Mesmerizing Ferrofluid-Display: Silently Controlled by Electromagnets

by SimenZhor

Disclaimer: This Instructable will not provide a straight forward way to build a big ferrofluid-display like our "Fetch". That project is so big and expensive that anyone who want to build something similar will almost certainly have different design requirements than we did. Therefore we will instead focus what we've learned from building "Fetch", which traps you should avoid, and which details you should pay the most attention to - as well as some tips and tricks for handling ferrofluid in general.

The team behind this project is currently discussing options for making a smaller, more affordable, ferrofluid-display for more realistic replication by hobbyists. When that work is done we will write a more detailed, step-by-step, Instructable and link to it here. To give a realistic perspective of the timeline we're working by, it is unlikely that such a project will be done before the end of 2021. It's developed entirely on a hobby basis by students at the University of Oslo.

All that being said: should you actually want to build "Fetch" yourself, all code and design-files are open source and available. It is a project in continuous development, so it is possible that technical details in this Instructable are already outdated when you read it. The main source of updated information will be Applied Procrastination on YouTube.

Supplies:

- Roughly 60ml EF-H1 Ferrofluid by FerroTec. We bought from this supplier.
- 252 Electromagnets. We used JSP-1515 from this supplier.
- 252 10mm M4 Screws. Example: AliExpress Affiliate Link
- 252 2-pin Connectors (both sides. Example: AliExpress Link
- 2 Acrylic sheets for laser cutting (can be replaced by aluminum sheet and CNC router)
- 10 of our custom printed circuit boards (PCBs). We are currently working on redesigning these to a version where 12 will be needed instead of 10. More info on our Hackaday.io page
- 1200W Server PSU. Example: AliExpress Affiliate link
- Power distribution board for the PS. Example: AliExpress Affiliate link
- 10xMolex-terminated (MiniFitJr) power cables (12 instead of 10 for new PCBs) Example: AliExpress Affiliate link
- Teensy 3.6
- Adafruit DS3231 RTC Module
- Some sheets of plywood
- Some wood-screws and wood glue
- Some 2mm glass sheets
- Some 6mm glass sheets
- Epoxy glue
- Kosher salt
- Distilled water
- Some wires

https://www.youtube.com/watch?v=5PFgVtzsXHM

Step 1: Design Requirements and "Nice to Have"s

After doing a few tests, proving to ourselves that the concept would work, we wrote down a set of requirements. Here we will discuss what was important to us and which design choices and restrictions that led to down the road.

The tank shall be clear and resist stains from the messy black Ferrofluid

Before starting our project we researched what had been openly shared about ferrofluid by others, and unfortunately it wasn't much. The best resource we could find was very detailed and good though, and it was right here on Instructables. "Really Beautiful Swimming Ferrofluids" by rogercarr is a great place to start for learning how to store ferrofluid for any kind of display. The biggest take-home message for us was to *leave saltwater in the tank for a while* ***before*** *adding ferrofluid.* This will be repeated as we discuss the tank further in the next step of this Instructable.

We also decided to go for a glass tank because of this requirement. It's easier to keep clean than acrylic, but after sharing our project online we've been approached by several other makers that have had some success with acrylic. A decent "super hydrophobic coating" should help making acrylic a good option, but we have not had the chance to test this ourselves.

The display shall be general purpose, not only special case

For many use cases, permanent magnets and servos will be a good way to actuate the ferrofluid. Especially if you want to make something specialized, like a clock. We however, wanted our display to be as "general purpose" as possible, so it *could* display a clock but also a lot of other cool animations. This meant we had to use electromagnets instead of permanent magnets and servos, which has several additional benefits:

- Easier mechanical design. We're electronics-engineers, so that's a big win for us.
- Fewer moving parts. That means fewer parts coming loose or breaking.
- Silent actuation. A dozen servos working in tandem makes a lot of noise!
- Thinner assembly. Since the electromagnets can be turned off entirely, nothing has to move away from the tank.
- Bonus that we're not implementing: It's possible to change the polarity of a magnet by sending current the other way. This would be a cool feature to add later.

