Large Than Life Piano Remake

DRAFT 2½

Created by Will Ebmeyer
2022-2023

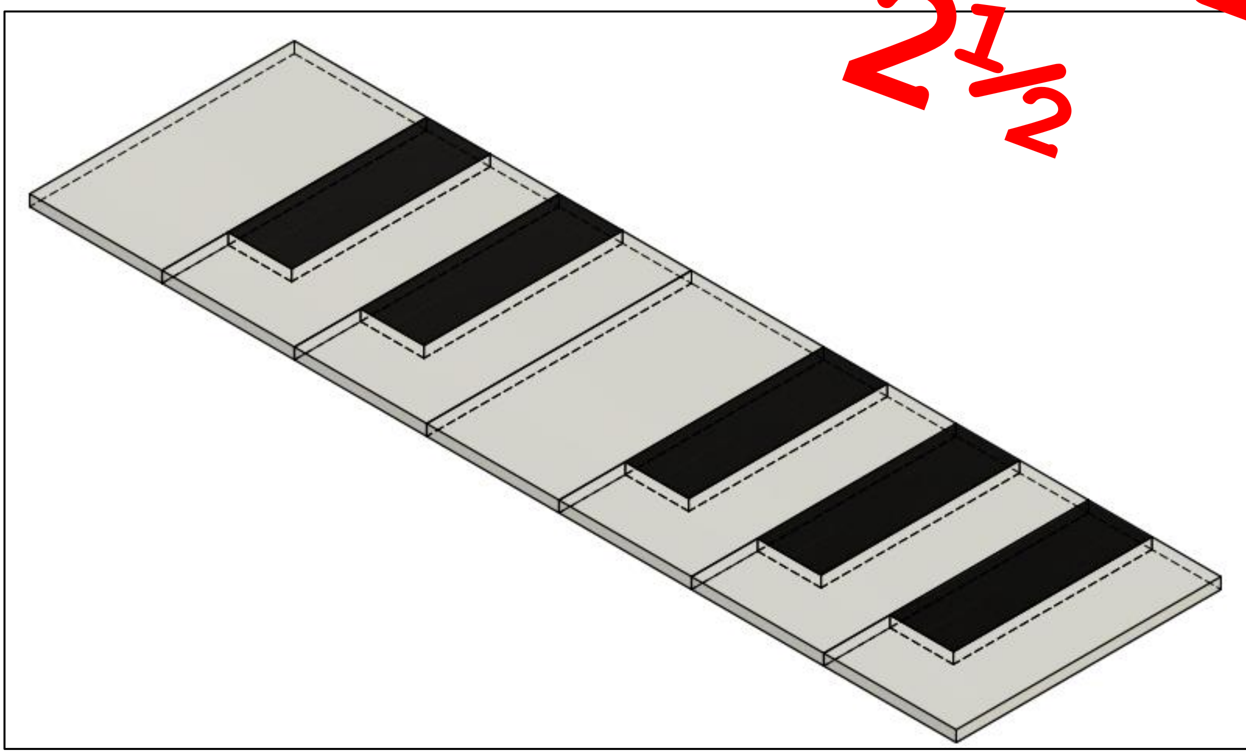


Table of Contents

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:

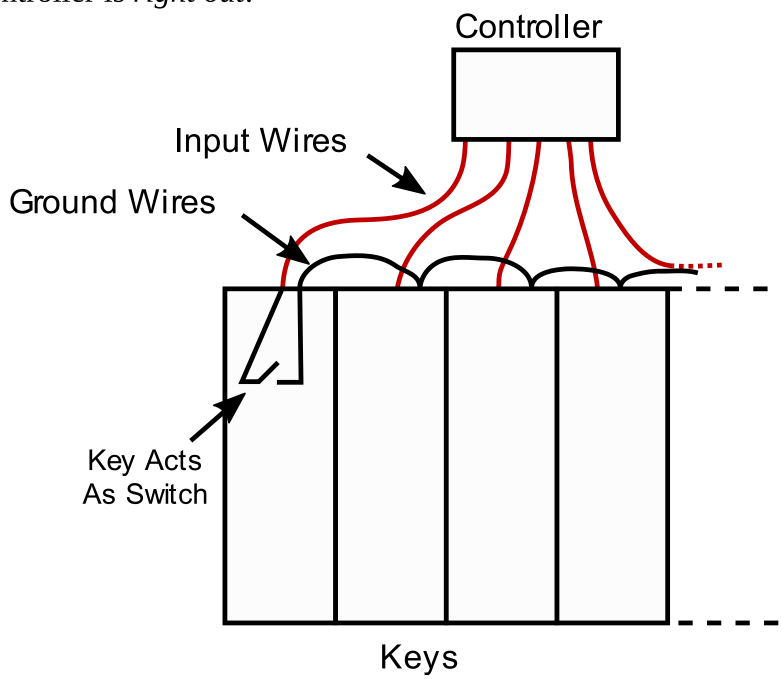
- Each key must be connected to the next key via a "common ground" wire.
    - Except for the last key, as there is no "next" key to connect. Instead, it must be connected to the controller.
- Each key must have an individual wire connected to the controller.
- Connecting a wire is as simple as:
    - 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 anyways*.

