Large Than Life Piano Remake

Final DRAFT

Created by Will Ebmeyer
2022-2023

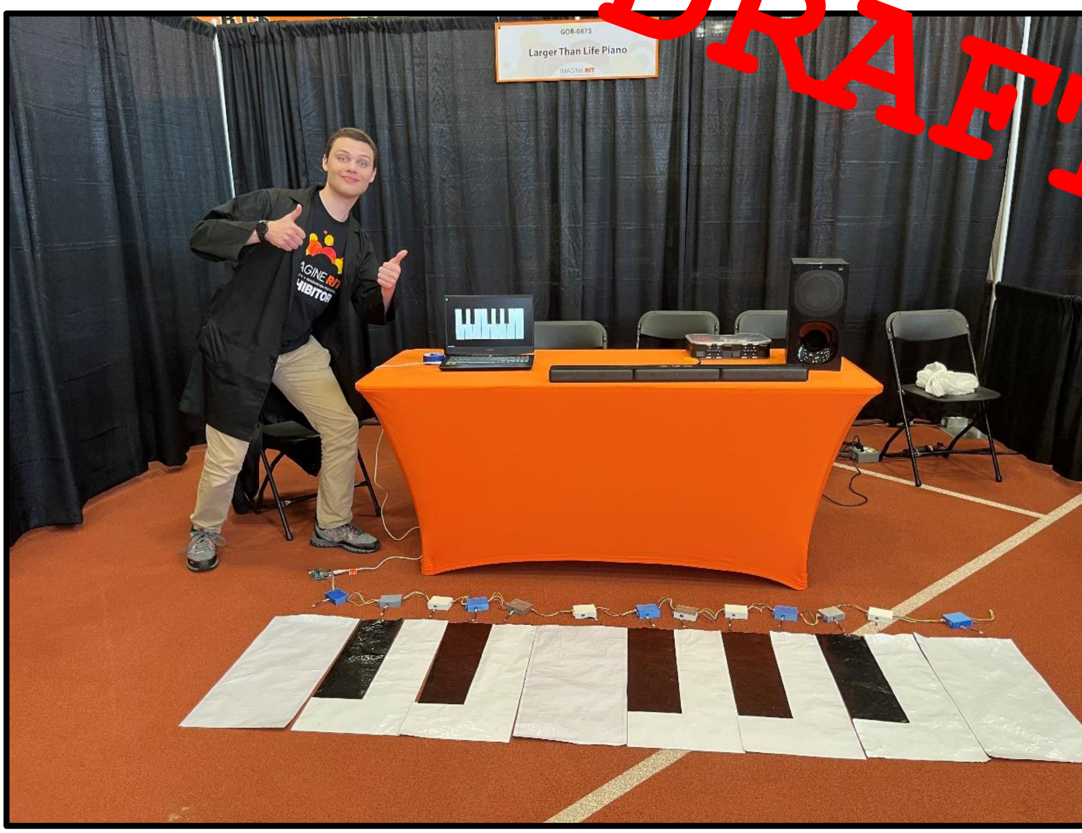


Table of Contents

Note: GitHub repository available here: https://github.com/Sciguy324/Big-Piano-2

Introduction

Larger than Life Piano is a recurring project that's gone on for several years now. It's taken on a handful of forms over the years:

Piano from (probably) Imagine 2019.

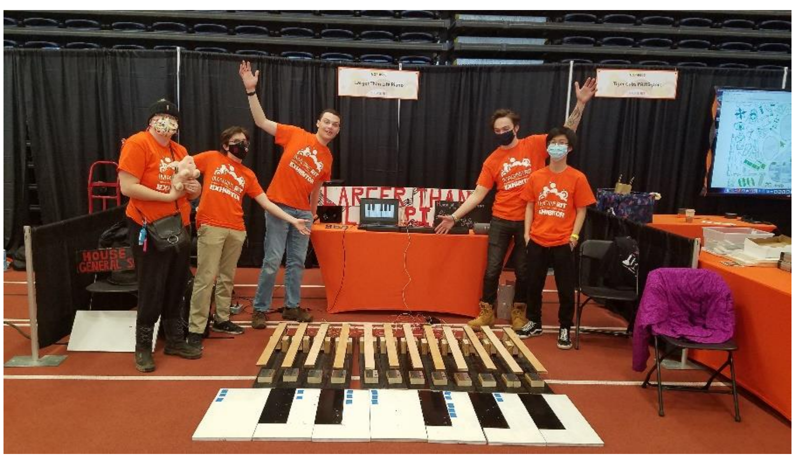
Though, I've got records of piano existing as far back as 2017. Sadly, I couldn't find any other photos ☹

I wasn't there to see it in action, but *I think* this version used capacitive touch sensors to detect key presses.

After suffering three years in the Gosnell storage room, old piano wasn't usable anymore. So, we rebuilt it in time for Maker Faire 2021.

Oh, we didn't have it working till the night before Maker Faire. Fun times.

This version is just a bunch of large mechanical switches that complete a circuit when you press on them. It then uses rubber bands to spring the key back to its resting position.



That's me! (Third on the left)

With the last piano barely surviving Maker Faire, I standardized and rebuilt the design for Imagine 2022.

We ended up showing it off a couple more times, too, including Maker Faire 2022.

The great thing about piano, above all else, is that it's *interactive*. Who *wouldn't* want to jump on a giant floor piano? So, if you're here to redesign piano once again, or are thinking of starting your own project, remember this: *interactive projects are fun* 😊. Inclusion of the emoji is mandatory.

I swear, if someone mentions *Big* one more time, I'm gonna commit spaghetti

Of course, piano isn't without its problems. First off: everyone's afraid of breaking it. It hasn't happened yet, but these mechanical switches are still just *wood*. Ideally, it would be cool to just jump or run across the piano without it breaking.

Secondly, this thing is a *pain* to assemble:

