP State Diagram has some alterations
- Main focus on hardware operations and commanding the motors to etch

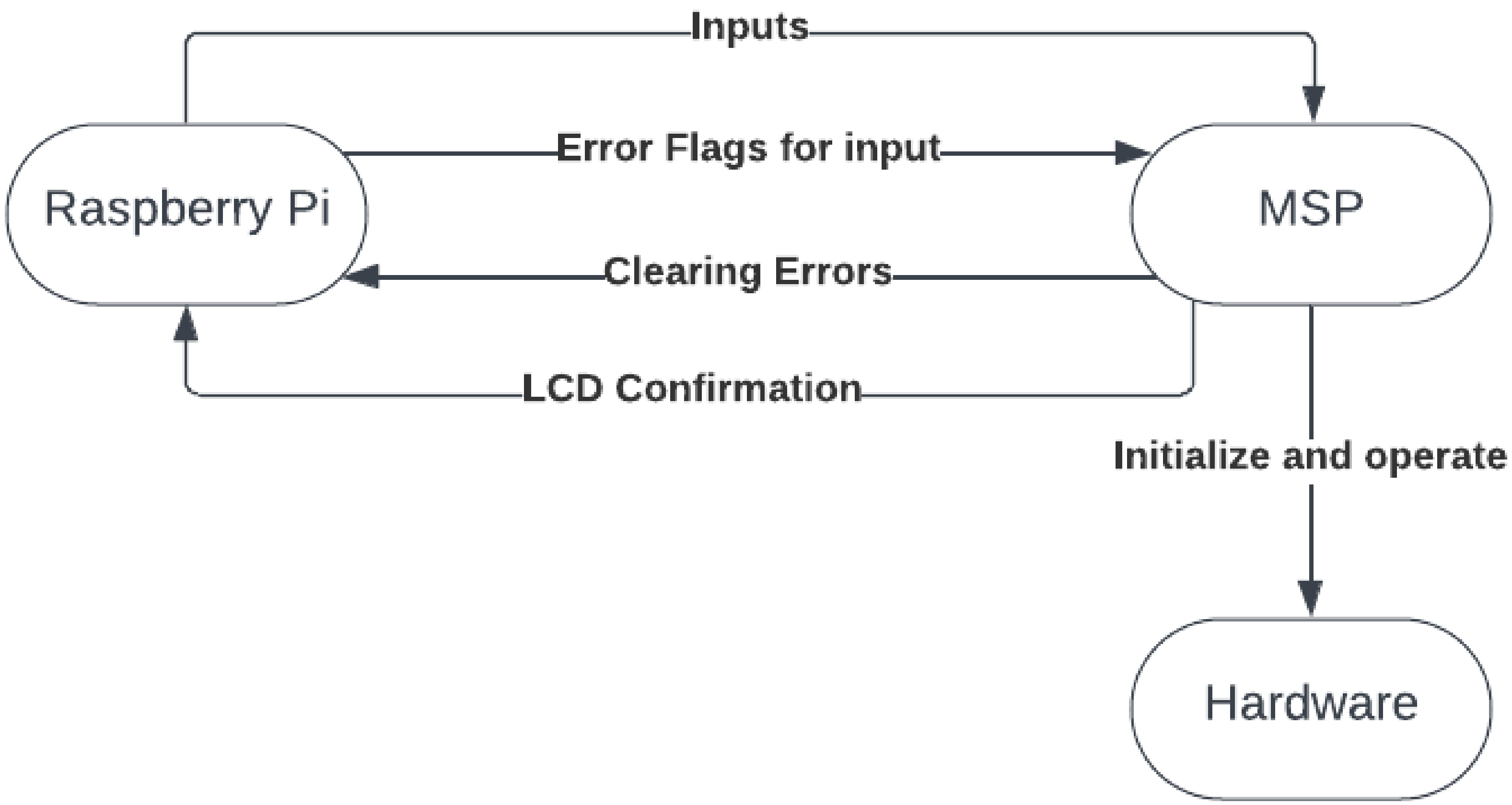
Class Diagram



Class Diagram (MSP)



Data Transfer Diagram

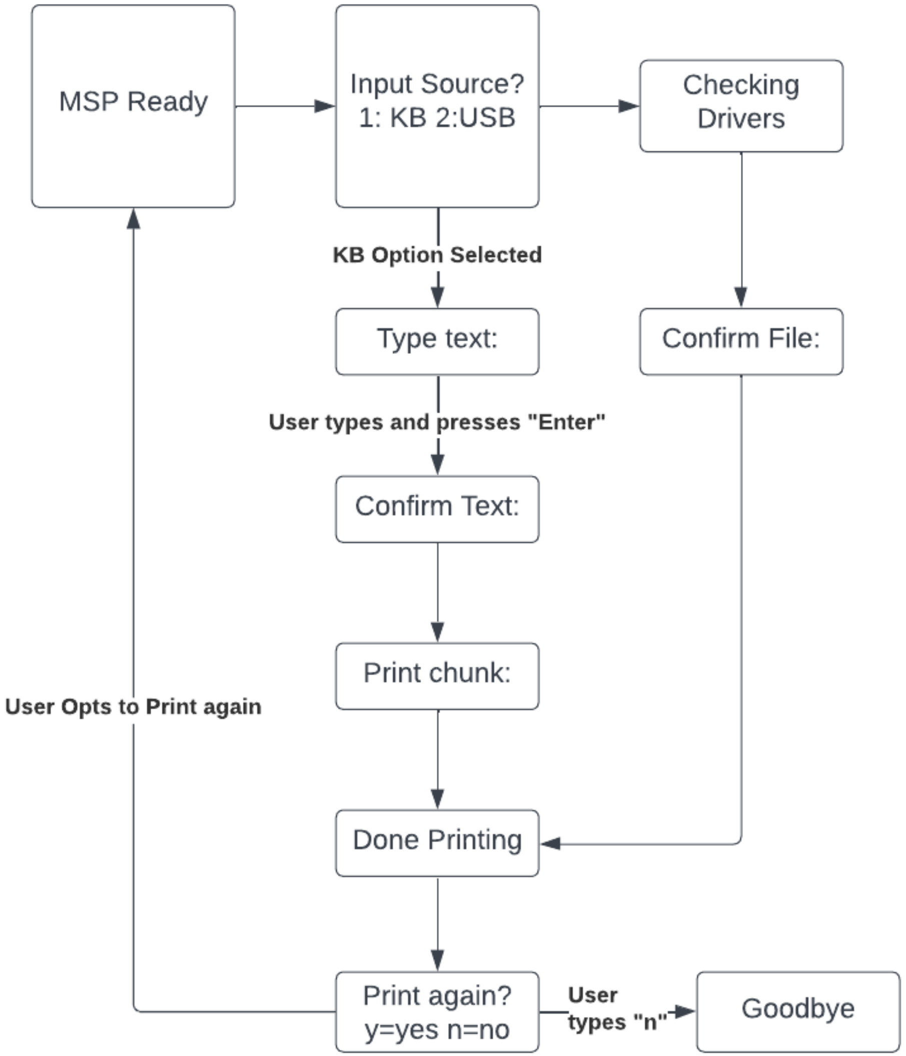


Communication Protocol

	UART	I2C	SPI	WIFI
Pros	Consistent, fast, easily programmable, stable, works especially well on MSP430FR2355 architecture	Simple, Master/Slave system with Pi/MSP. Good for sending packets.	Fast and Reliable	Wireless, very fast. Increase in possible features.
Cons		Requires many pull up resistors, not natively functioning. Requires interrupt-driven code.	Hard to implement Not as "plug and play" as UART	Hardware not present. Requires many layers of software.