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*:

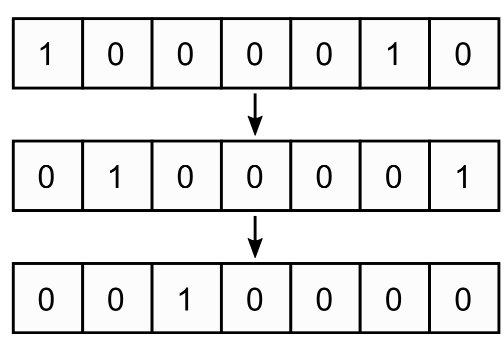


Old method of wiring keys to the controller.
This was an absolute nightmare when we had to set it up for the club/activity fair and had to recut all the lost wires on the spot.

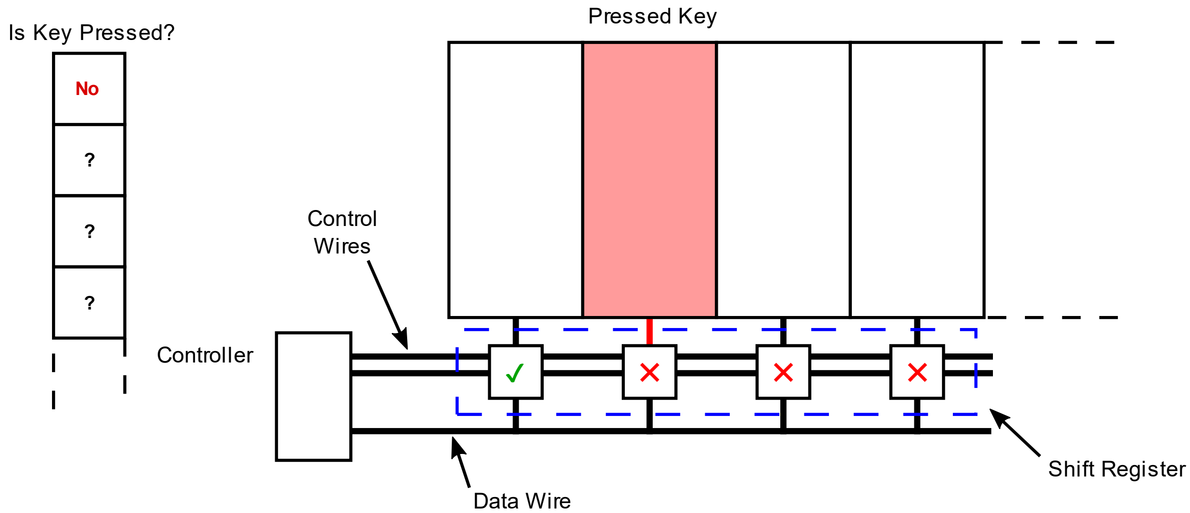
So, this redesign will consist of repeated, identical modules and a fixed number of wires leading back to the controller. Adding a key will then be as simple as plugging in another module at the end of the line.

Collecting Key States:

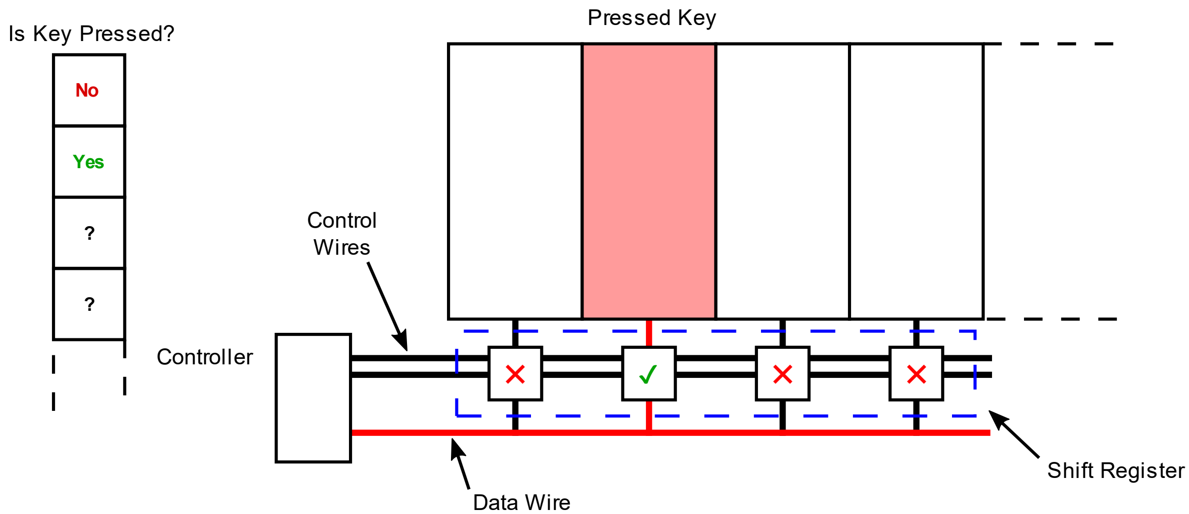
To accomplish this modularity, we will use a *synchronous shift register*. For those unfamiliar, shift registers are digital logic devices that send data down a line, one bit at a time (see figure on the right).



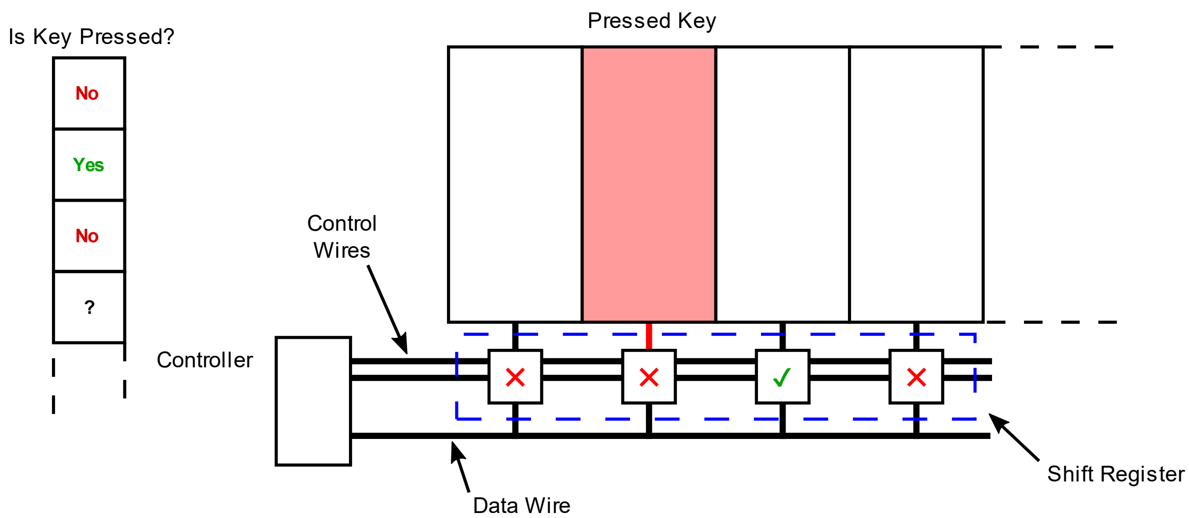
As shown in the following diagrams, this shift register will be used to select which key we wish to check the state of.



• 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.

