Building a Player Piano

Brandon Switzer

Note

While this goes into detail of the mechanics of the player piano, this is not meant to be a guide on how to build your own. This is only a documentation of my journey in building the specific finished product you've already seen. I encourage you to make your own build if you feel inspired, but please recognize that your journey will be different than mine.

Inspiration

- I was looking for a project to work on during the summer of 2017
- We've always had a piano at my home, but it had never been used
- Inspired by what I had seen in videos and places I've visited, I decided to build my own player piano
- I also felt inspired to attempt to build something similar to a commercial player piano with less money

Initial Design

- I wanted not only for the piano to make sound but also for each key to physically move
- At first I thought that I was going to use motors to move the keys, but I quickly learned at solenoids (linear motors) were what I should use
- I had decided to control the electronics with an Arduino-based development board
- The Arduino board would take MIDI data from an outside source and control the keys

First Tests

- The first tests were done to figure out how I would send MIDI information from a computer to the Arduino board
  - The Arduino board takes serial input to communicate with other computers
- After installing a few programs onto my computer, I was able to use a MIDI player called Synthesia to communicate and activate some LEDs with MIDI on an Arduino Uno
- I would later change the way the Arduino Board receives MIDI, but this was a proof of concept for MIDI communication with the Arduino board

First Tests

- https://youtu.be/O_DePbJAjg

Electronics and Engineering

Finding the Right Solenoid

- The most expensive part of the project was researching and finding which solenoids I would use
- Solenoid requirements:
  - Strong enough to push each key down
  - Strong enough to loudly play each key
  - Manageable voltage and current
  - Doesn’t overheat quickly
  - Small enough to fit all of them in the piano
  - Low price (less than $500 for all of them)

Finding the Right Solenoid

- Eventually I found the JF-1039B solenoid sold from Eshinede in Shenzhen, China
- The solenoid is 24v and has 25n of force
  - It’s listed to take 2a but it really only takes 750ma
- Since we ordered so many solenoids, the company gave us a free JF-1250B solenoid, which would be used for the foot pedal
- The total cost for one hundred solenoids including shipping was $435

Box of Solenoids

Solenoids and Pyramid

JF-1039B and Other Tested Solenoids

Finding the Right Power Supply

- Now that I had the right solenoid, I had to find a power supply to power them
- Since the solenoids take 24v, I knew that current was what I needed to be looking for in the power supply
- I wanted for the power supplies to be able to power more than ten keys (for ten fingers) at a time because some MIDIs are designed without playability considered

Finding the Right Power Supply

- I found a 24v 20a power supply on AliExpress for $25
- While this would be enough to power twenty-six keys, I kept in mind that I still had to power the foot pedal and that some MIDIs can play even more keys
- Since the power supply was cheap, I decided to buy two for $50 and combine them to power up to forty-eight keys at once!

Power Supply

Wired Power Supply

New Design

- The Arduino board takes MIDI signal and converts it into separate 5v outputs
- To expand the number of outputs on the Arduino board, I used shift registers to convert three outputs into eighty-eight unique outputs
  - Each output has PWM capabilities, so I'm also able to control the volume, or velocity, of the piano
- The outputs feed into a line of transistors that power the solenoids
- For debugging and entertainment, each output has an LED so I can see what keys are being powered

New Arduino Board

- I learned that with an ESP32 development board I would be able to accept MIDI input through Bluetooth
  - This means that I'm able to accept MIDI from a portable device like an iPad
- I also learned that I'm able to establish an Arduino Pro Micro board as a MIDI device through USB
  - This means that I don't need to install extra programs on my computer to communicate MIDI to the Arduino board with USB; I can just plug it in and MIDI programs will recognize it automatically