Communication Protocol Cont.

- Due to the ease of implementation and the hardware/software capabilities of the MSP430FR2355 and Raspberry Pi Zero the selected communication protocol is UART.
- Messages and acknowledgements sent back and forth between MSP and Pi in order to transmit data.

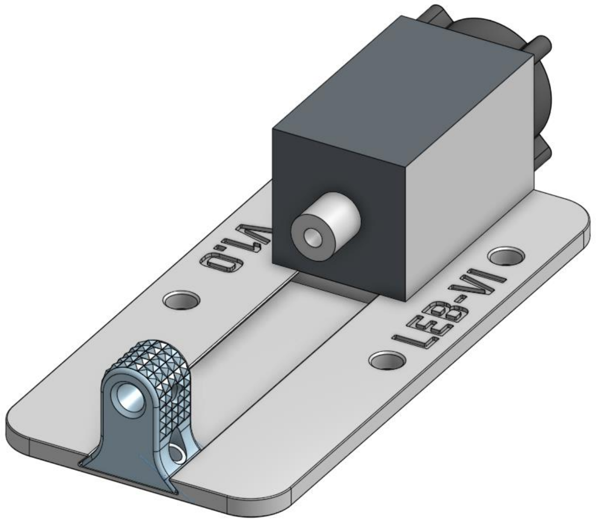


LCD Flow Diagram

- The LCD Displays a message of connecting to the MSP on launch.
- LCD prompts users for input
- Continues through confirmation and updates user on progress
- Signals goodbye on shutdown and MSP ready when it is ready to take in another input.

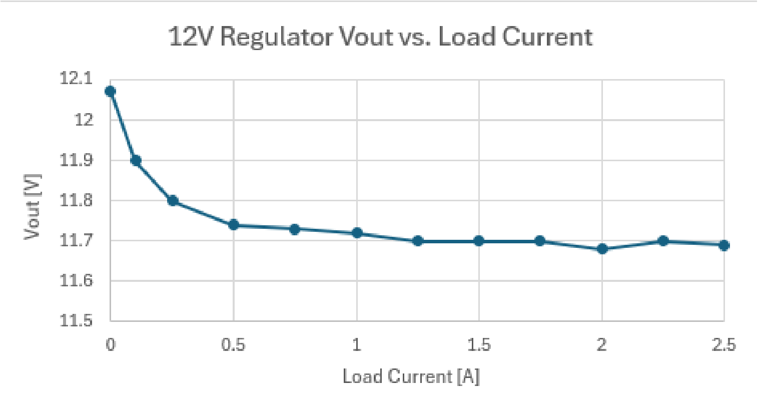
Prototyping and Testing

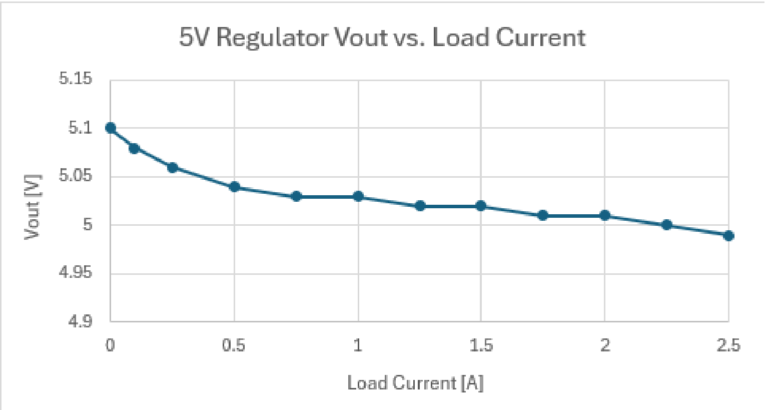
- Early Prototype of Free space fiber coupler
- Free space coupling is too sensitive and expensive for the scope of this project

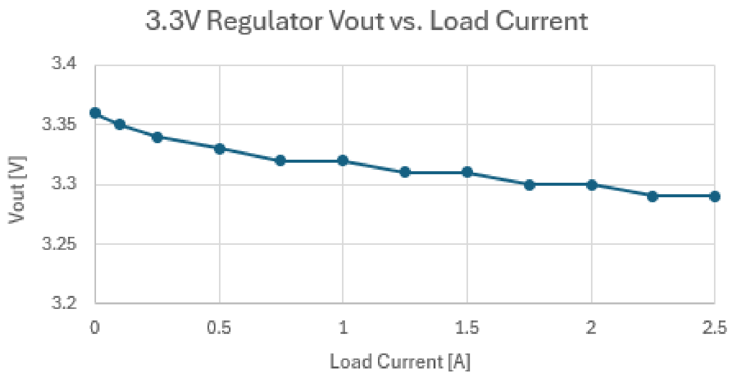


Voltage Regulator Load Testing

- Measurements were taken by putting the regulator on the breadboard and using a bench power supply to Vin set to 17 Volts, and the load connected to Vout, where we began at 0.1A and increased to 2.5A.
- 12V – We see the voltage loss to be ~0.3V after 1A.
- 5V – Voltage loss is very minimal here when increasing the load current.
- 3.3V – Voltage loss is also very minimal here when increasing the load current.
- This shows that our regulators are typically outputting the expected voltage under intense load, with a ~0.3V drop for the 12V regulator.