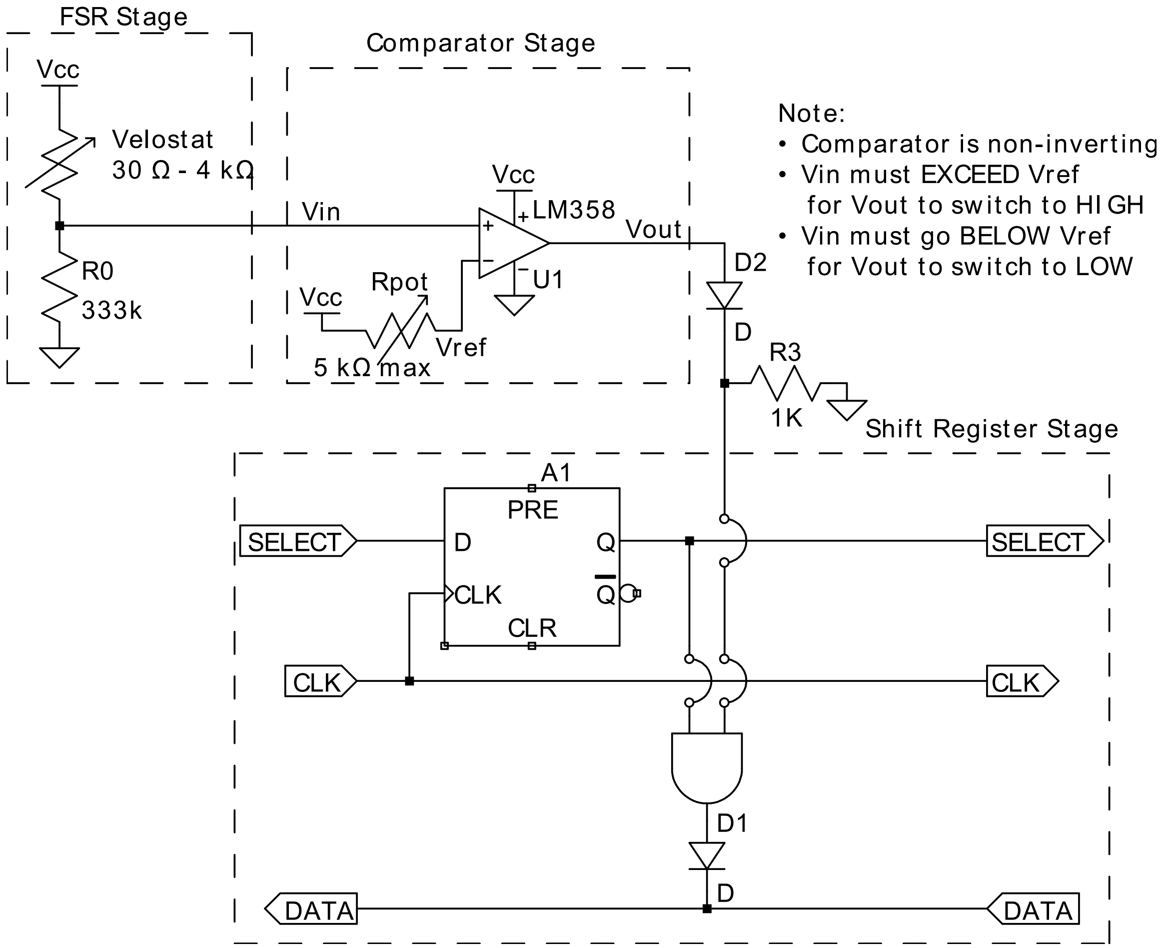
Individual Keys Circuitry:

The current 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.