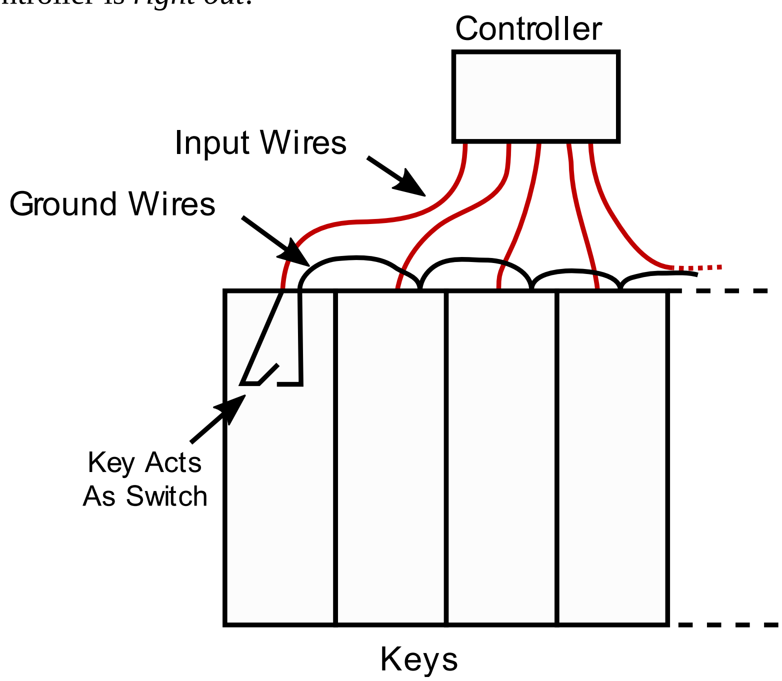
- Each key must be connected to the next key via a "common ground" wire.
  - Except for the last key, as there is no "next" key to connect. Instead, it must be connected to the controller.
- Each key must have an individual wire connected to the controller.
- Connecting a wire is as simple as:
  - Completely unscrew the corresponding screw on the terminal block.
  - Place the wire against the hole.
    - Hopefully the wire already has a crimp connector attached.
    - Otherwise, *God help you*.
  - Re-screw the screw to secure the wire in place.
  - Repeat 36 more times
- The rubber bands wear out over time, so it's important to replace them every now and again
- So does the tin foil.

So, in this revision, we're gonna focus on two main goals:

- Make piano keys entirely flat and highly portable, thus bolstering their durability.
- Make assembly fool-proof. We're talking quick-connect wires that just plug in without hassle—no knowledge of what each part does needed!
  - This means each key needs to be *modular*: it doesn't matter what order the user plugs stuff in, *it just works*.

Modular Design

Modularity is one of the central goals of this redesign. As such, the old method of wiring each key to the controller is *right out*:

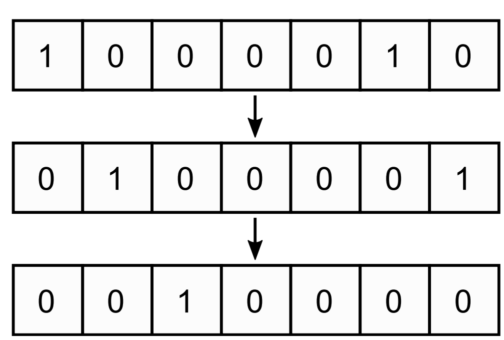


Old method of wiring keys to the controller.
This was an absolute nightmare when we had to set it up for the club/activity fair and had to recut all the lost wires on the spot.

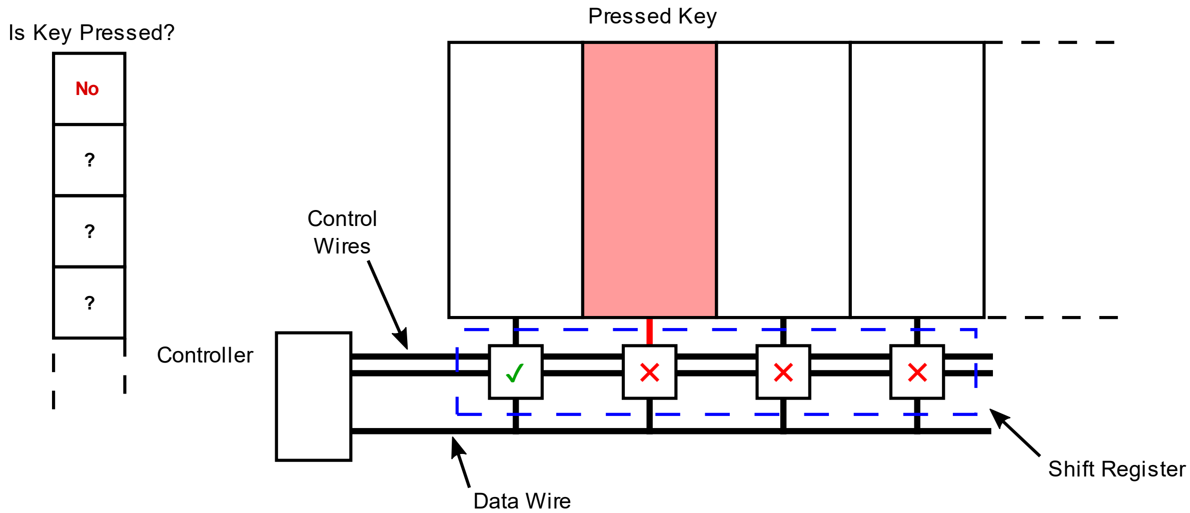
So, this redesign will consist of repeated, identical modules and a fixed number of wires leading back to the controller. Adding a key will then be as simple as plugging in another module at the end of the line.

Collecting Key States

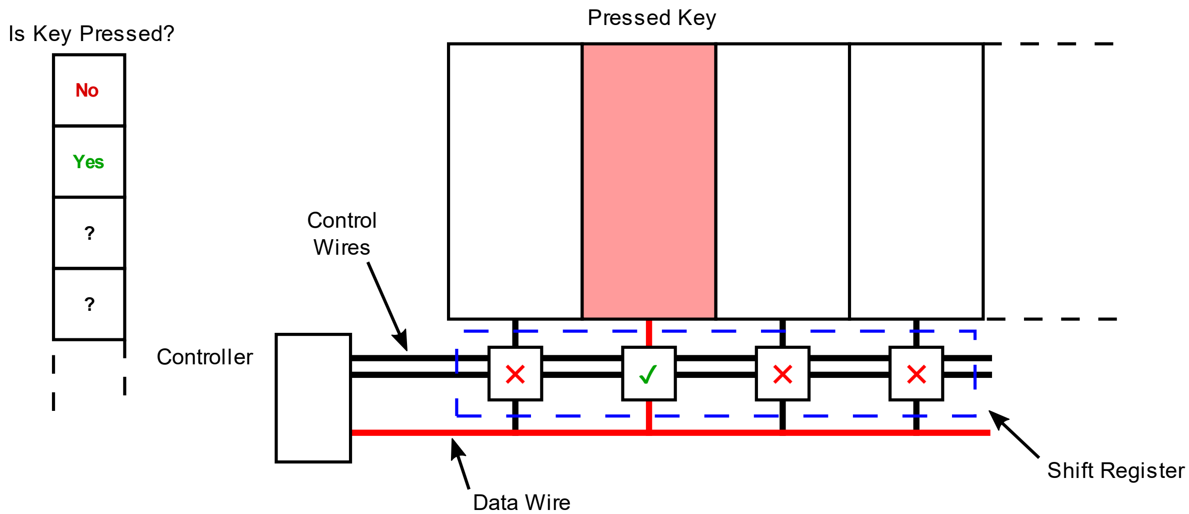
To accomplish this modularity, we will use a *synchronous shift register*. For those unfamiliar, shift registers are digital logic devices that send data down a line, one bit at a time (see figure on the right).