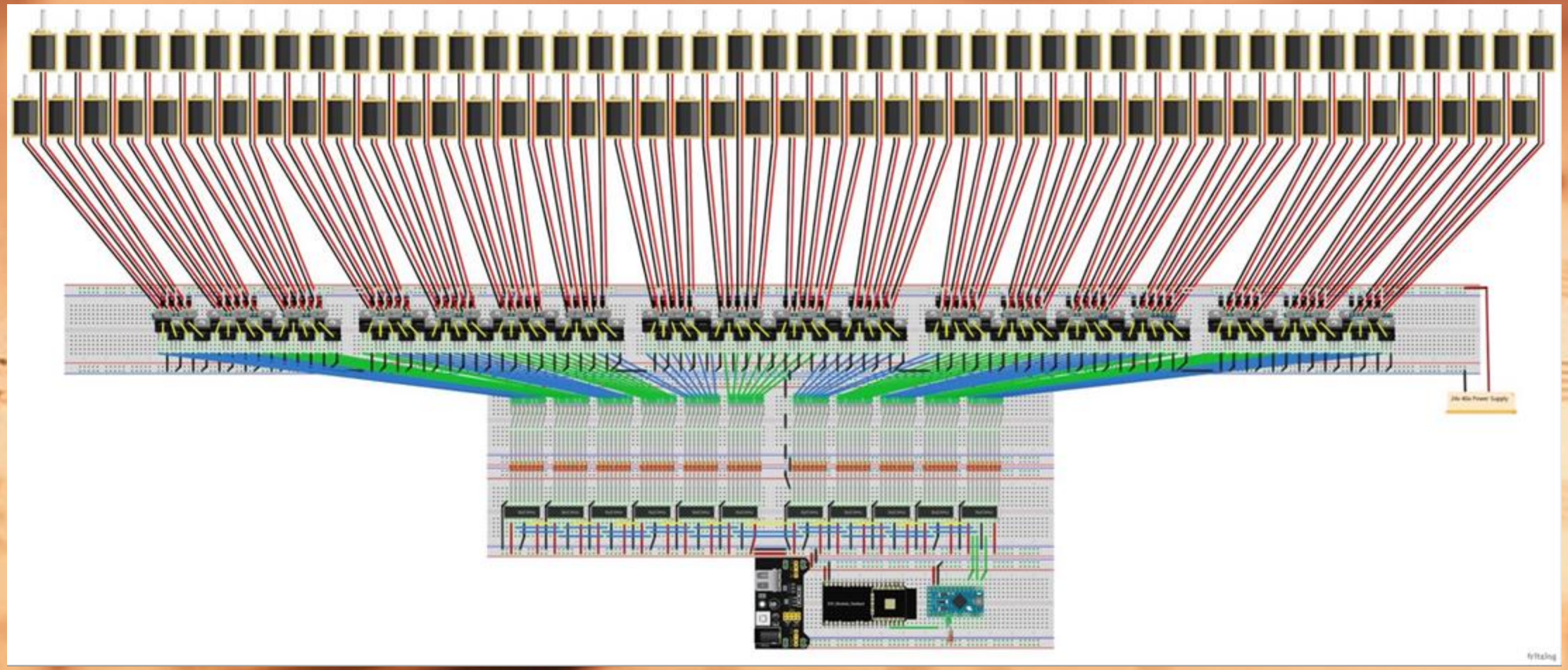
New Arduino Board

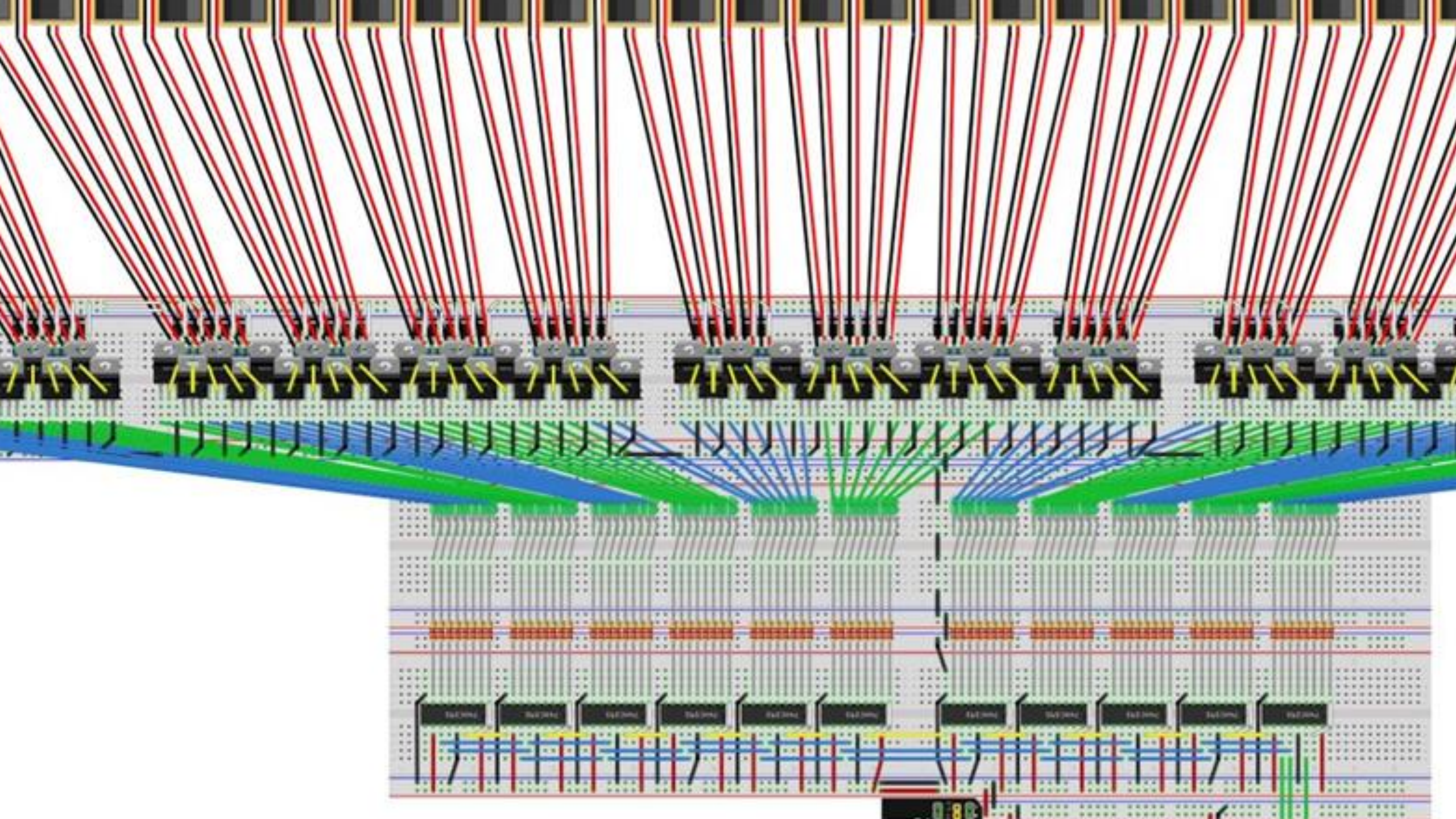
- To incorporate both Bluetooth and USB functionality, the ESP32 and Arduino Pro Micro to communicate with each other
- The ESP32 takes incoming Bluetooth signals and decodes it so it can be sent to the Arduino Pro Micro
- The Arduino Pro Micro can detect and decode both USB signals and signals coming from the ESP32, but it’s also the board that interacts with the solenoids