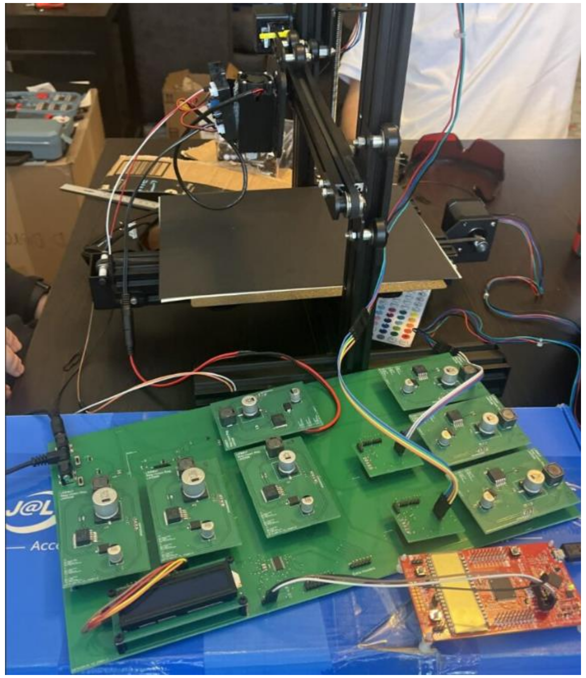




First Prints of Braille

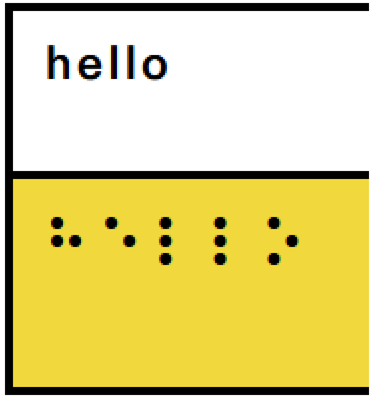
- We first began testing by seeing if the laser can print accurate, tactile bumps to the paper before we tested the logic of the motors that moved the motors to the correct orientation

- Then we hard coded a word to the MSP430 to verify that our motor logic was correct before using UART between the Raspberry Pi and MSP430 which was "hello"

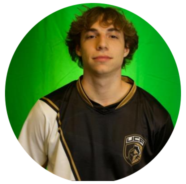


Early testing of fully assembled PCB and printing

First test of printing an array of dots to the paper

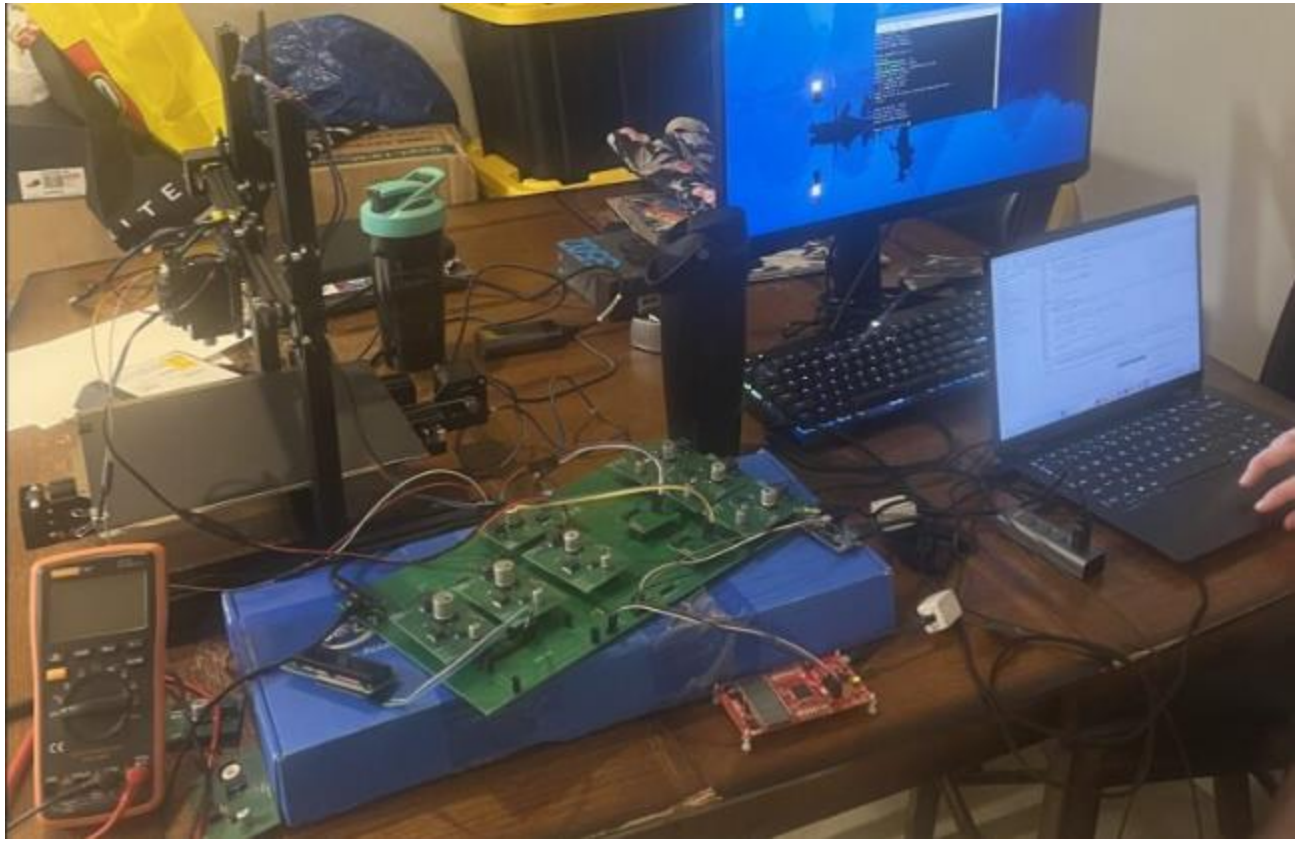


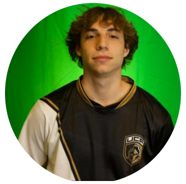
First complete word printed to the paper, slight misalignment on top row. This word is "hello"



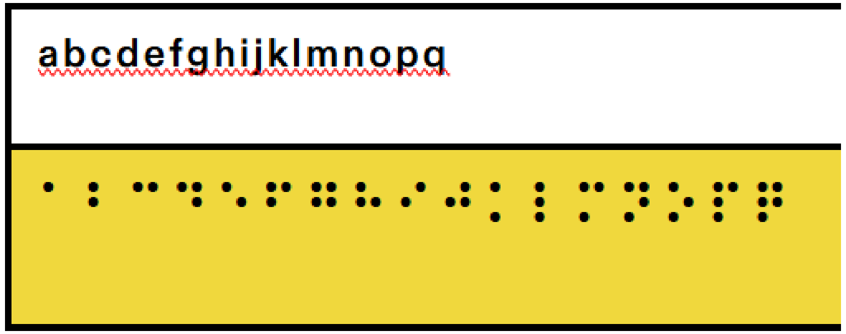
Hardware Prototyping and Testing

- The overall prototype of LEBBVI
  - Fully assembled PCB with the motor controllers, regulators, and laser being powered.
  - Gantry assembled with all peripherals attached; motor, laser, paper tray
  - Raspberry Pi Zero 2 displaying to the monitor while MSP430 dev-board programs the on-board MSP430FR2355 chip.