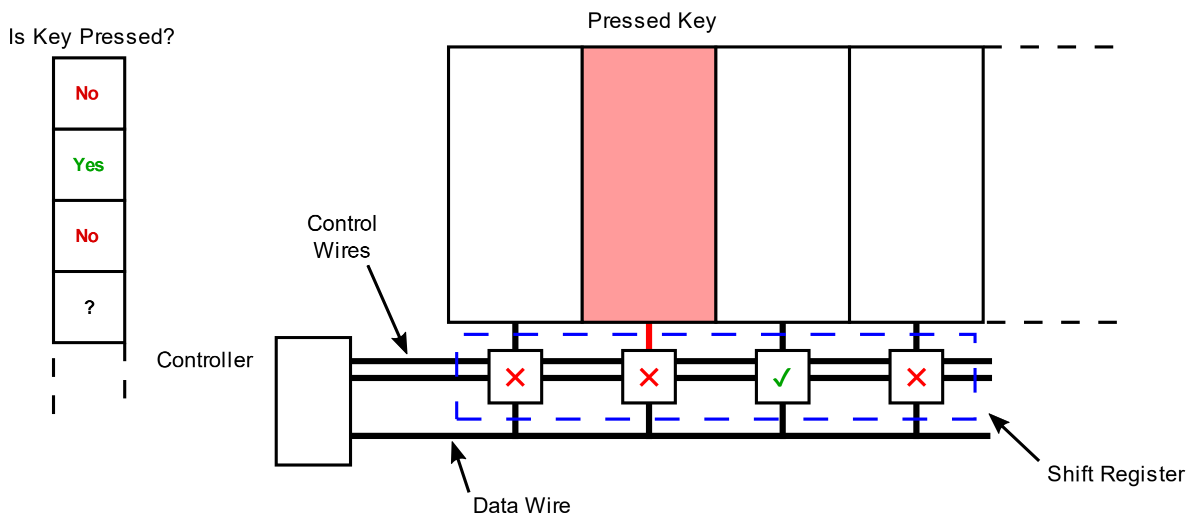
As shown in the following diagrams, this shift register will be used to select which key we wish to check the state of.



• Using a shift register, the controller sends a “get state” signal down the line. This signal connects the corresponding key to the data line.
• In the above case, the “get state” signal reaches the first key. Since the first key isn’t pressed, it leaves the data line unpowered. The controller detects this and registers the key as “not pressed.”



• The “get state” signal advances down the shift register.
• This time, the corresponding key *is* pressed, which powers the data line. Detecting this, the controller registers the key as pressed.



• This process of collecting one key state at a time repeats until all keys are accounted for.

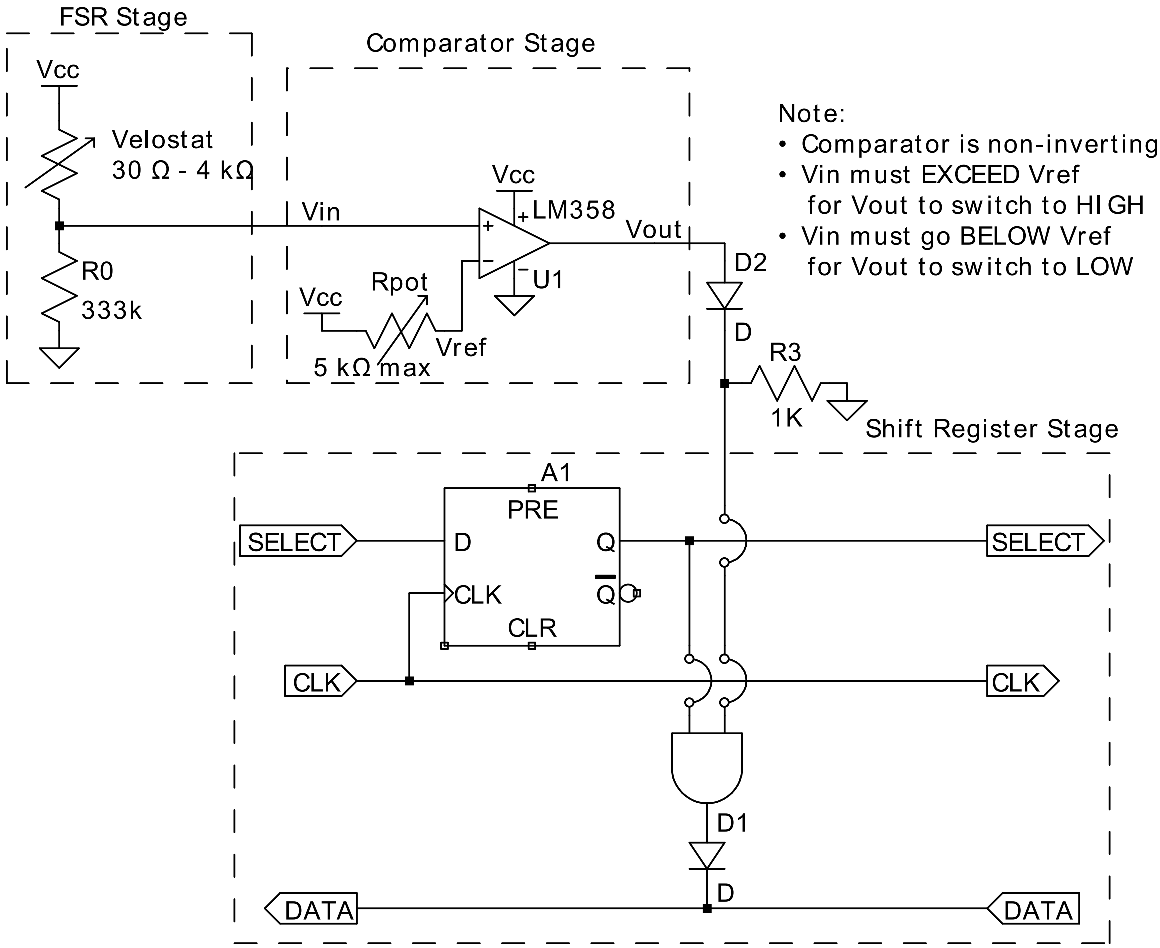
Individual Keys Circuitry

The plan is to use velostat as a force-sensitive resistor (FSR). As the user presses down on the key, the resistance of velostat will decrease, resulting in less of a voltage drop. Since velostat is just a flat (bendable!) sheet, this accomplishes our goal of having flat, portable keys.

So, using a comparator circuit, we can convert this analog voltage into a “pressed” or “not pressed” digital signal. This comparator will include a potentiometer, for tuning how much pressure is required for the key to be considered “pressed” versus “not pressed.”

Finally, this comparator leads into the shift register stage. If the shift register is currently selecting this key, the “pressed” signal should be allowed to pass into the data line. Otherwise, the signal *must* be blocked from the data line—otherwise it will start overlapping with other signals.