Mounting the Electronics

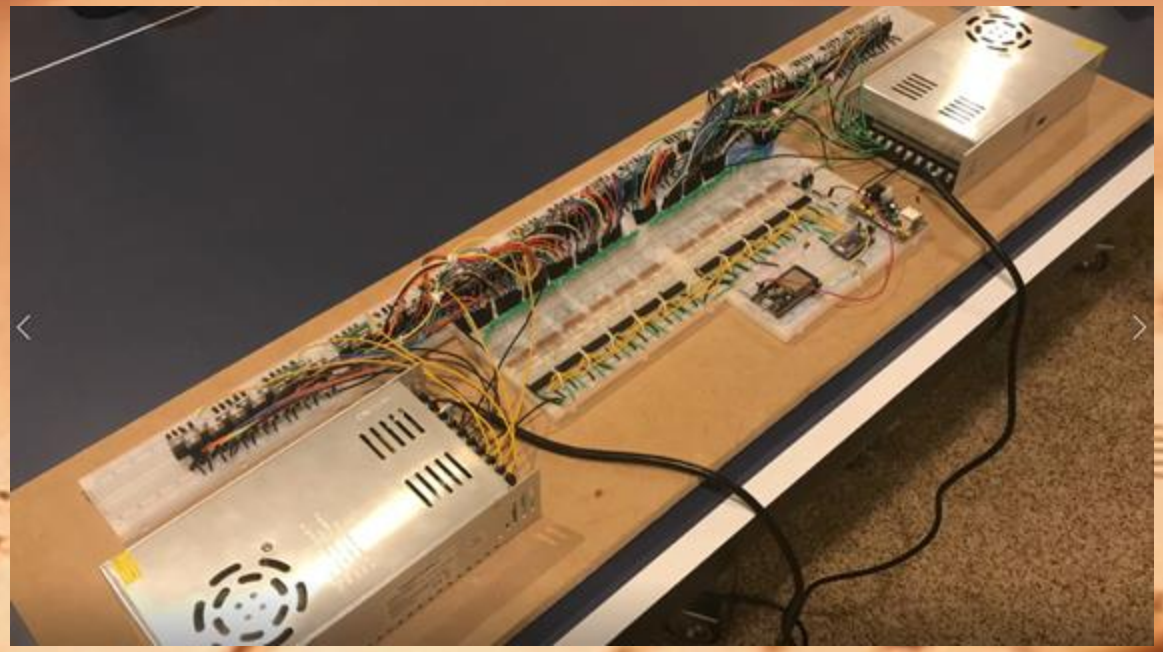
- The electronics are built onto solderless breadboards
- These breadboards are put onto a wooden board with the power supplies and are mounted under the piano
- The electronics are separated from the solenoids; therefore, if there’s a simple fix, I don’t need to take everything out