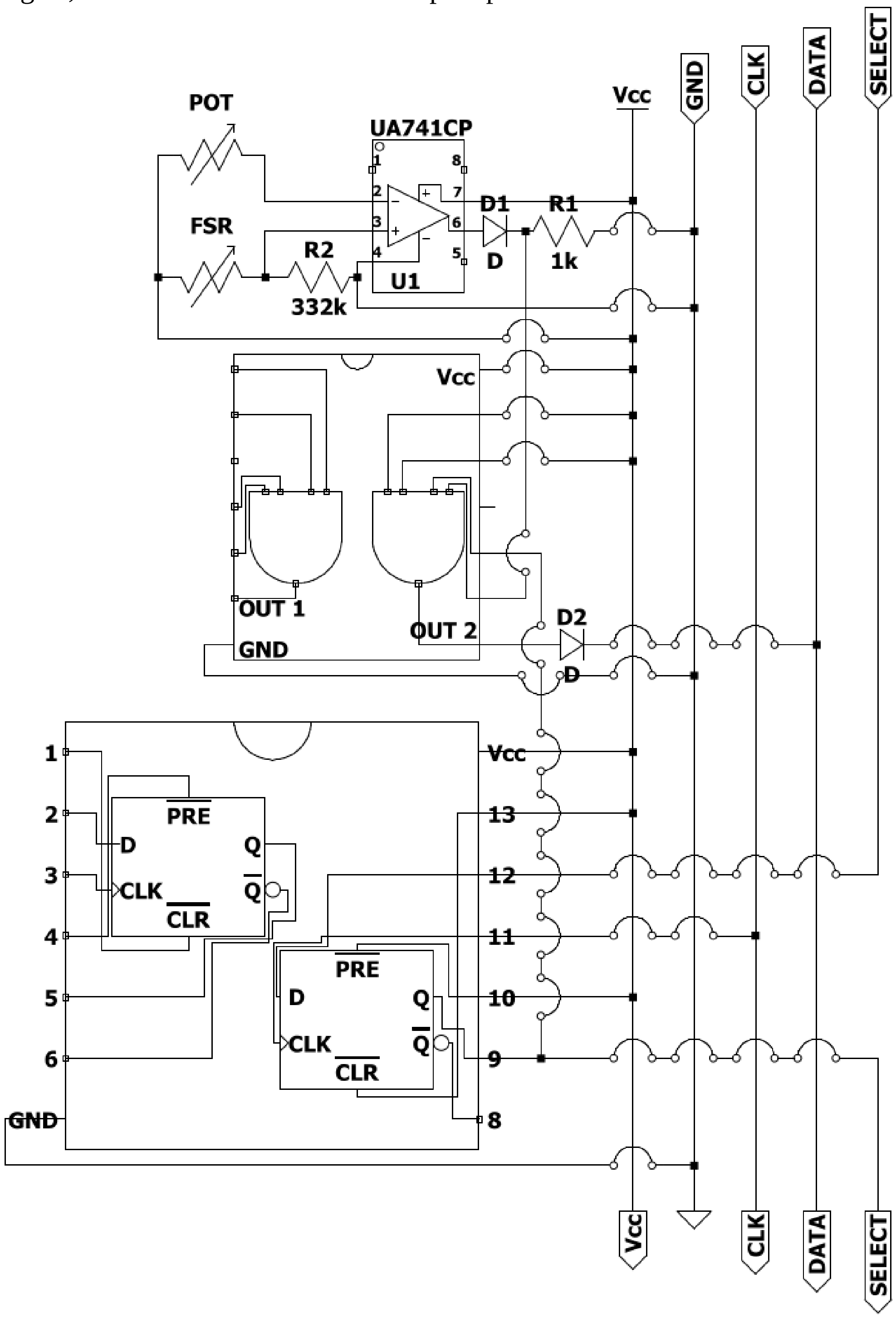
So, using a comparator circuit, we can convert this analog voltage into a "pressed" or "not pressed" digital signal. This comparator will include a potentiometer, for tuning how much pressure is required for the key to be considered "pressed" versus "not pressed."

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:

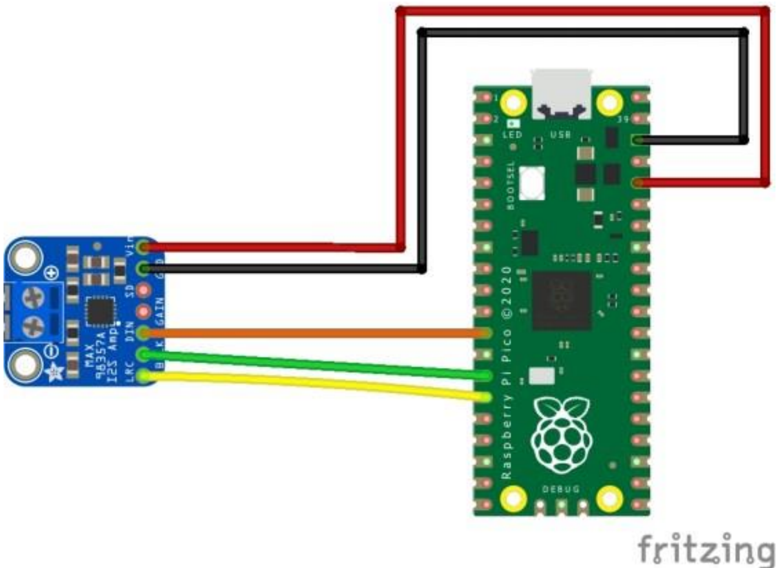

FSR Stage
Comparator Stage
Vcc
Velostat
30 Ω - 4 kΩ
R0
333k
Vin
Vcc
LM358
Vout
U1
Vcc
Rpot
5 kΩ max
Vref
D2
D
R3
1K
Note:
• Comparator is non-inverting
• Vin must EXCEED Vref for Vout to switch to HIGH
• Vin must go BELOW Vref for Vout to switch to LOW
Shift Register Stage
A1
PRE
SELECT
D
Q
CLK
Q
CLR
SELECT
CLK
CLK
D1
D
DATA
DATA

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



Controller

The old piano design relied on a connected computer to play notes. However, if piano is to eventually interface with a high-voltage system, I'd rather keep my laptop out of the mix. As such, the new design will be using an on-board Raspberry Pi Pico and audio amplifier connected to an audio output jack:



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

The Raspberry Pi already includes a micro-USB port, allowing for easy connection to any outlet using a standard adapter.