Altogether, the basic wiring for each key module will look something like this:



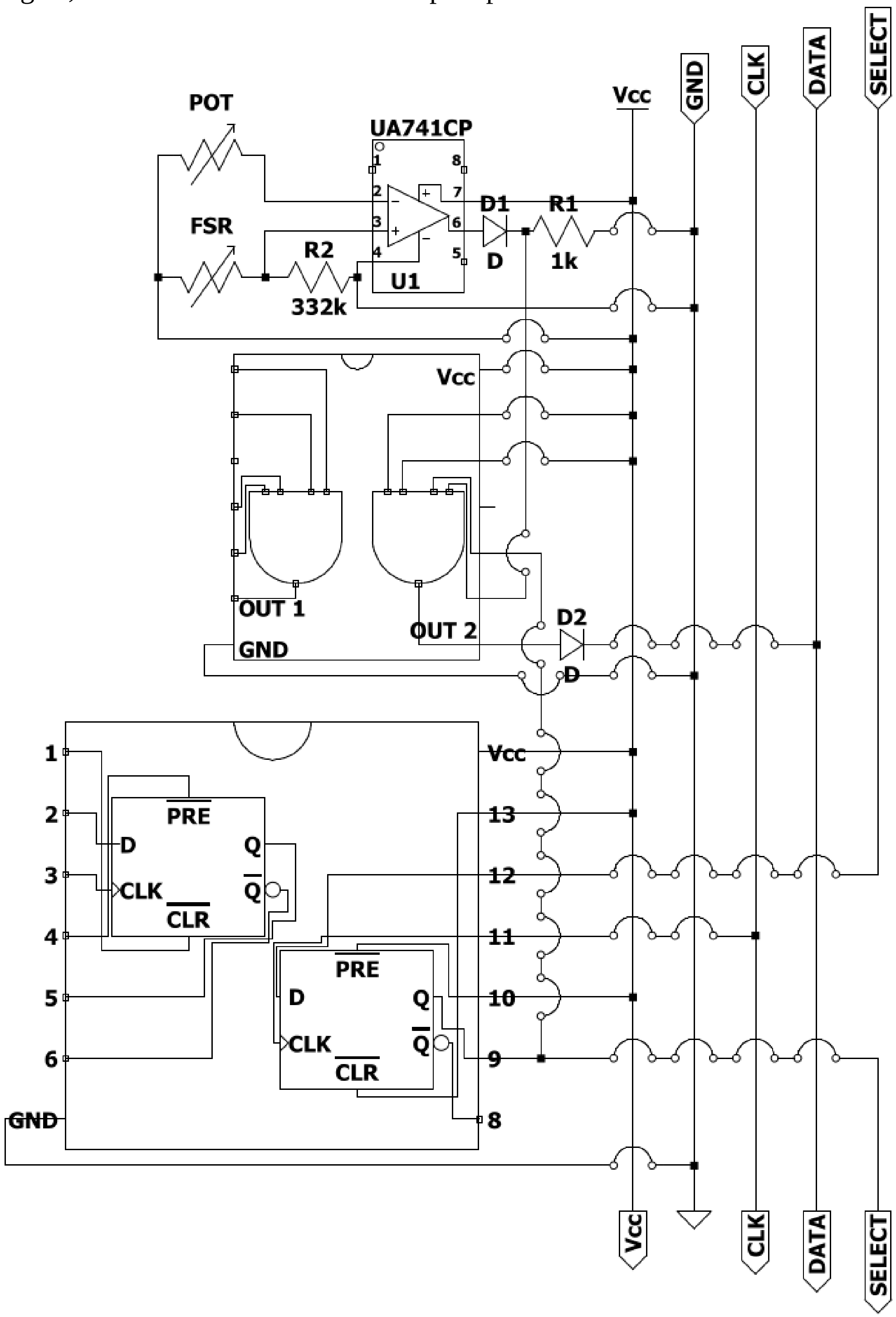
FSR Stage: Electrical resistance decreases as you press on the key. This means the output voltage will change depending on how hard you're pressing down.

Comparator Stage: Uses an op-amp to compare the voltage from the key to a potentiometer (variable resistor). If the resistance across the key is LESS than the potentiometer, the chip will output a HIGH signal.

- You can adjust the potentiometer to control the sensitivity of the key.

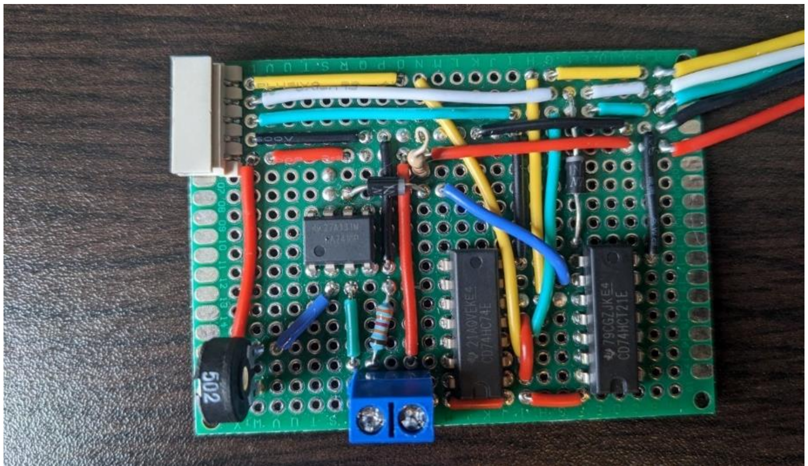
Shift Register Stage: Uses a D-flip-flop, an AND gate, and a diode to determine when this key should be connected to the DATA line.

Physically, this will be implemented using a UA741CP op-amp, a CD74HCT21E dual 4-input AND gate, and a CD74HC74E dual D-flip-flop:



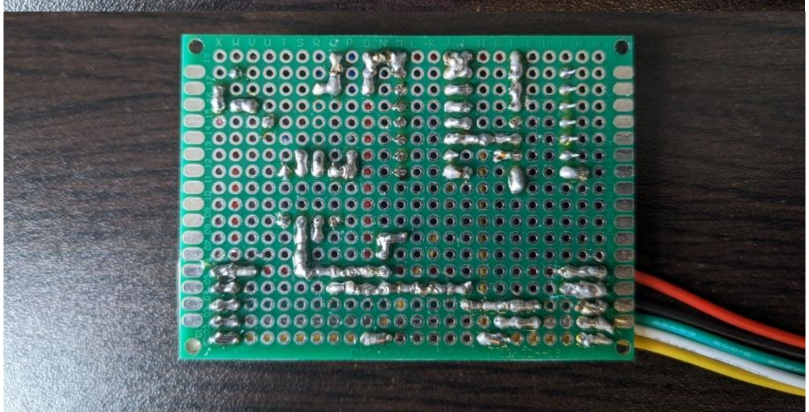
Assembled Boards

Here are images of the assembled boards.