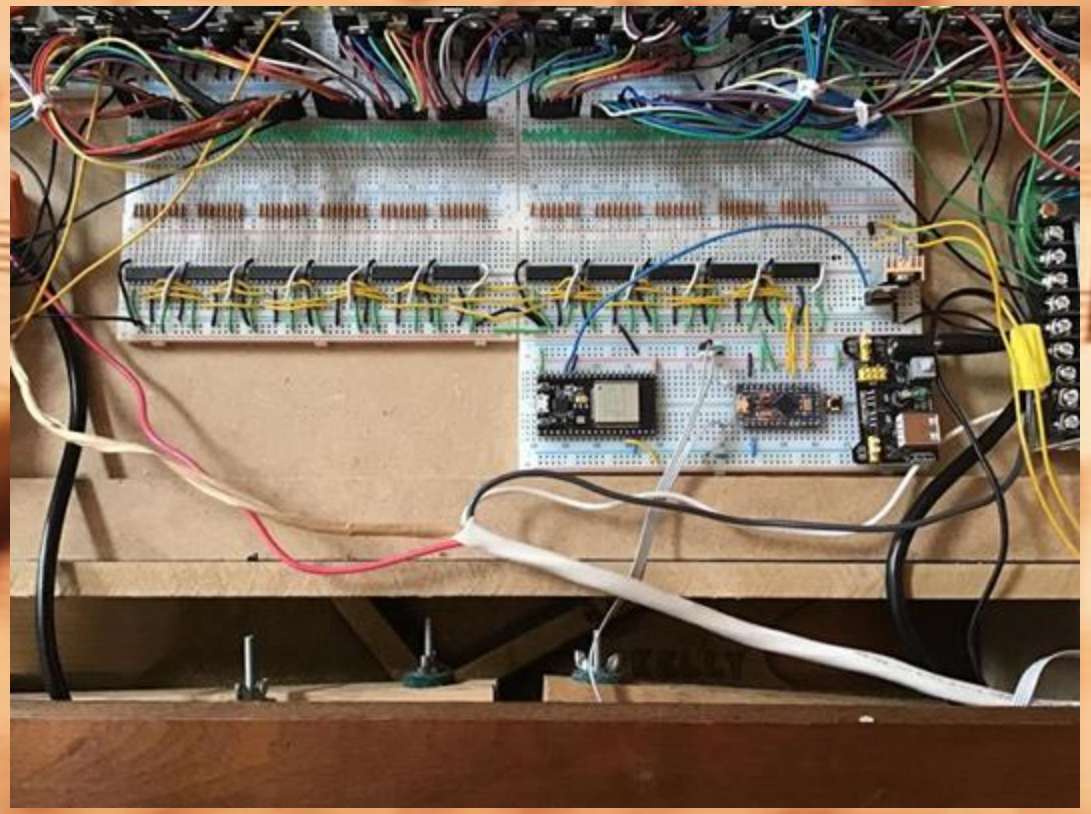
Visual Diagram of Electronics





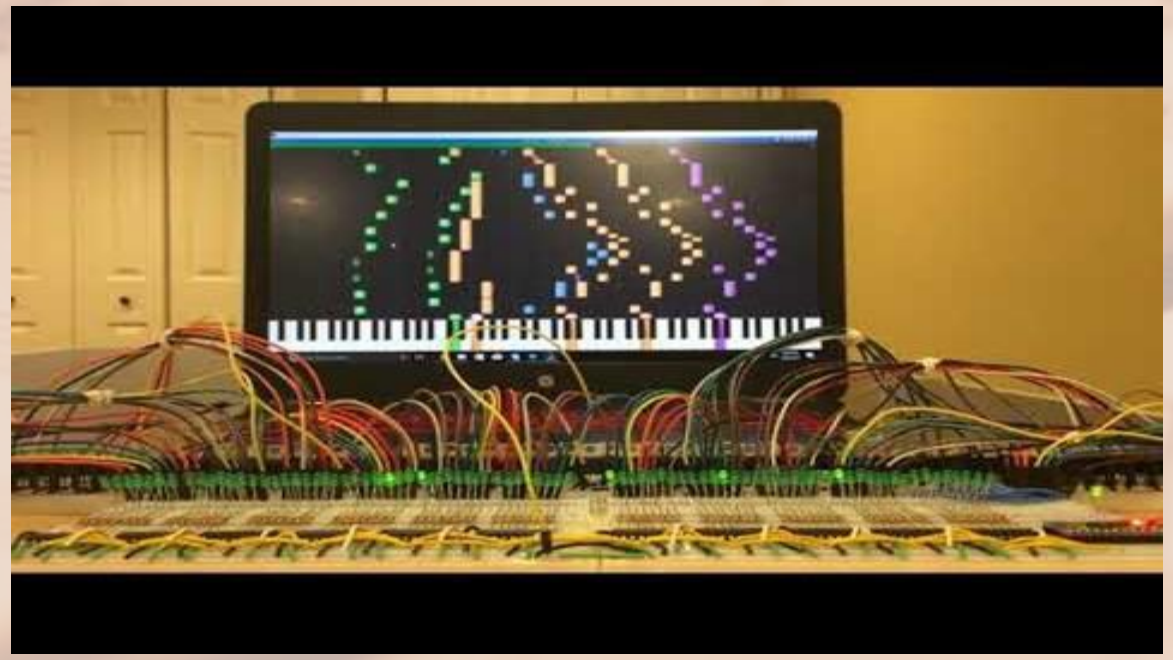
Electronics Board





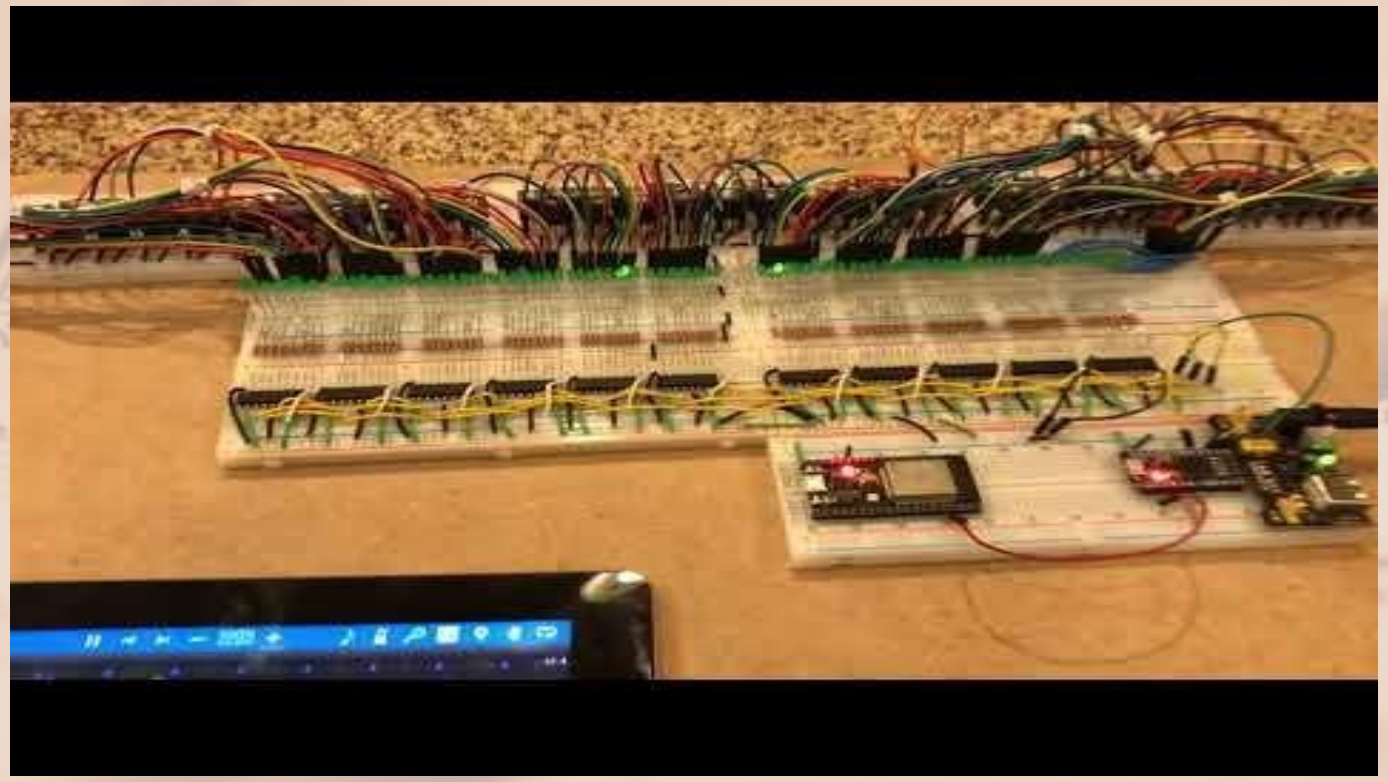
Electronics Test with LEDs

- Tested with impossible song “Death Waltz”
- https://youtu.be/nXQNJLkX5sg



Bluetooth Test With LEDs

- https://youtu.be/I-uOGNzoqZ4



Powering the Piano

- Each of the power supplies for the solenoids are powered using their own separate cords
- A smaller voltage for the Arduino Boards meant that a third, 3.3v breadboard power supply would take up a third outlet
  - All the Arduino boards are 3.3v
- A three-outlet power strip powers everything, and its wire extends to plug into an outlet outside of the piano