The ferrofluid shall be lifted vertically

We didn't want to "cheat" in order to lift our ferrofluid. So: no horizontal display with a mirror at 45 degrees or anything like that. If it could be done, we wanted to do it vertically. This means that we have to *fight gravity* at all times when displaying stuff, and in terms of design restrictions we need to be able to control each magnet individually and also ideally control the force they're exerting on the ferrofluid. In other words, we could not use any multiplexing technique to simplify the electronics, because that would decrease the duty-cycle too much for it to be able to lift ferrofluid. In the end we were also able to dynamically change the force of a given "pixel", and it is actually done by utilizing this exact phenomenon to our advantage. Because the electromagnets apply less force due at a lower duty cycle we can use Pulse Width Modulation (PWM) to control how big the ferrofluid blob is at a given point. This has limitations, though, such as a maximum and minimum size of the blob, but more notably: we just stated that we need individual control of the electromagnets, and no microcontrollers can provide 252 PWM outputs. We will discuss this challenge a bit later.

We need to be *able* to display a clock

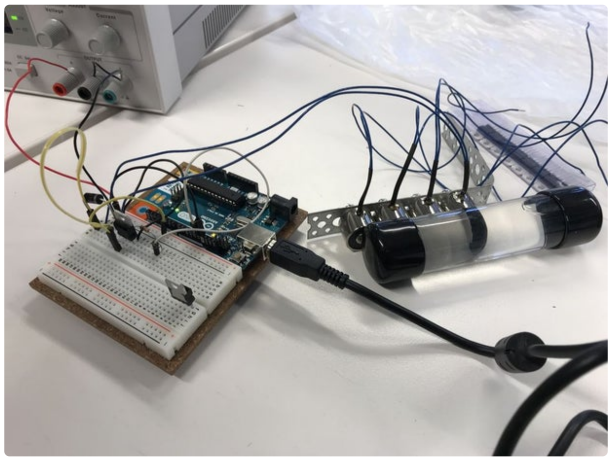
While we stated earlier that we want a general purpose display, we still want to be able to display a clock. So therefore we need enough magnets to display 4 digits and a colon separator, such as "13:37". The absolute minimum size of one digit is 5x3 pixels, but we also need one column of separation between each digit, and two rows of separation from the bottom. We therefore find that 7x17 is the smallest possible matrix we can use within this requirement.

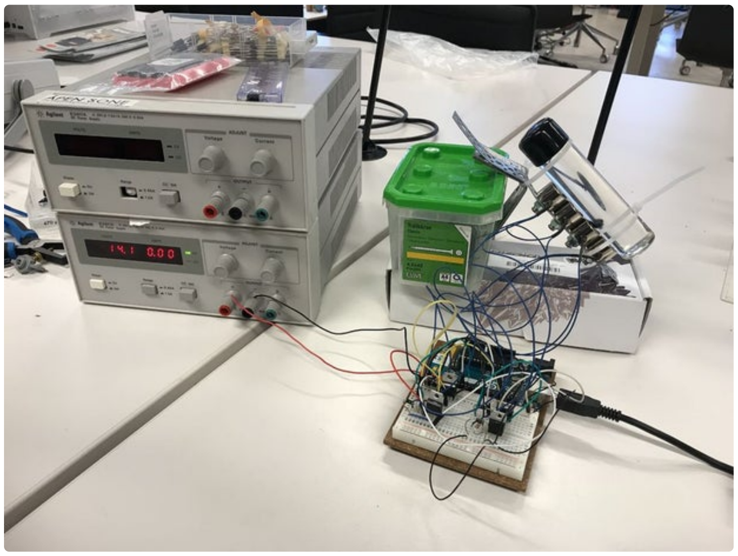
We want something close to a 16:9 format

Since a clock is not all we want to be able to show on our display, we did not settle for 7x17 magnets. We thought a 16:9 format would be nice to have, and we wanted to have as high resolution and as high pixel density as possible. This meant we had to make a tradeoff. Our total budget for the project was roughly $1500, and increasing the resolution quickly means a lot of magnets. In addition to this, smaller magnets are not necessarily cheaper than larger magnets. They also provide less holding force, and with high shipping costs we weren't able to test out several sizes before making the decision on which size we wanted to go for. The cheapest, most common, size is 20mm in diameter - but with that size the display would quickly grow huge so we wanted to cut it down as much as possible. In the end we decided to settle for 12x21 (252) electromagnetsmagnets with a diameter of 15mm, which seemed like a good deal to us. It was a bit of a gamble, but worked very well.