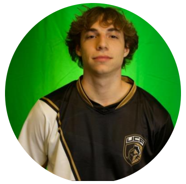


Further Braille Testing with UART Communication

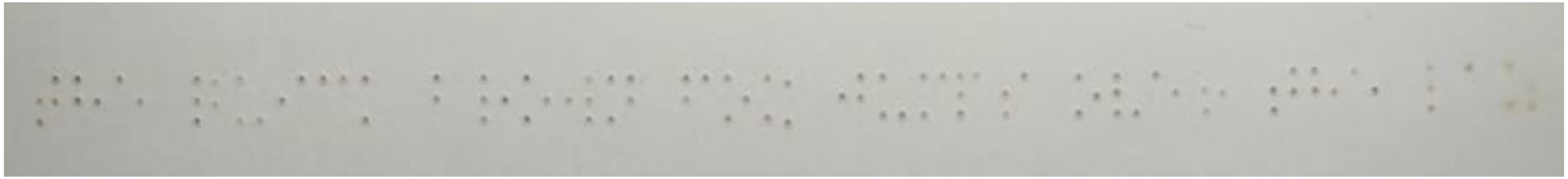


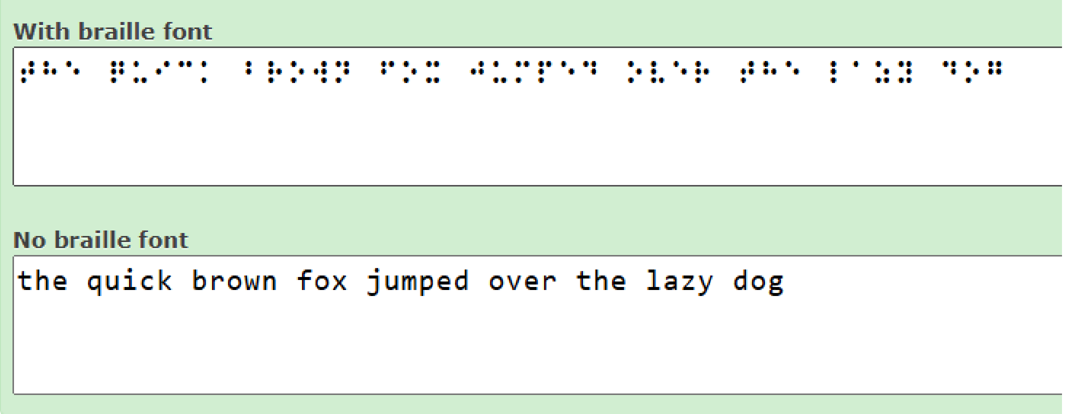
A - Q braille translation

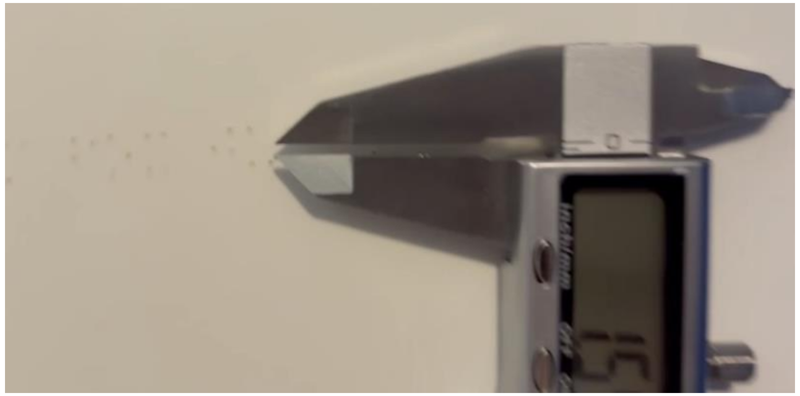
- We printed out 17 letters of the alphabet (a to q) from the console prompt on the Raspberry Pi that told the on-board MSP430 how to move the motors and when to fire the laser



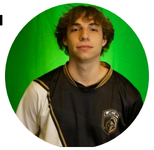
Full Print of a Pangram: Accuracy, Speed, Dot Size







- Printing a pangram displays translation accuracy, and here we see 100% accuracy from our English to braille translation
- The word cuts off at "laz" as we ran out of paper space
- This printed 39 characters (including spaces) in around 11-12 minutes, showing a print speed of ~3.4 - 3.5 characters/minute
- Measuring the dot size, we measure around 1.5mm for each dot, with some dots ranging from 1.3-1.6 in diameter
- This testing proves we are meeting our 3 "demoable" engineering requirements.



Software Testing

- Translation logic testing in VS Code using Python
- Development Environment Debugging
- Upload Code to Dev-Boards to test UART functionality
- Error checkers ingrained in code to error and limit test functionality
- Use of terminal to debug

