CSCE 483 Individual Notebook

Abhishek More
TA: Karl Ott

Table of Contents

Resources

Github:
https://github.com/JacksonHagood/MotorImpairmentKeyboard

Miro:
https://miro.com/welcomeonboard/b0NXS1pJdzRQSzdQakpkNjI0cDNxdUU2V1p2cGFwN0tDMnFHNXdLS1F6ZU1ObXZZa2szSkpEV2xLVzBRTnBLeXwzNDU4NzY0NTIxNjIzMTg0NDkz?share_link_id=517566475884

Jira:
https://keyboardists.atlassian.net/jira/software/projects/MIK/boards/1

Proposed Idea

We want to build a keyboard for motor-impaired users to be able to efficiently enter input to their keyboard, at an acceptable price. The keyboard keys will be inset into the keyboard, making it hard for users to accidentally hit keys. The keyboard will also provide three macro keys and LCD screens which will allow the user to quickly select and enter auto-complete suggestions.

09/21/22

I ordered the hardware. The hardware includes:

- Raspberry Pi Pico
- Key Switches
- LCD Switches

ORDER PLACED September 21, 2022 | TOTAL $23.79 | SHIP TO Abhishek More | ORDER # 112-5766288-2636239 | View order details | View invoice

Running late

4Pcs LCD1602 1602 LCD Module Blue Screen 16x2 Character LCD Display with IIC I2C Serial Interface Adapter Module

Buy it again

Track package | Return or replace items | Share gift receipt | Get help | Write a product review

Archive order

ORDER PLACED September 21, 2022 | TOTAL $43.29 | SHIP TO Abhishek More | ORDER # 112-7125053-8956203 | View order details | View invoice

Was expected by Monday

Gateron ks-27 Low Profile Key Switches for Mechanical Gaming Keyboards | Plate Mounted (Gateron Red, 90 Pcs)

Buy it again

Track package | Get product support | Return or replace items | Share gift receipt | Get help | Write a product review

Archive order

ORDER PLACED September 21, 2022 | TOTAL $20.82 | SHIP TO Abhishek More | ORDER # 112-3099381-5369040 | View order details | View invoice

Running late

Raspberry Pi Pico with Pre-Soldered Header Microcontroller Mini Development Board Based on Raspberry Pi RP2040 Chip,Dual-Core ARM Cortex M0+ Processor, Flexible Clock Running up to 133 MHz