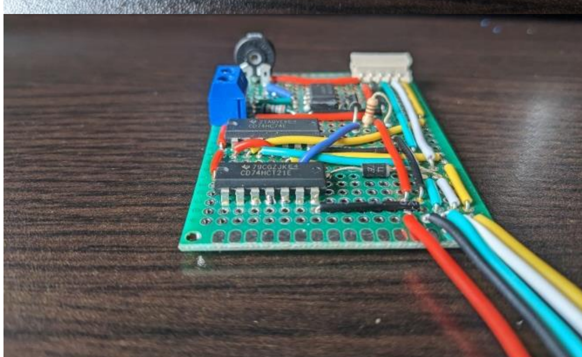


Note: You see that connection port on the left-side? I should've put that on the right side too, instead of just soldering the wires to the board.
Unfortunately, I got lazy.
Ah well.

Also: the AND gate and D-flip-flop are swapped from the previous diagram.



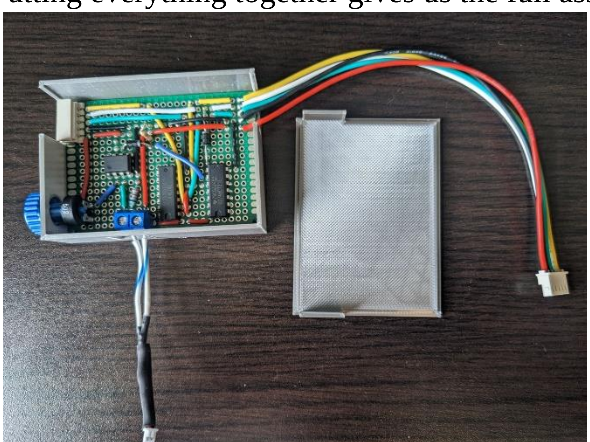
There's not enough space on the top on the top to connect everything, so here's the connections on the bottom.

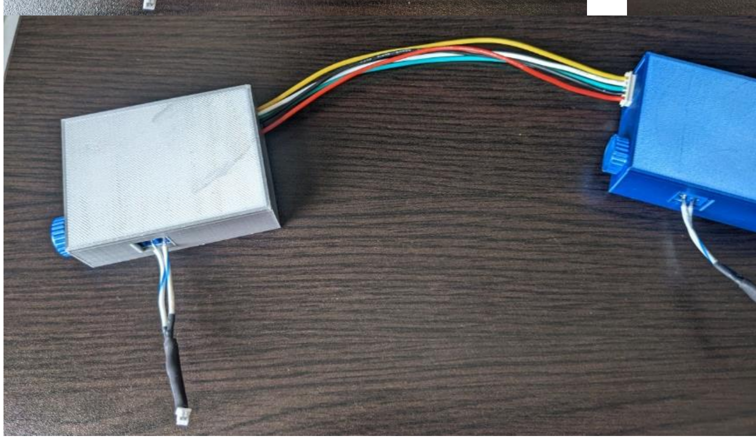


A few components that compliment the board, including the mini JST quick-connect wire, the sensitivity knob, and the protective casing (survived toddlers, 10/10 would case again)

3D-printable models can be found in the repository.

Putting everything together gives us the full assembly for the boards:





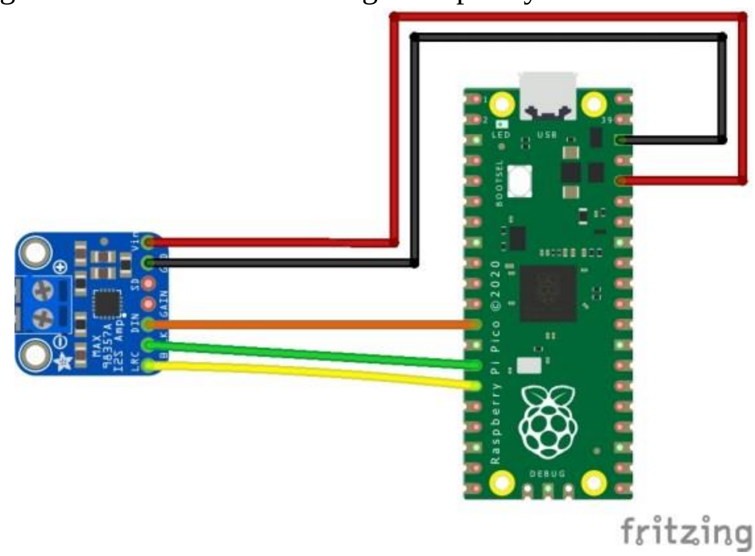
Connecting to the other boards is as simple as plugging the cables in to the next module.

These are fully modular, so you can swap boards if needed.

Controller

The old piano design relied on a connected computer to play notes. However, at one point there was a plan to interface the piano with a high-voltage system: “thundertune.” Briefly, this project played music using a spark gap.

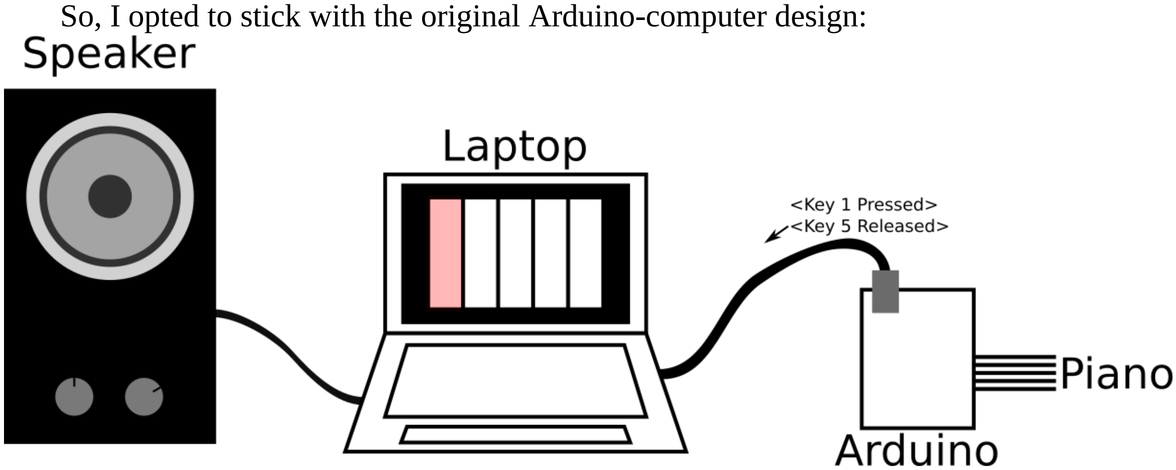
Since I’d like to keep my laptop far away from anything high-voltage, I had originally planned on building a dedicated controller using a Raspberry Pi Pico and audio amplifier:



Credit: https://www.recantha.co.uk/blog/?p=20950

Unfortunately, this plan fell through after thundertune was rejected by Imagine. Additionally, some testing found that the Pico played rather garbage audio. Even then, good luck storing more than a couple of notes with its limited capacity.