Mounting the Solenoids

- I used a clear plastic piece that runs the length of the piano and marked the center of each key onto it from above the piano
- I carefully transferred these markings from the plastic piece to the board which I would be mounting the solenoids on
- The markings on the board represents the center of the solenoid
  - This is where the solenoids hits each key

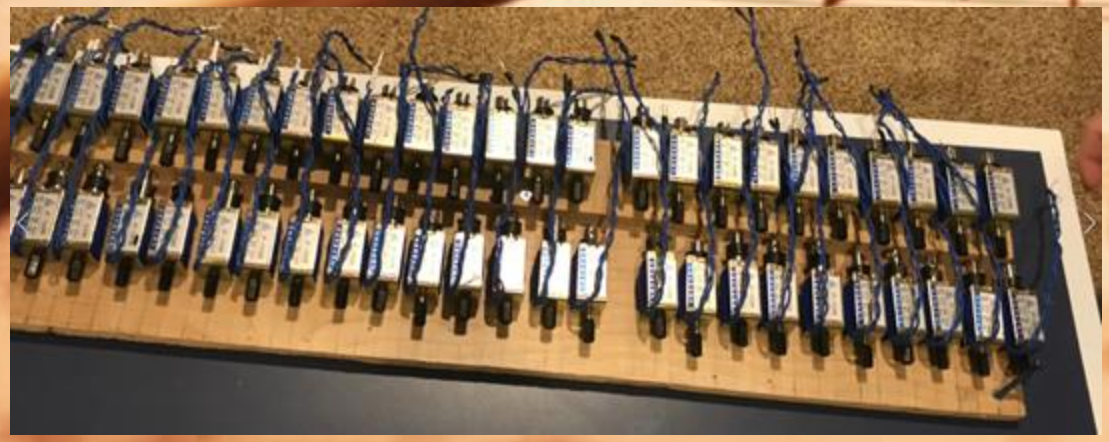
Mounting the Solenoids

- I had initially thought that I would be able to place the solenoids in a row on a single board along the bottom of the keys
- However, I had realized that the solenoids were too big to be placed side by side like this
- I only had three inches of space to work with under the piano

Mounting the Solenoids

- Instead, I staggered the solenoids along two different boards that gives enough room for the solenoids to be placed next to each other
- One board is mounted to the top of the underside of the piano and the other is mounted onto the same board with a half inch difference in position

Mounting the Solenoids



Solenoids to Hit Each Key

- There’s a thick board at the bottom of the key bead that runs all the way to the back of the piano
- This board blocks access to the bottom of each key from the underside of the piano, which is where the solenoids hit each key

Solenoids to Hit Each Key

- To give the solenoids access to each key, I used the same plastic piece I marked the center of each key on and transferred the markings to the bottom of the piano
- I drilled through the bottom of the piano to each key using the markings, which allows a small rod to move through the board and activate each key

Solenoid Length

- While there was now a hole from the center of each solenoid to the bottom of each key, the solenoids had no way to reach their respective key
- This is because the shaft that moves that is packaged with each solenoid is relatively short
- I needed to create an extension on the end of each solenoid that would increase its length and bring it to its key

Solenoid Length

- The end of each solenoid has a screw that allows the shaft to attach to other objects
- I attached plastic spacers to the ends of the solenoids so that I could easily screw in extensions on top of the solenoids

Solenoid Length

- To create this extension, I cut eighty-eight metal rods to the length I needed
  - Two sets of forty-four rods; one set with the length needed for the top row of solenoids and one set with the length for needed for the bottom row of solenoids
- I modified the end of each rod so that I would be able to screw it into the plastic spacers I had placed on top of each solenoid

Attaching the Extenders

- Each rod had to be carefully fitted onto each solenoid so that its length was perfect for each specific key
- I referenced the correct length for the rods by making sure that when a solenoid was fully activated, its respective key would be exactly at its fully pressed state

Attaching the Extenders

- After the rods were perfectly in place, I glued their connections to the spacers with epoxy to make sure that they would never lose their accuracy
- While the couplers did help me fit the metal rods, a lot of the rods had to be re-cut and fitted, making this one of the lengthiest parts of the project

Metal Rods and Spacers

Solenoid Sound

- When a solenoid is activated, there's a loud click of metal hitting metal because the shaft hits a stopper that disallows the solenoid from activating any farther
  - This becomes loud and distracting, even with the sound of the piano playing over the sound of the solenoids
- To prevent this, I fit each solenoid with two O-rings that stops the solenoid from clicking

Solenoid Sound

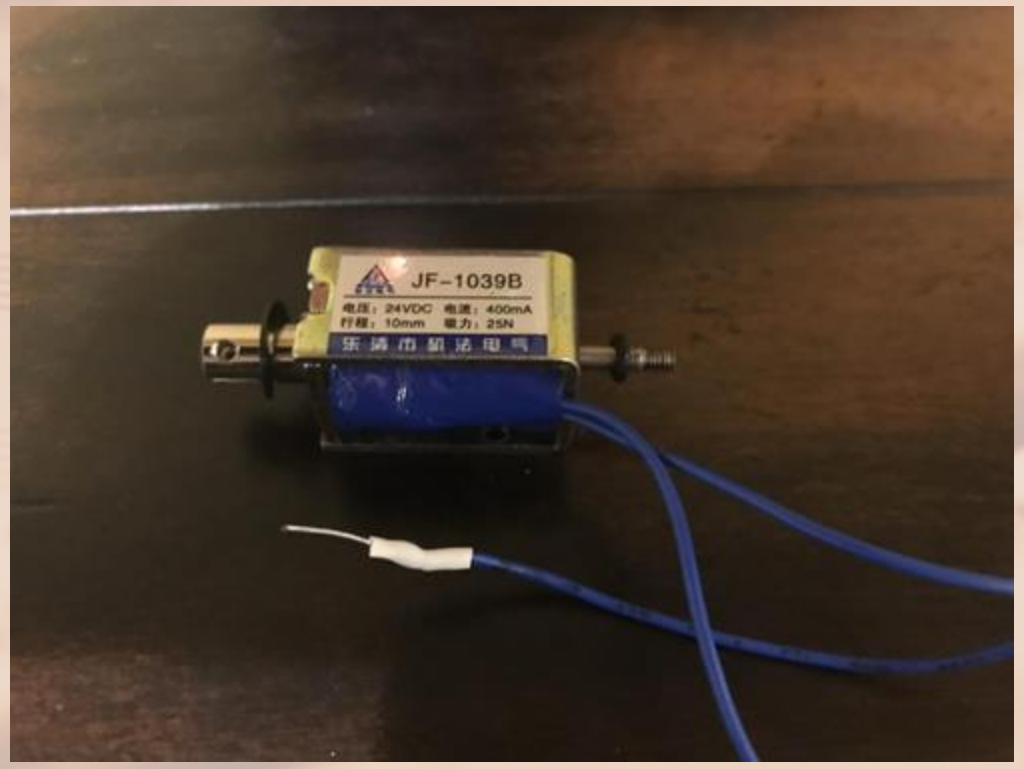
- While the O-rings help, there is still a little bit of sound that comes from to solenoids
- To quiet this, I stuck a flat insulation mat (usually used in cars) to the large board that covers the inside of the piano
- This deadens any additional sounds made by the solenoids