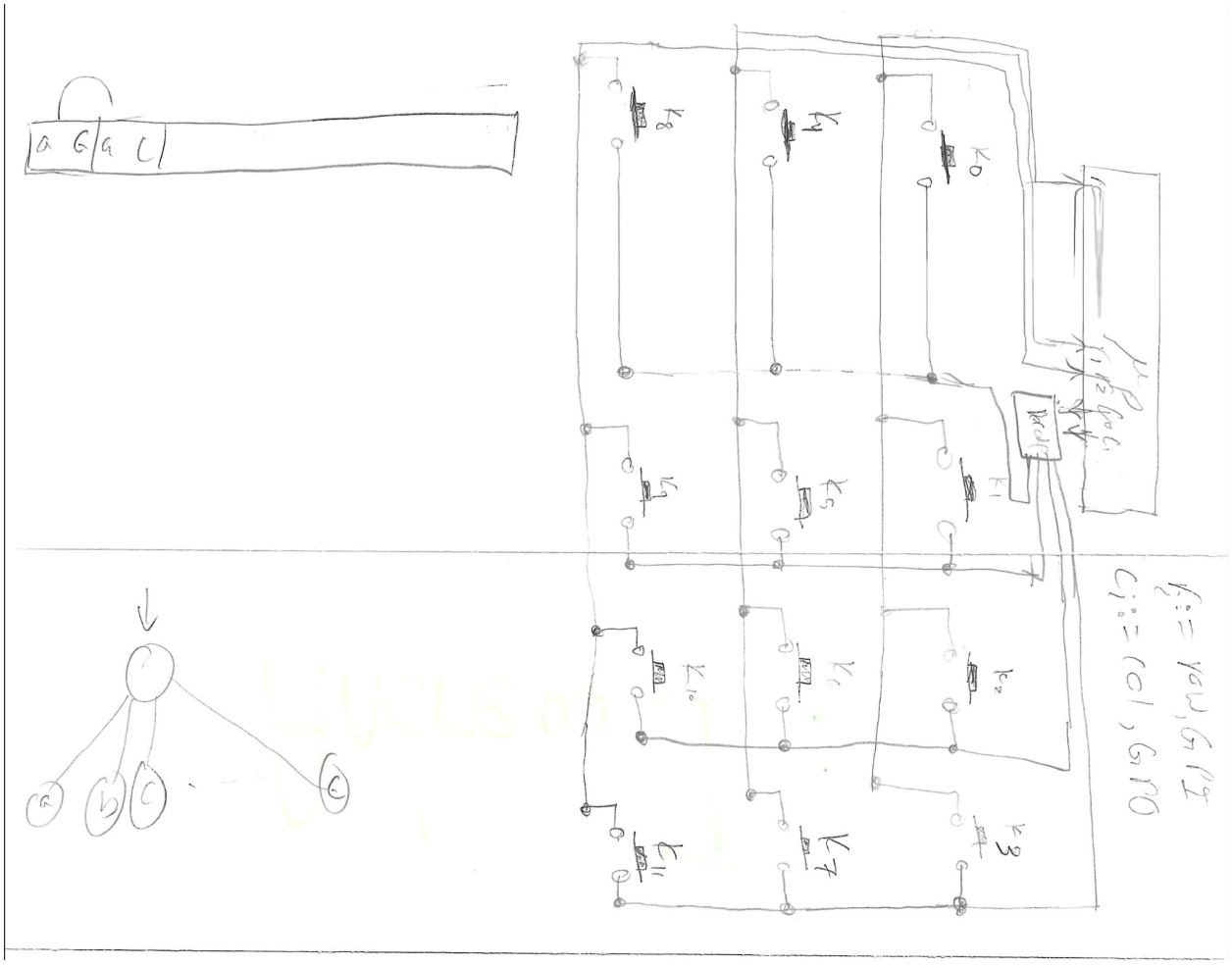
Buy it again

Track package | Return or replace items | Share gift receipt | Get help | Write a product review

Archive order

09/26/22

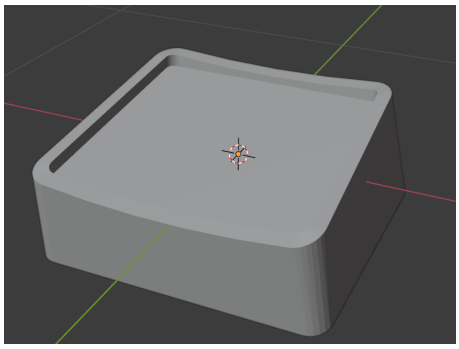
Discussed the circuitry for the keyboard PCB with the hardware team. Due to the limitation of 26 GPIO pins on the raspberry pi pico, we had to incorporate multiplexors into the design.



09/29/22

We discussed the keyboard layout. The keyboard layout must have the necessary keys. The position of the macro keys was discussed, as well as non-alphabetical keys, like arrow keys, delete, and function. Below is the keyboard layout we decided on.

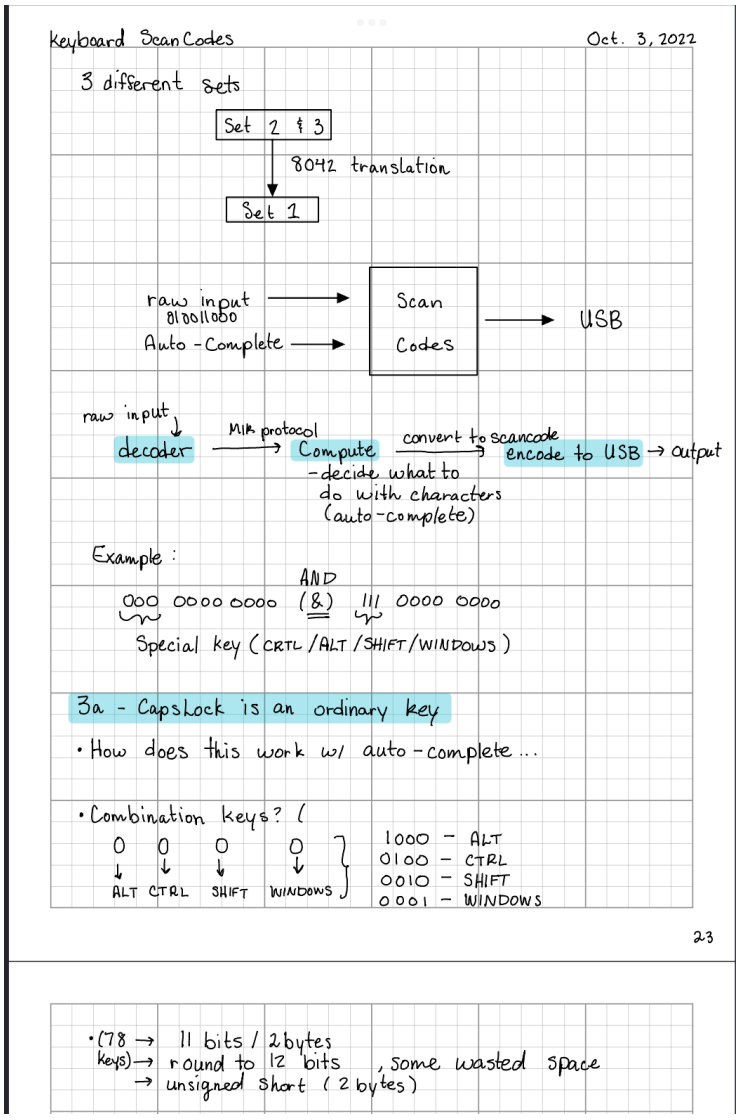
This allows us to finalize the purchase of the key switches and the quantity of keycaps to 3D print.