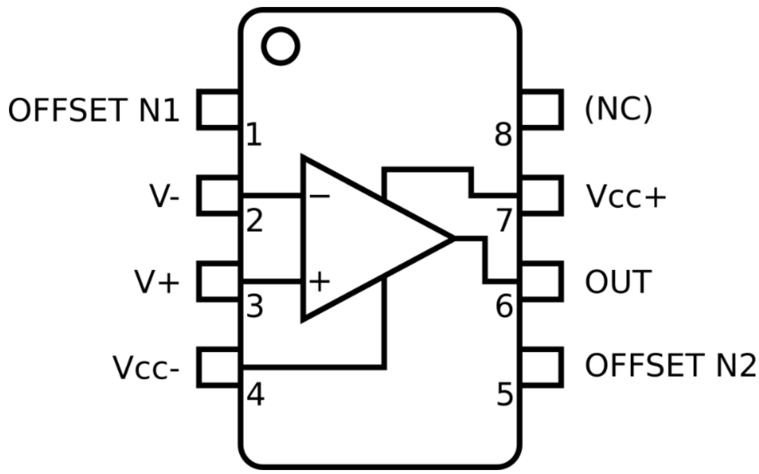
Required Materials

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	15	$ 74.25
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	$ 4.00	1	$ 4.00
Audio Amplifier Module	$ 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
		Subtotal	$ 164.24

Note: items marked as $0.00 are assumed to already be available

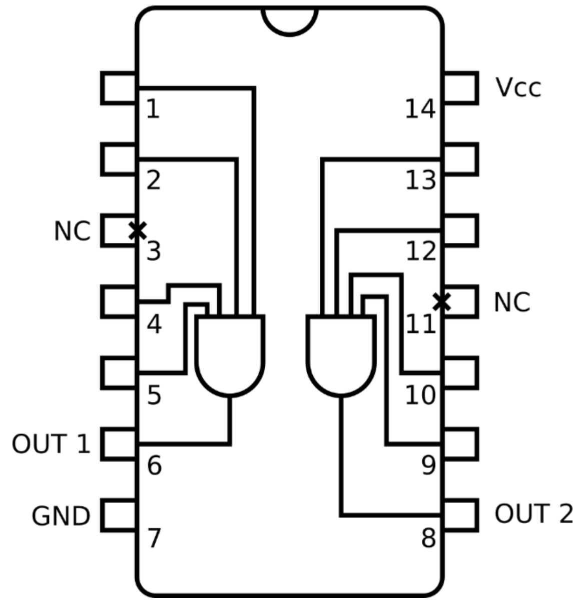
Component Pinouts

UA741CP Op-Amp:



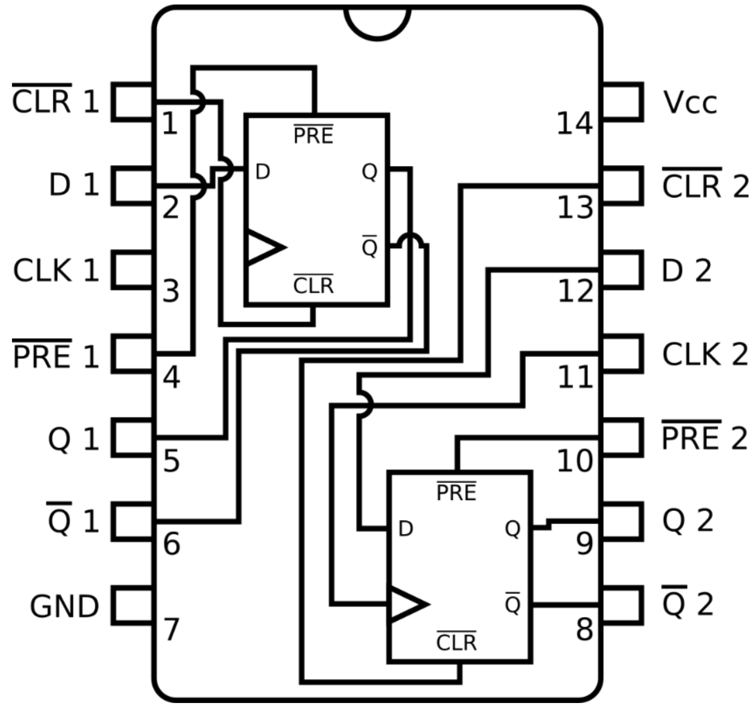
Pin		Description
Name	**Number**	
V-	2	Noninverting input
V+	3	Inverting input
NC	8	(No connection)
OFFSET N1	1	External input offset voltage
OFFSET N2	5	External input offset voltage
OUT	6	Output
Vcc-	4	Positive voltage supply
Vcc+	7	Negative voltage supply