Software Prototyping

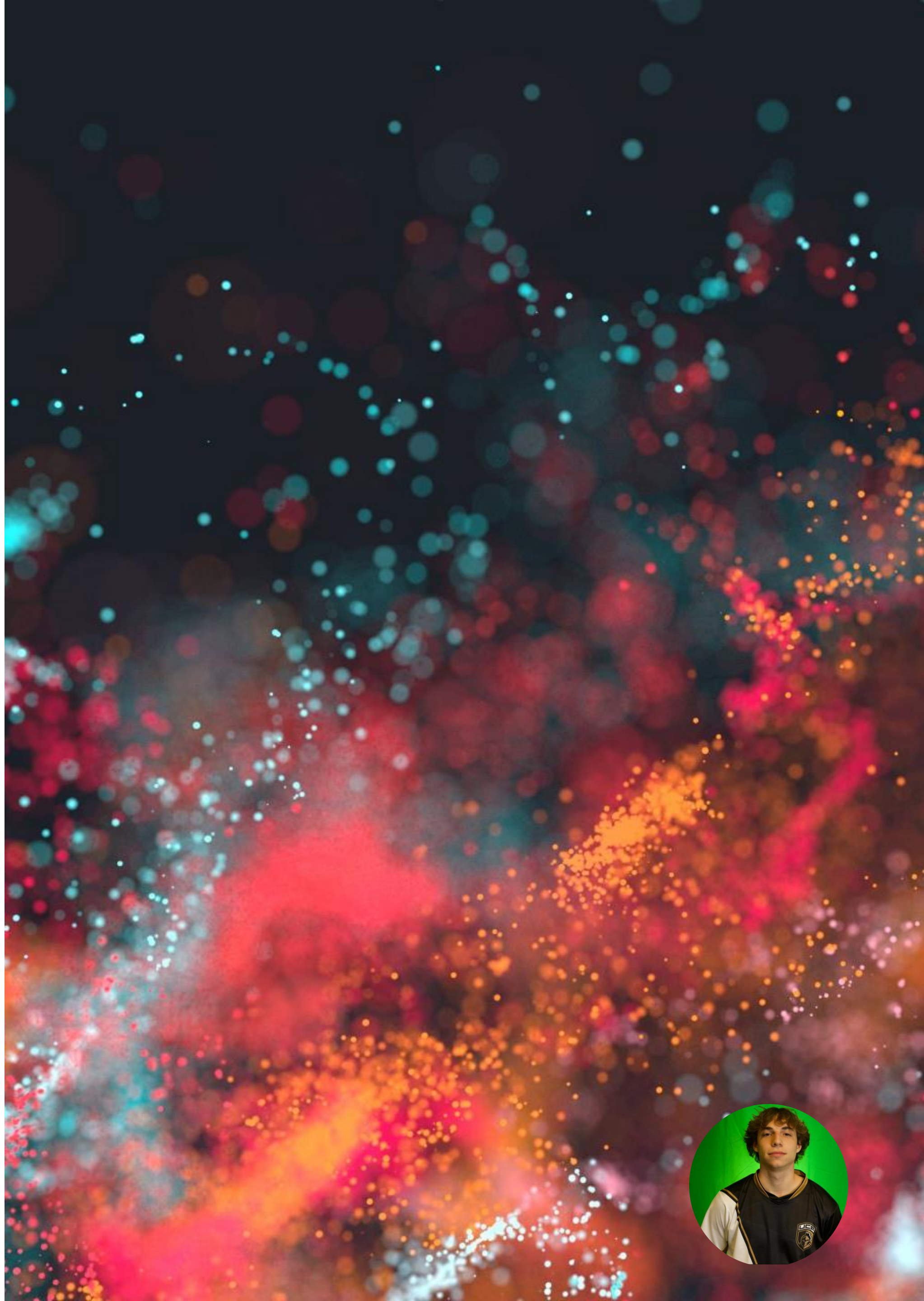
- Completed code uploaded to boards connected to hardware prototype configuration
- Testing of code validity and functionality
- Testing in phases and then merging of functions to ensure operation

Pre-existing Software

- Laser Etching Software itself is not affordably available
- BrailleBlaster – Transcription tool
  - Not embeddable software
  - Desktop application only
- LibLouis – Open Source Braille Library
  - Not embeddable software
  - Multilanguage and contracted/uncontracted support

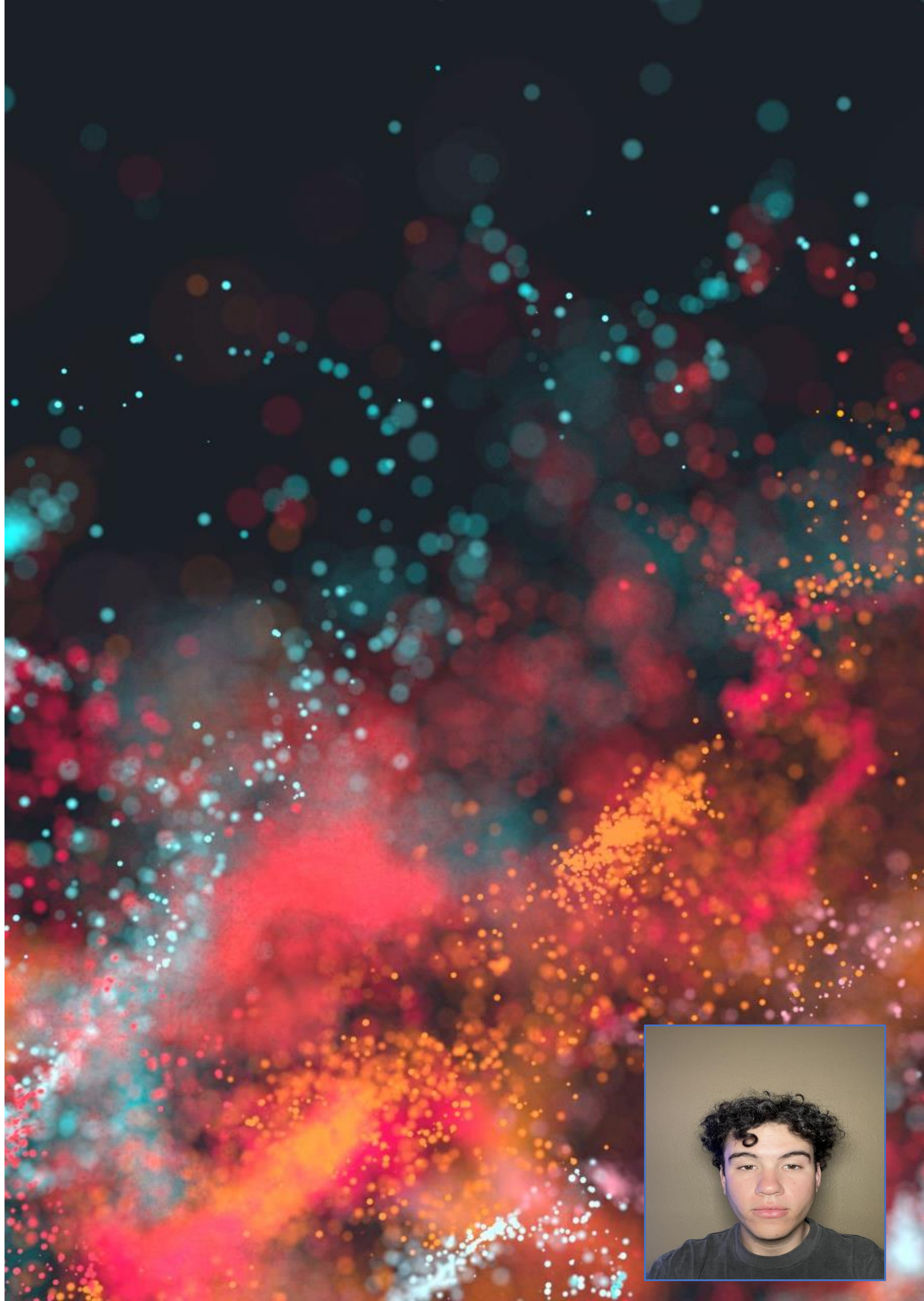
Performance Evaluation and Success

- The way we will evaluate the performance of our project is through successful prototyping and the overall function of our system
- Custom PCBs are working as intended and supplying everything the power it needs. We have successfully prototyped our braille printer which was shown in the previous slides.
    - The printed dots have accurate measurements to typical braille dots seen in the real world.
- UART communication is also working as intended and the user can confidently type out a sentence and see that our system can print out their sentence in braille.