10/3/22
Sending key input to the computer

https://www.scs.stanford.edu/10wi-cs140/pintos/specs/kbd/scancodes.html#toc9

We discussed sending key input to the computer. There are several problems with this, including the usb protocol and the timing of the key send. For example, if someone hits control and c at the same time, we have to figure out which key to send. We looked into scan codes, which are codes that can be sent to the computer, rather than an ascii value.



10/5

We ended up deciding the encoding to send key presses to the keyboard. We will make our own custom encoding called MENA, which will include the modifier key, as well as additional keys such as arrow keys and caps lock

Also, we ran into a hardware shipping issue. I ordered the parts from my amazon account but they did not arrive. We decided to refund that order and order parts from another account.

Order Placed: October 5, 2022
Amazon.com order number: 113-1701191-9429020
Order Total: $74.94

Not Yet Shipped

Items Ordered	**Price**
1 of: *Raspberry Pi Pico with Pre-Soldered Header Microcontroller Mini Development Board Based on Raspberry Pi RP2040 Chip,Dual-Core ARM Cortex M0+ Processor, Flexible Clock Running up to 133 MHz* Sold by: Bicool-US (seller profile) Condition: New	$13.25
1 of: *4Pcs LCD1602 1602 LCD Module Blue Screen 16x2 Character LCD Display with IIC I2C Serial Interface Adapter Module* Sold by: Shutao (seller profile) Condition: New	$15.99
1 of: *Gateron ks-27 Low Profile Key Switches for Mechanical Gaming Keyboards | Plate Mounted (Gateron Red, 90 Pcs)* Sold by: Ranked llc (seller profile) Condition: New	$39.99

Shipping Address:
Andrew Imwalle

United States

Shipping Speed:
FREE Prime Delivery

Payment information

Payment Method:
MasterCard | Last digits:

Billing address
Andrew Imwalle
15 WOODED BROOK DR
THE WOODLANDS, TX 77382-1231
United States

Item(s) Subtotal:	$69.23
Shipping & Handling:	$0.00
Total before tax:	$69.23
Estimated tax to be collected:	$5.71
Grand Total:	**$74.94**