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

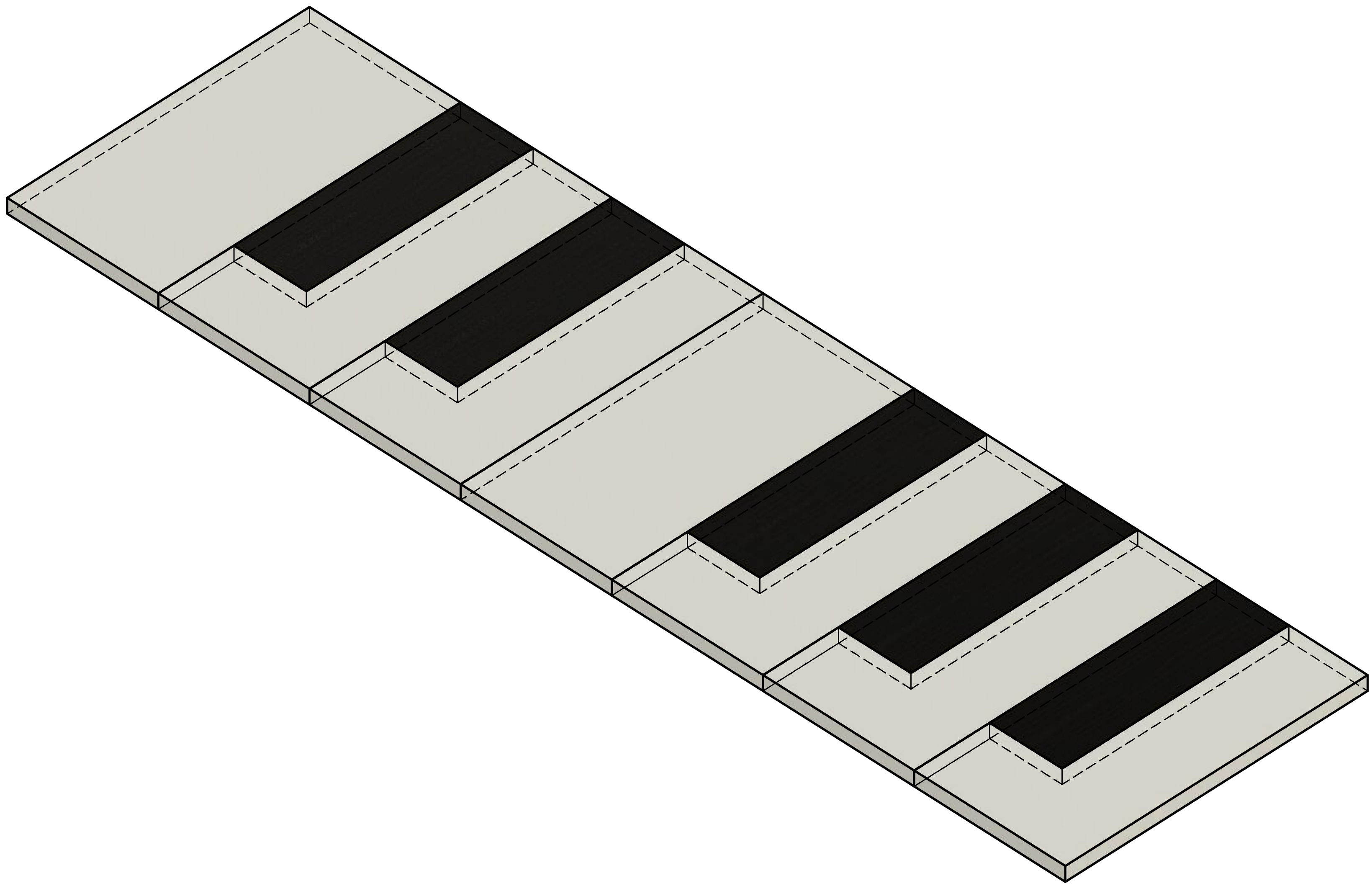
CD74HC74 D Flip-Flop:



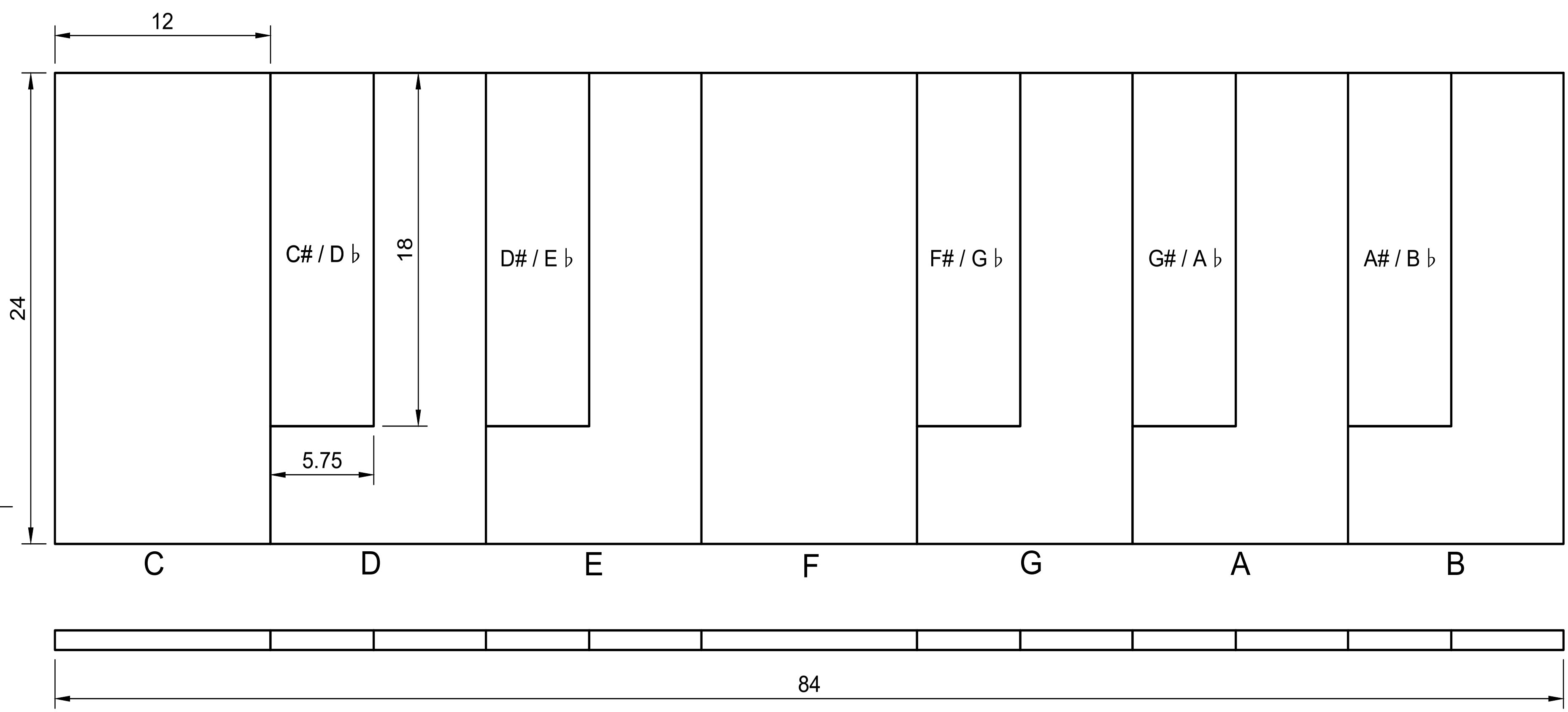
Pin		Description
Name	**Number**	
$\overline{CLK}$ 1	1	Clear input (active low)
D 1	2	Data input
CLK 1	3	Clock input (positive edge triggered)
$\overline{PRE}$ 1	4	Preset input (active low)
Q 1	5	Output
$\bar{Q}$ 1	6	Output (inverted)
CLR 2	13	Clear input (active low)
D 2	12	Data input
$\overline{CLK}$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



Dept.	Technical reference	Created by	Approved by		
		Will Ebmeyer 9/30/2022	ur mom		
		Document type	Document status		
		Title: Piano Remake Full Piano	DWG No.		
			Rev. 1	Date of issue	Sheet 1/2



Dept.	Technical reference	Created by Will Ebmeyer 9/30/2022	Approved by ur mom
		Document type	Document status
		Title Piano Remake Key Layout	DWG No.
			Rev. 1 · Date of issue · Sheet 2/2