Stages of Preparing a Solenoid

- 1. Original Solenoid

Stages of Preparing a Solenoid

- 2. Take off spring, nut, and improve connectors



Stages of Preparing a Solenoid

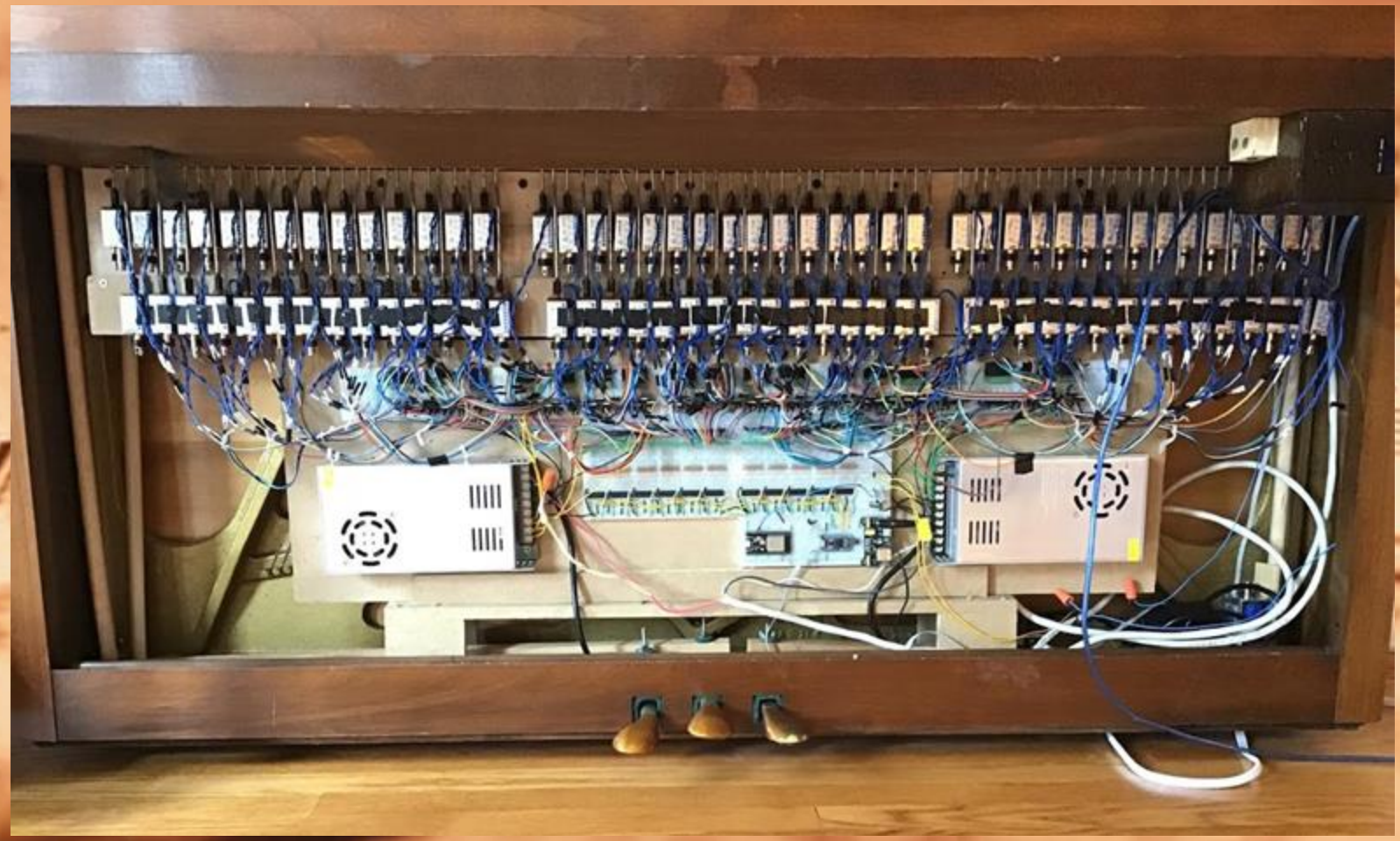
- 3. Attach rubber O-rings and plastic spacer (the metal rod is put in place in the piano)

Fully Mounted Solenoids

Solenoid First Tests w/ Random Keys

- https://youtu.be/GK7qUAPS8F0

Mounted Electronics



Improving Solenoid Performance

- While the solenoids have a stroke of 10mm, the keys do not (based on where the solenoids hit)
- Also, because the solenoids are calibrated to be fully activated when a key is also pushed all the way down, there is space between the top of the extender and the bottom of the keys when the solenoids are not activated