10/17

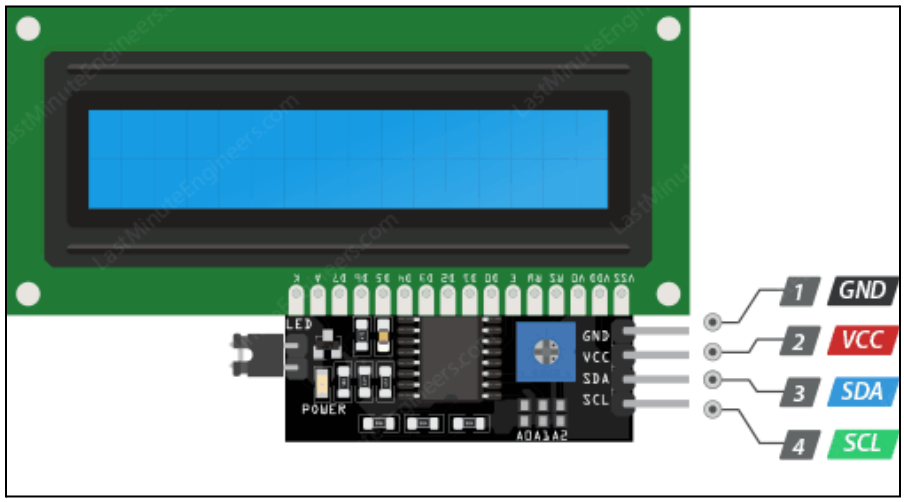
I researched how to send the keyboard key presses from the raspberry pi to the user's computer. Originally, we were planning to use a USB A => USB A cable to connect the Pi to the computer. However, I found that data cannot actually be sent this way, as the USB A ports on the Pi do not support data transmission. There needs to be a USB On-The-Go (OTG) port for key presses to be transferred to the computer. Luckily, the Pi has a USB OTG port built into the power port. This means that we only needed a USB A => USB C cable to send data.

Reference: Stack Exchange

Update on hardware: I put in the refund request and got the refund. Additionally, the parts were delivered late so we now have two sets of parts

10/19

I started researching how to run code on the Raspberry Pi Pico. We require the code to be able to output the text suggestions on the 1602 LCD displays. The displays take input via I2C, so I had to learn about how I2C works on the Pi Pico. We settled on using Micropython to run code on the Pico.



1602 LCD Display Pinout

10/23

The hardware team met separately on a Sunday. We did more research on executing code on the Pico. We realized that it was probably better to use the Pi instead of the Pico to process the computation and output to the LCD. The Pi offers more I2C channels and it would be better to not send data back to the Pico, once the suggestions are found.

10/26

I researched how to connect and send data from the Pi to the three LCD displays. I found some code on github that connects a singular LCD to the Pi using C++ and the WiringPi library. I eventually got this code working on the Pi but it only allows for one LCD and isn't modular. I need to create a class for LCD and allow it to initialize all three LCD's based on their address.

http://wiringpi.com/reference/i2c-library/

https://github.com/Szpillmann/LCD1602A

10/31

I created the LCD class, as well as an LCDList class. The constructor requires the I2C address of the LCD display. We ran into a problem here, since the I2C address is built into the hardware and we have 4 identical LCD displays, meaning each display had the same address. We require different addresses so that we can control them differently. I found that the address could actually be changed by bridging two pads located on the display. Andrew helped me solder the two pads together. This allows the displays to have different addresses, which is evident when I used the I2C scan tool on the pi.