So, I opted to stick with the original Arduino-computer design:

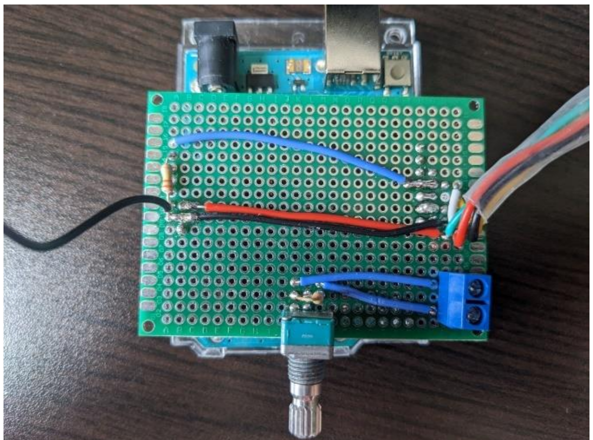


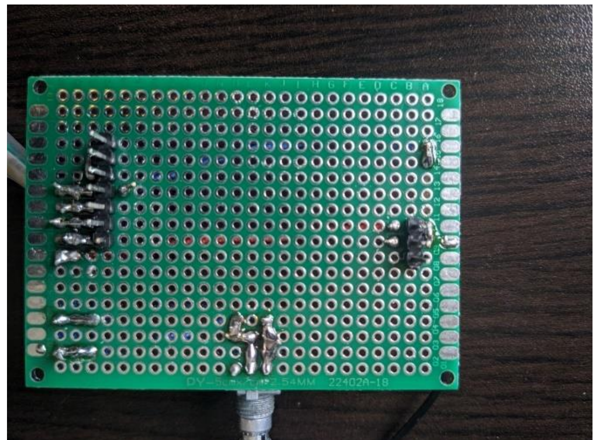
- Arduino sends key press/release signals over serial. (Ex: <Key 1 Pressed>)
- Laptop interprets events to play or stop notes.
- Speaker allows you to hear the piano in a crowded room. BRING A SPEAKER, YOU WON’T HEAR IT OTHERWISE!

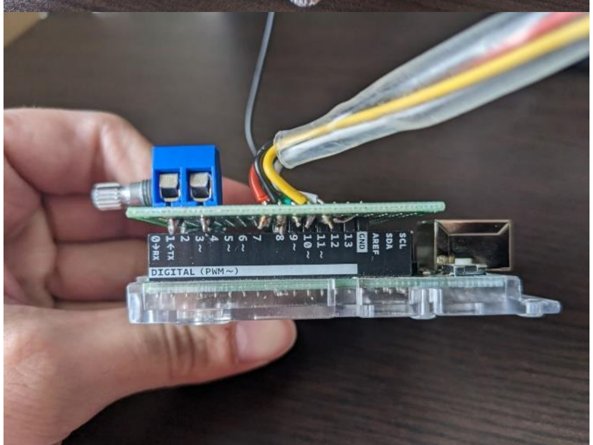
Assembled Controller

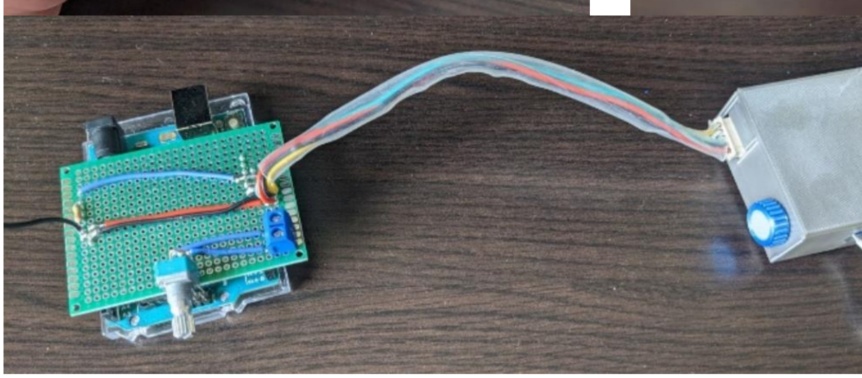
This board was originally meant for testing. Indeed, it still includes a loose ground wire for attaching a voltmeter, as well as a potentiometer for simulating a key. However, it ended up being the final board I went with for Imagine.

As seen below, the board acts as a shield that sits on top of the Arduino:









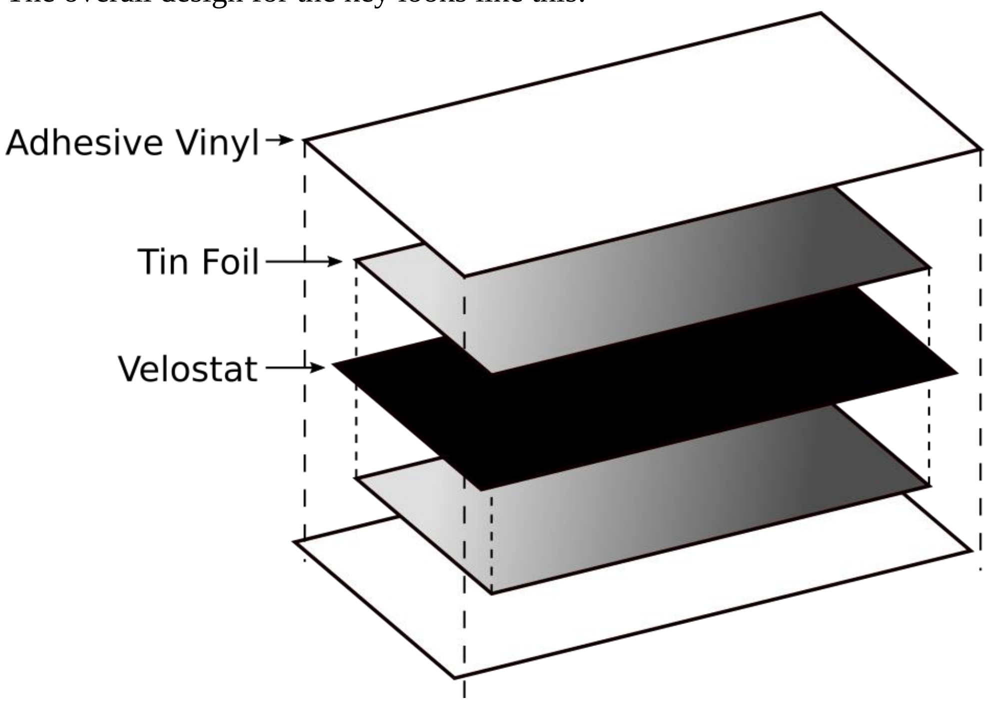
Note: the **DATA** line **must be connected to ground with a resistor**. This pulls its voltage down whenever it's unpowered. I used a **330 kΩ** resistor.