Improving Solenoid Performance

- This means that it will take longer for the solenoids to activate the keys and that the keys will undergo more stress
- To improve this, I added smaller rubber O-rings between the upper shaft of the solenoids and the spacer to make sure that each solenoid was still touching the bottom of each key, even when none are activated

Solenoid with Extra O-rings

- Amount of extra O-rings for each key is unique

Protecting the Keys

- With so much movement between the solenoids and the keys, I realized that the bottom of each key (where the metal rod strikes) would become damaged
- To fix this, I glued felt cut-outs to the bottom of each key

Keys with Felt

Foot Pedal

- The most used foot pedal is the right-most one, which is called the damper pedal
  - This sustains the sound of each key
- Since the other pedals are rarely used, I decided to only incorporate the damper pedal into the electronics
- To power the foot pedal, I used the JF-1250B solenoid I got for free with the rest of the solenoids

Foot Pedal

- Because the foot pedal requires a lot of force to push down, I needed to find a way for the solenoids to activate the damper pedal without using too much force
- I decided that instead of pushing down the whole foot pedal, the solenoids would only push through the few millimeters that the damper pedal activates

Foot Pedal

- The damper pedal has a wooden piece on the bottom of the piano that acts like a lever when pushed down
- A strong spring is placed under the wooden lever so the lever only has a few millimeters left to move and less force is required to activate it
- The solenoid is screwed into the wooden lever so that when it activates, it pushes the lever easily while only pushing a few millimeters

Foot Pedal Heat

- Originally I had used one JF-1250B solenoid to push the entire damper pedal
- While this worked, the solenoid and transistor that powered the solenoid needed too much power and overheated
  - This is especially an issue because the damper pedal is activated for long periods of time during songs

Foot Pedal Heat

- To fix this I used two solenoids in series
- This halves the current and divides the resistance by four
- Overall it puts less stress on the power supplies, transistor, and solenoids while still giving the same amount of force

Final Foot Pedal

Control Box

- Later into the project, I realized that it would be a good idea to add a control box where I could turn on/off the piano, change the volume, provide an easier place to send MIDI input through USB, and change different settings in the program
- I constructed a small box and finished it in a dark color to match the color of the piano
- The control box is powered by another Arduino Pro Micro
  - This means that I can input MIDI into the piano through a USB port on the box

Control Box

- A 16x2 LCD screen on the front prints a welcome message, different menus, and volume
- 5v, ground, and a serial wires for communication with the ESP32 go into the piano
- Other wires from the piano connect to the power supplies (including the breadboard power supply) so the piano can be turned on or off with a single switch

Control Box

Programming

Role of Arduino Pro Micro (in the piano)

- Receives USB MIDI messages
- Receives MIDI messages from the ESP32 through serial
- Takes these messages and immediately converts it into output for the shift registers and solenoids
- USB functionality is used for testing and debugging because it doesn’t schedule any notes

Role of ESP32

- Receives MIDI messages through Bluetooth
- Receives MIDI messages and setting changes from the Arduino Pro Micro in the control box through serial
- Schedules when notes will actually be played from the MIDI messages
- Sends scheduled MIDI messages to the Arduino Pro Micro in the piano to activate the solenoids
- Most powerful board and does the most work

Role of Pro Micro (in the control box)

- Receives USB MIDI messages
- Sends MIDI messages to ESP32 through serial to be scheduled
- Remembers different variables and settings for the ESP32
  - These are saved on the Arduino Pro Micro even when it’s turned off, but the ESP32 is what uses these settings
- When the piano starts up or a variable is changed, serial is sent to the ESP32 to change the setting(s)

Scheduling

- Notes need to be processed (scheduled) by the ESP32 because the piano has limitations that MIDI does not
- For example, MIDI files can overlap multiple versions of the same note over itself
  - Obviously, on a real piano, a note can’t be played twice at the same time
  - If this was sent directly to the solenoids the piano wouldn’t make the sound the MIDI intended to make
- It’s important to understand the piano and its limitations in order to schedule notes

Why “Schedule?”

- The reason why I use the word schedule when I talk about the ESP32 programming is because when the ESP32 processes a MIDI on or off command, there’s a series of timed steps that occur in order to turn a key on
- The ESP32 also schedules these steps so that two keys activated at the same time with different velocities will sound at the same time
- There is a universal delay for when any action will be completed, and everything that needs to be done to complete an action is scheduled by subtracting from this set delay
  - Delay is created by taking into account the longest possible time to turn a note on and off

States of a Key Press

- I separated the activation of each key into different stages that the ESP32 schedules
- The ESP32 knows the state of each key at any time
- When the ESP32 is given either a note on or note off command it uses the current state of each key to see if the note can be scheduled or not
  - For example, if for some reason the ESP32 is given a note off command and the note is already off, it wont schedule anything

States of a Key Press

- 1. Note is off
- 2. Note startup
  - When the note is beginning to turn on but the velocity of the note doesn’t matter yet
  - The ESP32 will schedule the solenoid at full power to make this part quick
- 3. Note activation
  - When the note goes to hit the string and the velocity matters
  - The ESP32 will change the PWM of the solenoid to create velocity
- 4. Note is on
- 5. Note is turning off
  - Time between when the solenoid deactivates and the key is moving to its off position

Fast Deactivation

- When a key moves from its activated position to its deactivated position, the only force pulling the key up is gravity
- However, when the key is quickly pressed and let go immediately, the note will deactivate quickly because the force used to activate the key will help the key “bounce back” into position
- The ESP32 accounts for this in its programming, and this is what allows notes to play extremely fast

Special Cases

- There are special cases where a MIDI file will send information that is impossible for the piano to play
- Sometimes the ESP32 is able to change the way it interprets and schedules this information to still create the intended sound