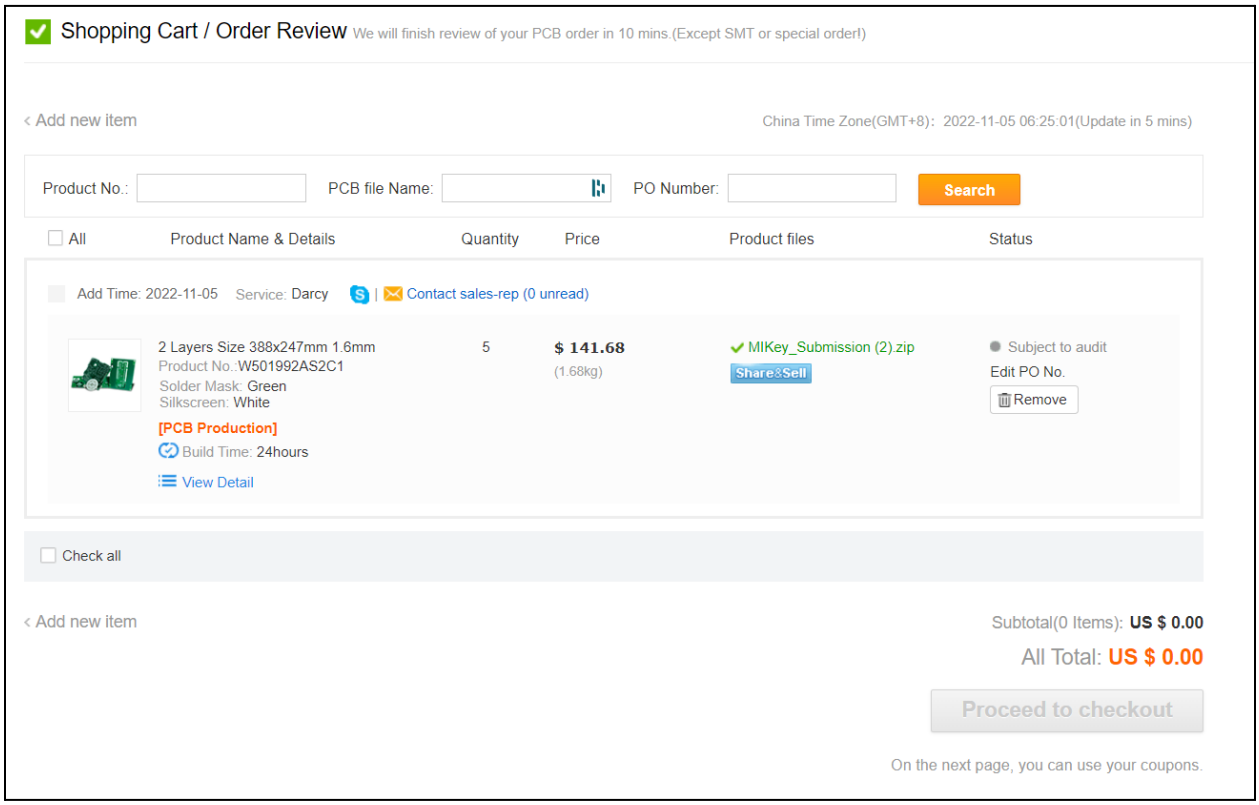
I created a LCDList class to contain all the LCD's and equally distribute commands across them. Our main driver file will have a string array of suggestions. When these suggestions need to be updated, the driver simply needs to call LCDlist.suggest(array) to update all the displays at once. The LCDList class handles any number of displays as well. This means that we can test with two displays or even 5, depending on how many suggestions we want to show to the user.

11/2

Integrated the LCD and LCDList classes into the driver program. We used I2C to interface with the LCDs. An I2C connection was used to send data to the LCDs. The alternate connection is SPI; however, for our use case, I2C was the best option. I2C connections only require two wires, serial clock (SCL) and serial data (SDA), to transmit data bi-directionally. Additionally, the address system and shared bus of I2C allows for multiple devices to be connected via the same wires. Compared to SPI, which uses four lines, I2C is simpler in complexity and requires less GPIO pins. The only downside of I2C is the speed of data transfer, but this is negligible since we are only transmitting three strings at a time.

11/7

We ordered the PCB from PCBway. We sent a schematic containing the circuitry we needed for the keyboard to work. There were slots for the Pico and the decoders that we used to get the key presses from the key matrix