Piano Keys

As mentioned earlier, the keys use velostat as a force-sensitive resistor. Of course, to actually *use* that velostat we'll need three things:

- A metal contact on both sides of the velostat.
  - Must be lightweight to avoid putting pressure on the velostat
  - I used **aluminum foil**.
- A protective cover to protect the foil and velostat, and to hold everything together.
  - Must be lightweight.
  - Must be durable.
  - I used **adhesive vinyl**.
- A connection to the board

The overall design for the key looks like this:



Here, the tin foil is stuck to the adhesive vinyl. Then, the velostat is cut slightly larger than the foil to prevent the two sheets from touching. If the two foil sheets touch, the key will short out and effectively be useless. Finally, the two foil-vinyl halves are stuck together with the vinyl in-between them. For exact dimensions, see the appendix.

Although imperfect, here's the process I used for fabricating the keys:

1. Cut out *two* pieces of vinyl with the dimensions of the keys: one for the top, and one for the bottom.
2. The vinyl will want to curl, so use some tape to hold it down:

3. Carefully peel up the non-stick backing and place a strip of the backing that's about 1.5 cm wide. Do this on all but one side—you'll need one side to place the wires into later.
    a. If you do not have any strips of non-stick backing for this, you'll need to carefully cut some out from your vinyl.
    b. To avoid getting dirt on the sticky side of the vinyl, keep the rest of the non-stick backing on the vinyl until you're ready for the next step.
4. Remove the large sheet of non-stick backing (not the strips: leave those where they are). Take a roll of tin foil, line it up, and roll it across the adhesive. This step is very prone to defects, so do be careful.
5. Usually, you can just tear off the excess foil—anything that's already stuck to the adhesive will stay behind. If needed, use an exact-o knife to CAREFULLY remove any remaining foil without cutting the vinyl.
6. Apply this process to both sheets of vinyl you cut out.
7. Cut the velostat down to size. Exact dimensions are in the appendix, but I found that a piece of clear scotch tape was about the size needed. So, place a strip of tape on one side of the velostat, and use that to cut off one edge.
8. Remove the guide strips from ONE of the vinyl sheets, and carefully place the velostat on the adhesive.
    a. If all goes well, the velostat should COMPLETELY COVER the foil. If not, you've done goofed.
    b. The edges of the velostat should partially (not fully) overlap with the adhesive vinyl.
    c. Make sure there is ONE side where the velostat fully covers the adhesive. That way, when you stick the keys together, you can still access the inside.
9. Place the guide strips back on the vinyl sheet.
10. Place the two assemblies together, making sure to line them up. Carefully take the guide strips on both sheets and remove them, allowing the two halves to stick together.
11. Repeat for all remaining keys.

Assembled Keys

For the connector, I soldered a mini JST connector to some plain stranded wire. Then, I stuck the leads to the foil with some electrical tape.

Required Materials

Item Name	Unit Price	Amount	Total Price
D Flip Flop (CD74HC74)	$ 0.59	14	$ 8.26
AND Gate (CD74HCT21)	$ 0.43	14	$ 6.02
1N4002 Diodes	$ 0.09	28	$ 2.52
~~1 kΩ Resistors *(Used Construct's instead)*~~ ~~*(Need 15, sold as 100-pack)*~~	$ 0.0344	100	$ 3.44
~~47 kΩ Resistors *(Unused)*~~ ~~*(Need 15, sold as 100-pack)*~~	$ 0.0344	100	$ 3.44
332 kΩ Resistors	$ 0.0920	15	$ 1.38
5x7 cm Perfboards *(Need 13, sold as 20-pack)*	-	13	$ 9.99
Velostat Sheet	$ 4.95	16	$ 79.20
Op-Amps (UA741CP)	$ 0.50	14	$ 7.00
Potentiometers	$ 0.65	14	$ 9.10
~~JST 3-Pin Connector Pairs x13 *(Unused)*~~	$ 9.99	1	$ 9.99
~~JST 2-Pin Connector Pairs x13 *(Unused)*~~	$ 7.99	1	$ 7.99
2-Pin Screw Terminal Block	$ 0.00	13	$ 0.00
Molex Crimp Connectors (0008701039)	$ 0.073	85	$ 6.21
Molex Right-Angle 5-Pos Header (022057055)	$ 0.297	15	$ 4.46
Molex 5-Pos Conn. Receptacle (0050375053)	$ 0.149	15	$ 2.24
~~Raspberry Pi Pico *(Unused)*~~	$ 4.00	1	$ 4.00
~~Audio Amplifier Module *(Unused)*~~	$ 3.95	1	$ 3.95
Speaker	$ 0.00	1	$ 0.00
~~Audio Jack~~	$ 0.00	1	$ 0.00
Audio Cable	$ 0.00	1	$ 0.00
~~Wall to Micro-USB Cable~~	$ 0.00	1	$ 0.00
White Adhesive Vinyl (1 ft x 22 ft)	$ 15.99	1	$ 15.99
Black Adhesive Vinyl (1 ft x 11 ft)	$ 7.99	1	$ 7.99

Aluminum Foil	$ 0.00	1	$ 0.00
22 AWG Solid Core Wire	$ 0.00	1	$ 0.00
22 AWG Stranded Wire (Multiple Colors)	$ 16.95	1	$ 16.95
Arduino	$ 0.00	1	$ 0.00
USB B to USB A cable	$ 0.00	1	$ 0.00
		Subtotal	$ 210.12

Note: items marked as $0.00 are assumed to already be available. Items that are crossed out were never used in the final design.

Future Recommendations

Save Yourself the Pain

For the love of God, **don't assemble the boards from scratch**. That's thirty-five hours of soldering in the construct I'm not getting back. Fun fact: there are companies that make custom circuit boards for real cheap. ***USE THEM!!1!!11!*** Alternatively, Raspberry Pi Pico's sell for about $4. So, if you have a way to power and coordinate thirteen of them, you could use their analog pin to read each key.

I pity the poor soul who fails to heed this advice.

I Don't Know How to Desolder

On *five* separate occasions, I had an op-amp fail on me for reasons unknown. I was able to fix three boards by cutting all connections to the broken chip and rerouting them to these handy dandy "IC sockets."

In the future, I'd recommend using these sockets wherever you have chip. That way, you can just replace a chip without hassle.

Case in point: when the fourth chip failed during Imagine, it happened to be on a board that used one of these sockets. So, I was able to quickly replace it with a spare.

These Op-Amps are SUS ඞ

I have no idea why, nor do I have the electrical expertise to know why, but those UA741 op-amps kept failing on me. I had FIVE separate chips fail on me across FOUR separate boards. Maybe I was driving them at a voltage they weren't designed for? Maybe

Apparently, these particular chips are kinda obsolete. Like, forty-five years obsolete. You should probably investigate more modern alternatives. For example: the LM358.