Special Cases

- Note on command while the note is already scheduled
  - If the note is on, the note will deactivate and reactivate
  - If the note needs to begin deactivating and reactivating while it’s in the middle of turning on, it will deactivate with fast deactivation
  - If the note is also scheduled to turn off but its current scheduling doesn’t give time for the note to reactivate, the note off command will be rescheduled so that the note has time to reactivate
  - If the note is activating while also scheduled to turn off but its current scheduling doesn’t give time for the note to reactivate, the rest of the note will be rescheduled so that the note deactivates with fast deactivation and then reactivates normally
  - Otherwise the command is ignored

Special Cases

- Note off command while another version of the note is still activated
  - This is a result of when a MIDI overlaps the same note multiple times
  - The ESP32 keeps track of how many variants of the same note are activated
  - The note will only turn off if it’s the last variant of the note

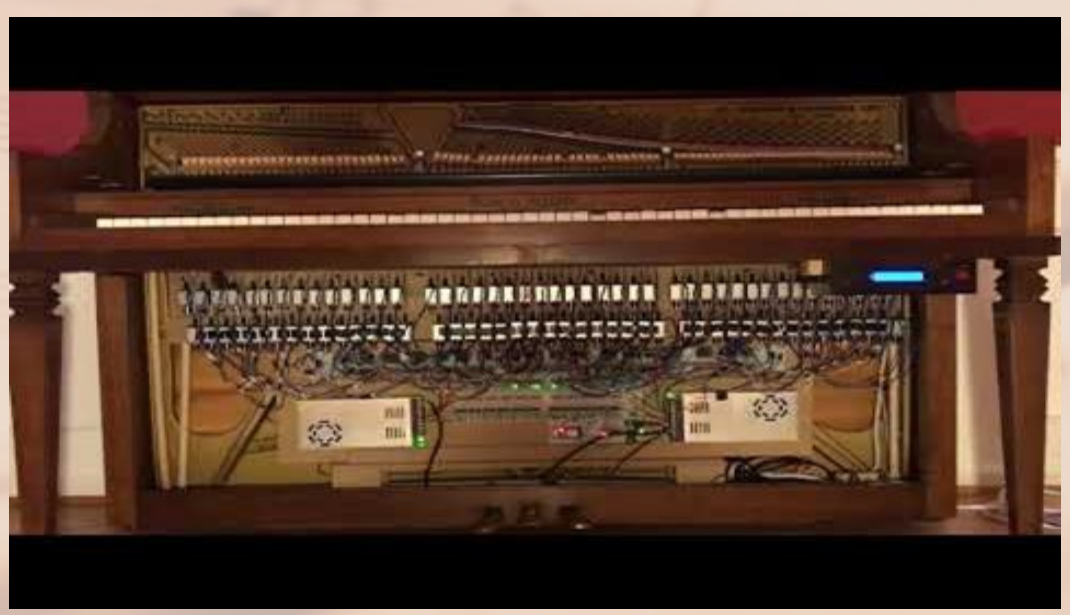
Final Product

Time-lapse and Showcase

- https://youtu.be/S7Bd992k368

Example Song – La Campanella

- This song shows the efficiency of the solenoids and how fast they are able to play note
- https://youtu.be/lDdaMBsiQBA



Features

- Play any MIDI file with a computer program or smart device app with the output as the player piano
- Record your own songs with a keyboard and play it back on the piano with a computer using a MIDI recording program
- Can be used to practice, play along with the piano as a duet, or even play impossible MIDI files that no human could play (Death Waltz or Circus Galop)
- Using an app called PianoStream you can stream music to the piano like a radio
- Play around with converting sound files to MIDI files – you can hear your voice playing from the piano!

Advantages Over Commercial Player Pianos

- Commercial player piano systems cost at least $7000 and go over $10000
- If you actually want to play a song on your commercial piano you need to pay extra (for each song)
- Commercial systems don’t always provide a wide range of input
  - Usually comes with an app that you have to use
- I could install a player piano for a client for $2000 maximum, and I could bring down the price if there was a special reason to
- There’s no need to pay for songs because MIDIs are publicly available and always being made
  - Play any song you can think of, whether it’s old or new or made for piano or not

Future Ideas

- Be able to record your own MIDI files using the piano
  - Put sensors under each key to detect when each key is pressed
  - These sensors will also be able to detect velocity
  - Record and instantly play back your own songs without using an external keyboard
- Incorporate both audio and MIDI
  - Play a song on the piano but also have an audio portion of the song play over on a speaker
  - This can easily be done, but there’s currently no app that supports it (using a computer would be too complicated)

Parts - Solenoids

- Keys: JF-1039B Solenoids (88)
- Pedal: JF-1250B Solenoids (2)
- M4 Screws (184)

Parts – Development Boards

- ESP32 Development Board
- Arduino Pro Micro Development Boards (2)
- Male Pin Header Strip (Soldered to Pro Micros)

Parts - Power

- 24v 20a Power Supplies (2)
- Breadboard Power Supply (1)

Parts - Wires

- Ribbon Cable
- Assorted Hook-up Wire - 22 AWG
- Power Strip – 3 Outlet

Parts - Electronic Components

- Solderless Breadboards – MB-102 (9)
- Terminal Block – 10 Pins PCB (9)
- Boat Switch – 4 Pins
- Assorted Push Buttons
- LCD Display – 16x2 IIC/I2C Blue Backlight

Parts - Electronic Components

- Shift Registers – 74HC595 (11)
- LEDs – Green Diffused 3MM – or Diodes (89)
- Resistors - 220 OHM (88)
- Darlington Transistors – TIP120 NPN (90)
- Potentiometer – 10k OHM

Parts - Other

- Spacers – Hex Plastic M4
- Assorted Mini O-Rings
- Steel Wire
- Epoxy
- Heat Shrink
- Insulation – Car Mat Sound Deadener
- Solder
- Heatsink

Cost

- Everything that went into the piano cost less than $700
- Additional costs from testing and equipment goes upwards of $1000
- After uploading a few videos to YouTube I generated enough ad revenue to pay for the project