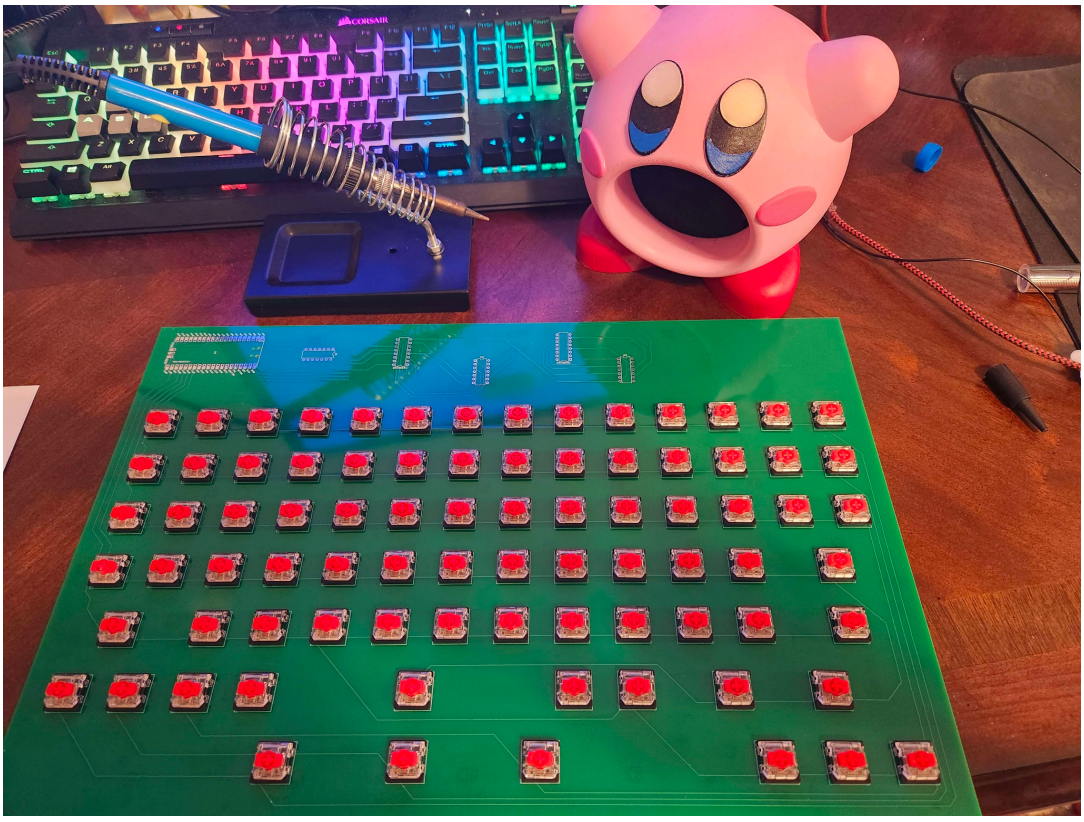
11/14

The PCB arrived and we began soldering the key switches onto the PCB. We connected the pico to the PCB and ran into some issues that we did not foresee.

PCB and PICO issues:

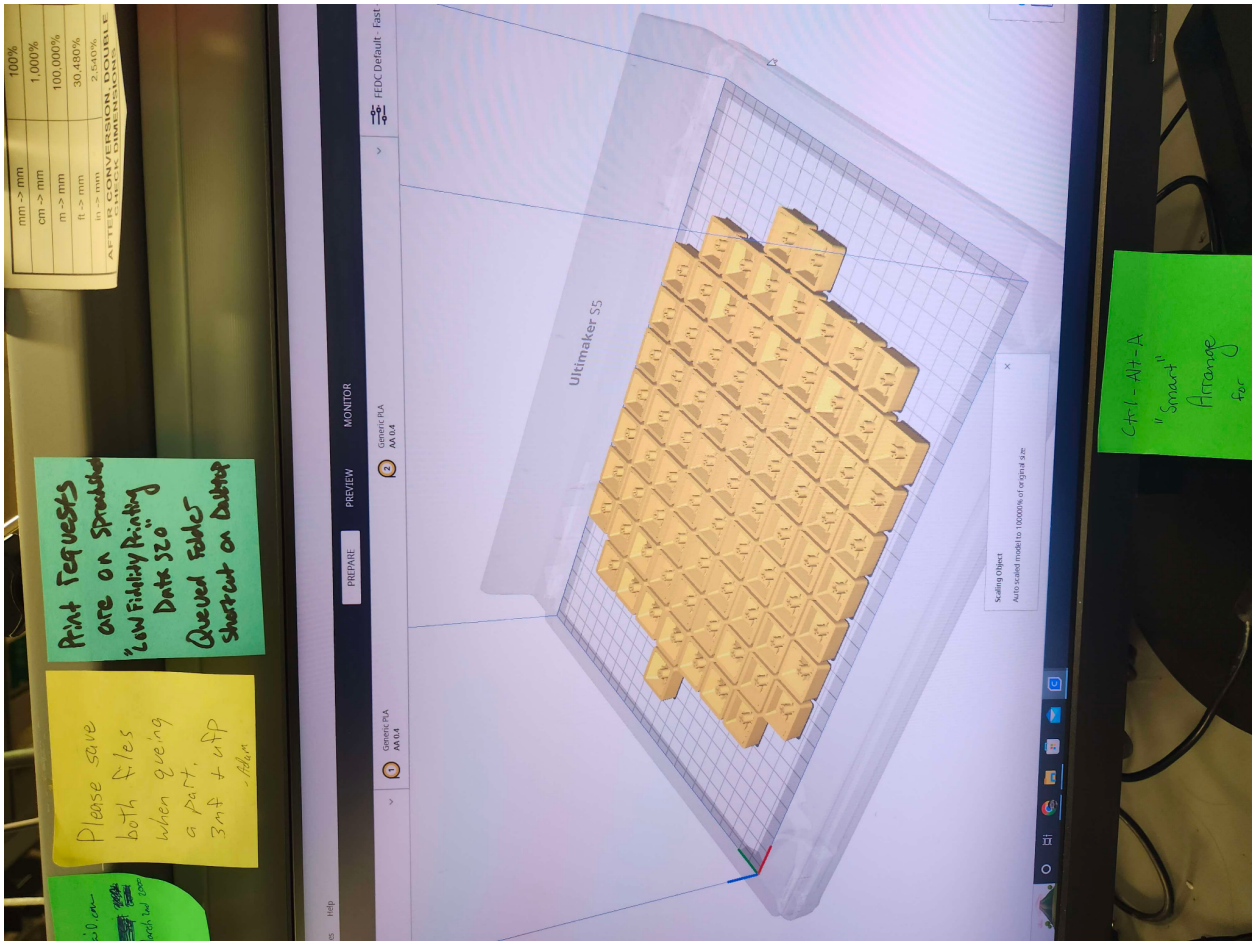
- PCB connects all unused pins to GND
- Unfortunately, power pins are also connected to GND
- PICO would not start up when soldered onto the board
- Plan right now is to only connect pins that are necessary onto the PCB