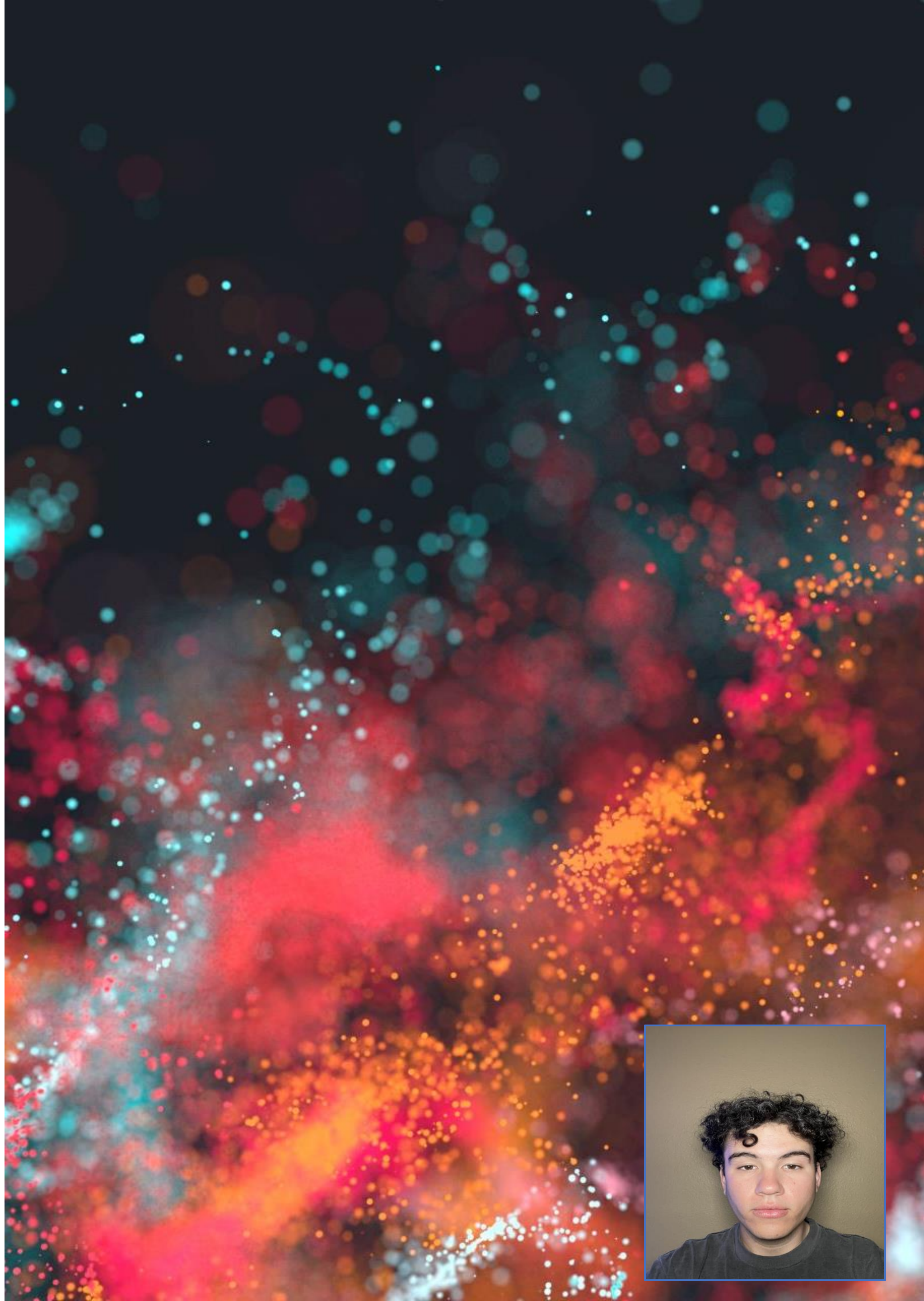
Problems and Proposed Solutions

Problem	Solution
Increasing the font size while minimizing technical issues	We found a special kind of paper called "Swell Touch Paper" made for Braille that swells up when heated to avoid future problems such as burning through standard paper
MSP430FR6989 too large/dense to solder	Moved to MSP430FR2355 since it is a smaller package with easier integration

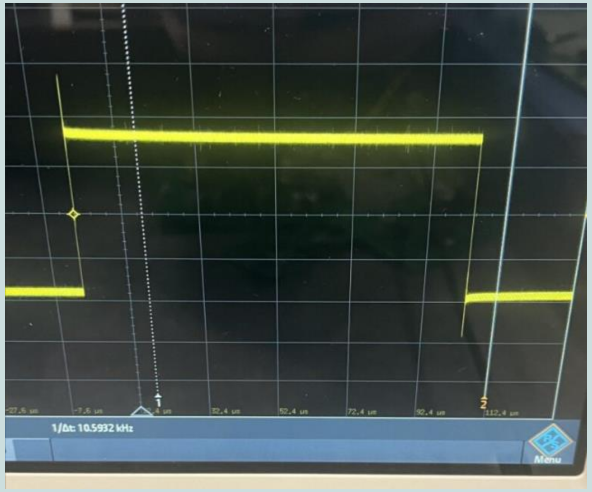


Problems and Proposed Solutions

Problem	Solution
Programming the MSPFR2355 on the custom PCB	Programming the MSP430FR2355 on the custom PCB initially failed because Code Composer could not recognize the device. The wrong target MCU was selected, the SBW programming pins were not wired correctly at first, and a capacitor with an incorrect value on the reset/SBW line was interfering with the programming signals. After selecting the correct target in Code Composer, fixing the SBWTDIO and SBWTCK connections, and removing the incorrect capacitor, the MSP430 became fully programmable and communication worked as expected.
Stepper motors causing jerky movement and dot misalignment	During early motion testing, excessive torque led to sharp, jerky belt movement and misaligned Braille dots. The DRV8825 drivers were supplying more current than the mechanical system could handle, resulting in abrupt acceleration. By lowering the torque through proper Vref tuning on the motor drivers, the movement became smoother and more controlled, which produced consistently aligned and precise dot placement.



Problems and Proposed Solutions

Problem	Solution
UART Communication issues 	We ran into issues with UART communication between the MSP430 and Raspberry Pi. With the help of Dr. Weeks, we debugged what was wrong with our software. We solved this issue by finding a TI forum for our specific chip with a working UART function. We then verified that UART was working as we had the Raspberry Pi send over 0x00 to the MSP and then have the MSP transmit if back. Hooking up the oscilloscope on the TX pin of the MSP, we see a wave that displays 0x00, confirming UART functionality. We then implemented the UART function in our code and continuned to work on the project.