Hold on, didn't my circuit diagram originally call for an LM358? DAMN IT!

The Real Key Was the Friends We Made Along the Way

Overall, the piano keys performed GREAT at Imagine—holding up against the greatest stress-test known the mankind: toddlers. Not only are they durable, but you can *roll them up*; I literally carried the entire piano to and from Imagine in a single bag! So, I'd highly recommend sticking with this same or similar design for the keys.

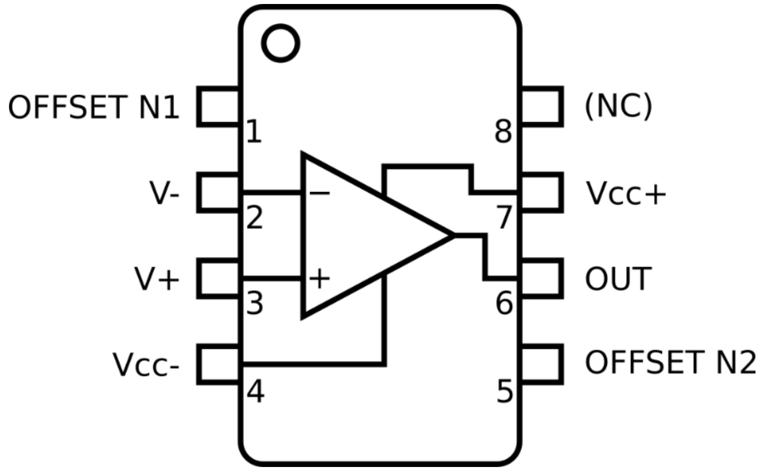
However, they're not without drawbacks. Just to name a few:

- The vinyl slips around easily, so we had to tape the keys down. Maybe use something with a bit more friction?
- The velostat tends to drift a bit, forcing me to occasionally readjust their sensitivity.
- The aluminum foil is impossible to apply perfectly flat, leading to visible defects in the keys. Perhaps try using strips of foil instead of one big sheet in the future?
- Sometimes a key would be stomped on in *just* the right way to form a pressure point. Considering the key senses pressure, this is quite the problem. During Imagine, I had to tear a key apart and put it back together again (twice) to remove such a pressure point.

I also had a member recommend fabricating the keys as one singular giant mat, instead of thirteen tiny mats. I'll leave this one up to you. The rest are *not* up to you. I will find you.

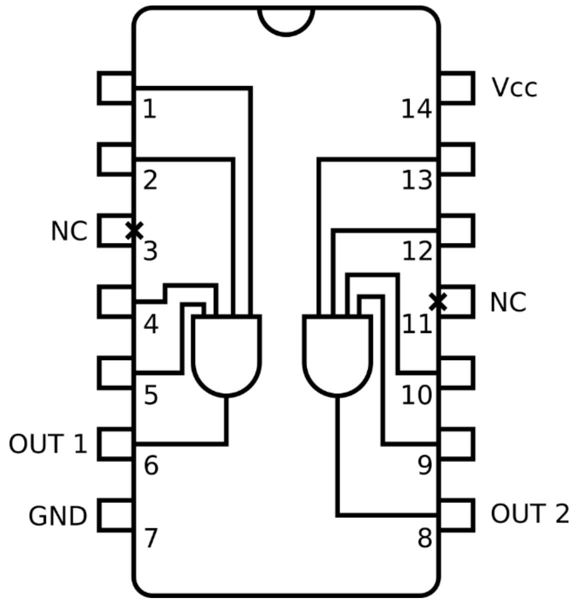
Component Pinouts

UA741CP Op-Amp:



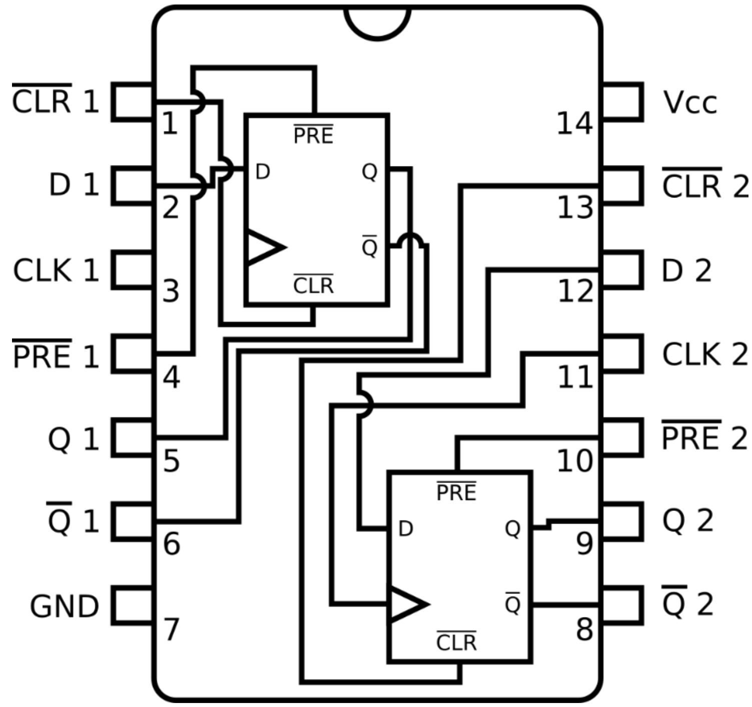
Pin		Description
Name	**Number**	
V-	2	Noninverting input
V+	3	Inverting input
NC	8	(No connection)
OFFSET N1	1	External input offset voltage
OFFSET N2	5	External input offset voltage
OUT	6	Output
Vcc-	4	Positive voltage supply
Vcc+	7	Negative voltage supply

CD74HCT21 AND Gate:



Pin		Description
Name	**Number**	
Gate 1 Inputs	1, 2, 4, 5	First AND gate inputs
Gate 1 Output	6	First AND gate output
Gate 2 Inputs	9, 10, 12, 13	Second AND gate inputs
Gate 2 Output	8	Second AND gate output
NC	3	(No connection)
NC	11	(No connection)
Vcc	7	Voltage source
GND	14	Ground

CD74HC74 D Flip-Flop:



Pin		**Description**
Name	**Number**	
$\overline{CLK}$ 1	1	Clear input (active low)
D 1	2	Data input
CLK 1	3	Clock input (positive edge triggered)
$\overline{PRE}$ 1	4	Preset input (active low)
Q 1	5	Output
$\bar{Q}$ 1	6	Output (inverted)
CLR 2	13	Clear input (active low)
D 2	12	Data input
$\overline{CLK}$ 2	11	Clock input (positive edge triggered)
$\overline{PRE}$ 2	10	Preset input (active low)
Q 2	9	Output
$\bar{Q}$ 2	8	Output (inverted)
GND	7	
Vcc	14	

General Dimensions

The following pages contain dimensions for the full piano and its components.