We removed the unnecessary pins from the Pico so no pins will be unintentionally powered or grounded.



11/16

80 keys were submitted for printing at the FEDC. These are the keycaps that will be placed on top of the keyswitches.



11/28

Did some user testing with the team to measure some data about the keyboard. We took a reaction time test that involved pressing a button when a light flashes. This was done to compare the latency of the keyboard against a real keyboard. We found that our keyboard has negligible latency difference when compared to a real keyboard.

Person	Jackson		Brittany		Andrew		Abhishek		Connie		Max	
	MIKey Times	Normal Times	MIKey Times	Normal Times	MIKey Times	Normal Times	MIKey Times	Normal Times	MIKey Times	Normal Times	MIKey Times	Normal Times
	401	281	347	301	254	286	258	264	329	274		
	322	315	421	271	291	276	284	238	279	285		
	303	325	313	290	261	273	245	246	304	338		
	276	346	325	331	261	316	258	300	284	247		
	285	296	344	319	263	247	257	245	303	284		
	311	297	360	283	288	265	260	250	291	287		
	336	318	357	302	268	272	256	227	298	247		
	306	271	380	273	230	276	282	245	434	266		
	321	318	343	305	277	267	256	264	300	268		
	284	279	332	268	266	263	265	243	304	256		
Average:	314.5	304.6	352.2	294.3	265.9	274.1	262.1	252.2	312.6	275.2		
Deviation	35.81200916	23.73558042	30.6478203	21.10845012	17.42571535	17.95333457	12.08718329	20.087586	44.76159068	26.56982415		
	MIKey First		Normal First		Normal First		MIKey First		Normal First		MIKey First	

MIKey average	288.775
Normal average	276.525

We also did testing with real world paragraphs to see how many keystrokes we could possibly save. We found that:

- We predicted 39/40 words
- We saved 30% of keystrokes compared to a regular keyboard

This shows that our keyboard can really be beneficial for users with motor impairments.

11/30

Started working and gathering information for the final presentation and report

Things I want to include in the report:

- LCD Hardware
- LCD Software
- Benefits of using I2C for the LCD
    - Less wires
    - Less GPIO pins used
    - Ability to send data in parallel
- Proposed Budget
- Final Costs
- Why they were different
- Manufactured unit cost (assuming we made 1000 units in bulk)

11/30

Final Cost

Product Name	Cost (dollars)	Purchase Site
Raspberry Pi Pico	20.82	Amazon
LCD Displays	23.79	Amazon
Keyboard Switches	43.29	Amazon
PCB Material	43.16	PCBWay
Total	131.06	

Proposed Cost

Product Name	Cost	Purchase Site
Raspberry Pi 0	79.99	Amazon
LCD Displays	15.99	Amazon
Keyboard Switches	47.98	Amazon
O-rings	6.99	Amazon
PLA Filament	18.99	Amazon
PCB Material	~90	pcbway.com
Total	259.94	

The biggest difference between the final overall costs and the proposal occurred for a variety of reasons. We realized that the Raspberry Pi 0 was not needed and could be replaced by the cheaper Raspberry Pi Pico. We used an existing Raspberry Pi 4 to handle intense computations and the Pico to only feed keystrokes to the Pi. Next, after receiving the keyboard switches, we realized that we did not need O-rings, since the low-profile switches we received already reduced the actuation distance. Additionally, we realized that we could print the keycaps and the housing for free at the FEDC.