Major Challenges

- The major challenge of this project was getting the MSP and Raspberry Pi to communicate correctly and then tuning the motors and laser through the software to print out repeatable and consistent dots with tactile bumps.
  - Countless hours were put into the software side to be able to accurately print out the dots
- Testing the PCBs also proved to be a major challenge as there were minor mistakes in the first design
  - Incorrect component values, pins left floating, etc
  - These were fixed with the help of Dr. Weeks, and no re-design was needed

Proposed Improvements

- Overall improvements that can be made if we were to build a new one are:
  - Purchasing better equipment
  - Making PCBs more compact
    - Even re-designing them in the process
  - Polishing the look of the overall system so it doesn't look so "prototype-esque"
- Continuing to build the software to make it as perfect as possible
  - This would require many more days of testing and countless of hours debugging, however.

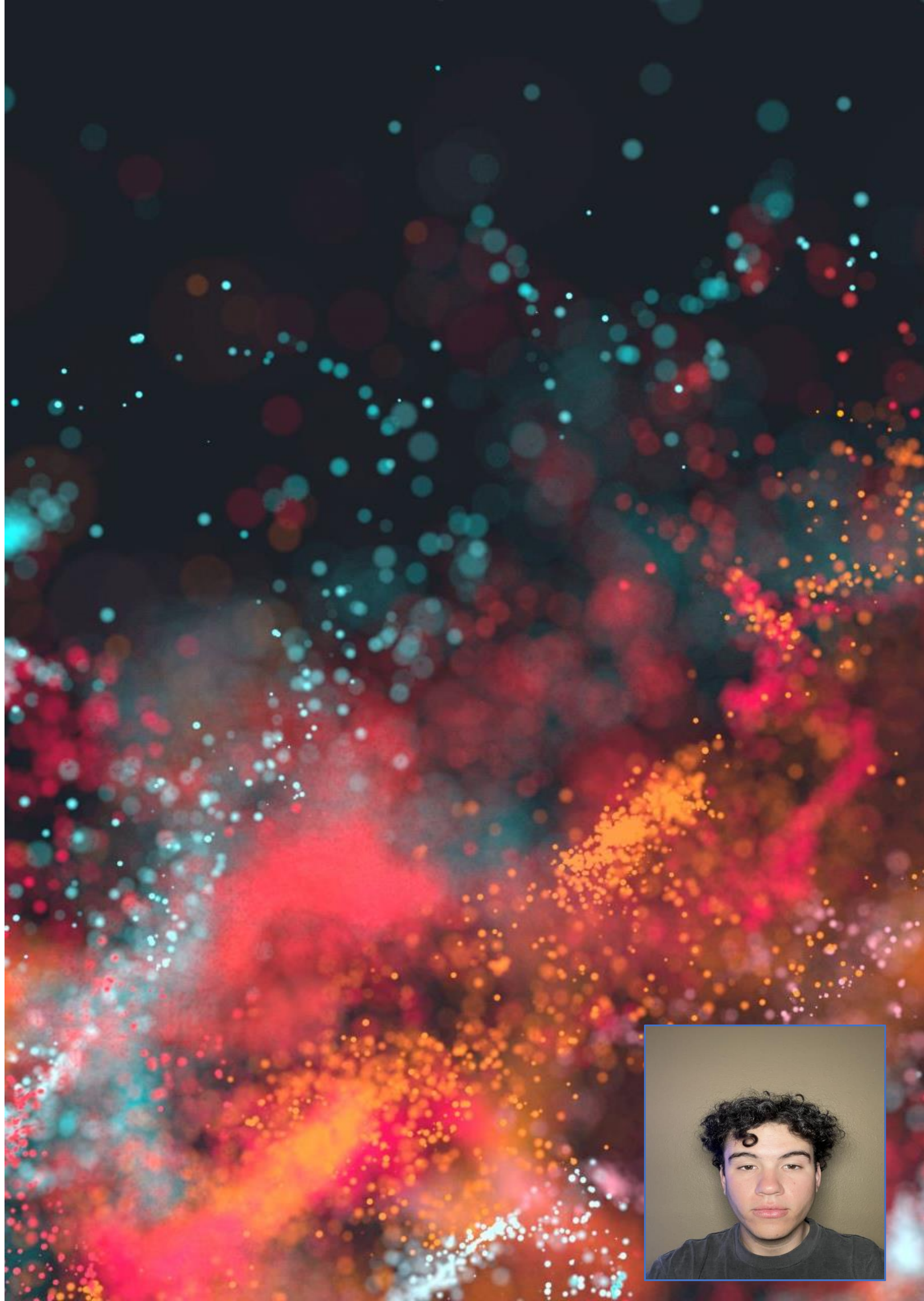
Administrative Content

Project Budget Overview

Item	Estimated Cost
Laser	$45.00
Stepper Motor(s)	$30.00
PCBs	$90.00
PSU System	$70.00
PCB Components	$90.00
Total	**$330.00**

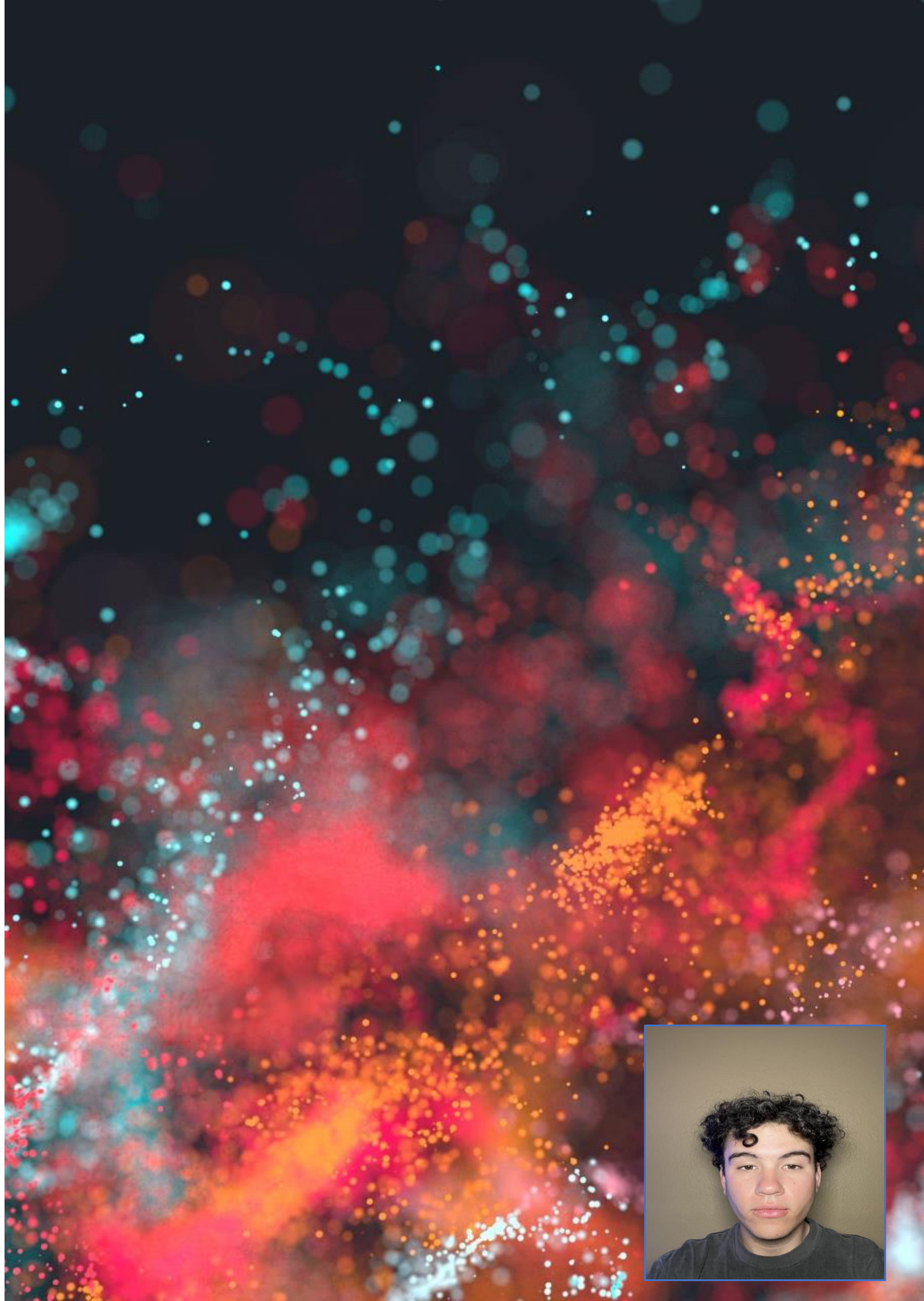
Table 18: Estimated Cost

Original Budget set in Senior Design I



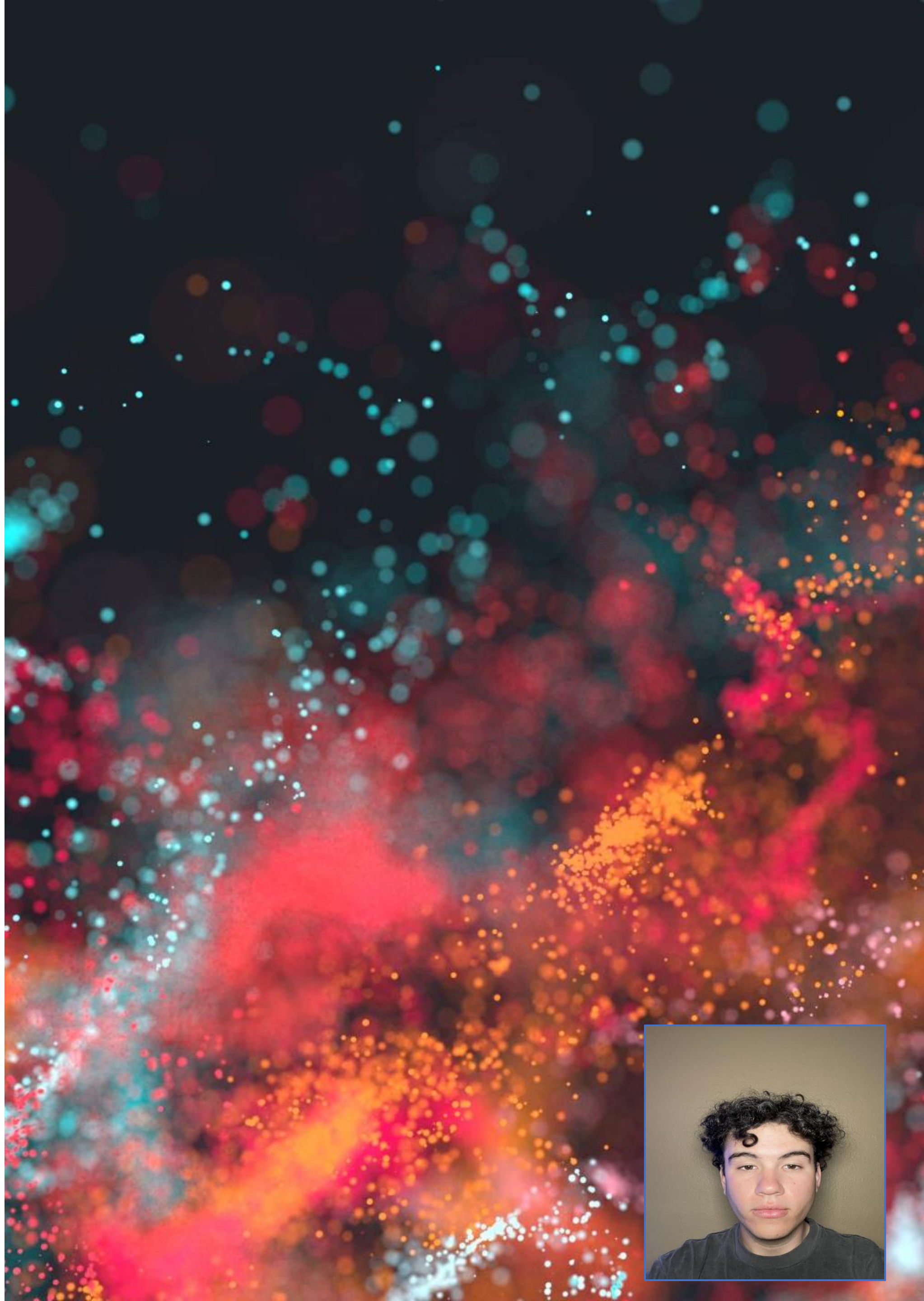
Bill of Materials

Item	Quantity	Price
PCB @JCLPCB	1	$200
Components	1	$250
NEMA 17 Stepper Motor	2	$28
5V Fans	2	$18
SHNITPWR 3V ~ 24V 3A 72W Adjustable AC/DC Power Supply	1	$18
4S 30A Li-ion 18650 BMS	1	$4.50
Samsung 25R 18650 2500mAh 20A Battery	8	$20
Oxlasers 500mW 405nm UV Laser	1	$48
Raspberry Pi Zero WH	1	$25



Work Distribution

Student	Primary Tasks	Secondary Tasks
John Childs	PSU, Power regulation, buck/boost layout, Power Distribution	PCB Design
Raul Perez	PCB, MCU firmware, motion control	UI design
Antonio Duford	UI design, comm stack, test automation, braille translator	Motor control system
Vincent Pagliuca	Laser optics, safety interlock, ablation tests	Soldering, Building Housing



Thanks for watching!