We want to have on-board storage for animations

The logical choice here is to include an SD card slot somewhere, and to make things easy for ourselves we decided to base the system around a Teensy 3.6 development board. It has an added benefit of being extremely fast compared to the most common Arduino boards, plus it's almost perfectly compatible with all the benefits Arduino brings. We did use an Arduino Mega for prototyping though, and with the next generation of PCBs (V2) we could've probably kept using it. More on this when we discuss PWM later.





Step 2: The Tank

The first major obstacle we'd have to overcome was how to manufacture a tank that didn't stain easily. Since we are associated with a University we were fortunate enough to have a lot of equipment and help available to us in order to solve this problem, but that also means it's the step we have the least detailed knowledge of how to replicate - but we *do* have some good tips, so don't stop reading just yet. As mentioned in the previous step: "Really Beautiful Swimming Ferrofluids" by rogercarr is a great place to start for learning how to store ferrofluid for any kind of display. That's what we did, and we will refer to the procedure to that Instructable throughout this step. We had already decided that we wanted to use a glass container, so we set up an experiment attempting to determine what kind of treatment would be the best to remove impurities inside the tank. Impurities and rough surfaces gives the ferrofluid places to get stuck, leaving stains, so it's important to get a smooth surface. We tested 4 types of treatment: lab grade acid, lab grade base, regular vinegar and "nothing" for control. In addition we used two different types of salt to make the brine used for suspension. One that contains anti-clumping agents, and one that's plain sea-salt (if it's marked as Kosher, that's a good sign for making sure you get the latter). We made two of every single combination, allowing us to tell if our procedure was inconsistent - and it was.

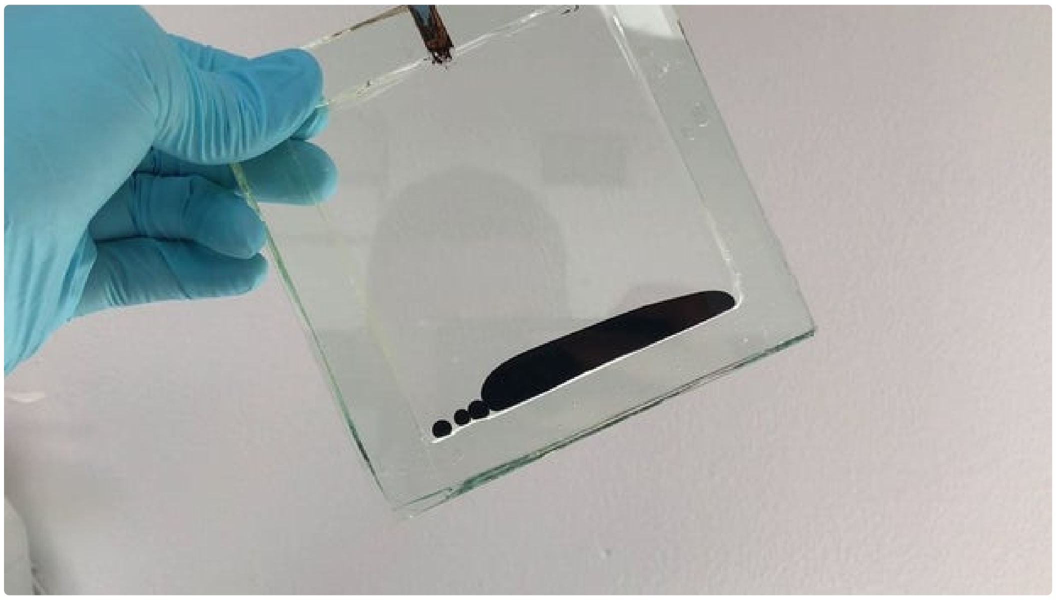
That meant the results of that experiment were pretty much inconclusive. Some samples stained heavily and others didn't do too bad, but none were perfectly clean. The dead giveaway that something was wrong was that we would get different results from the identical samples. But that didn't mean we didn't learn anything!

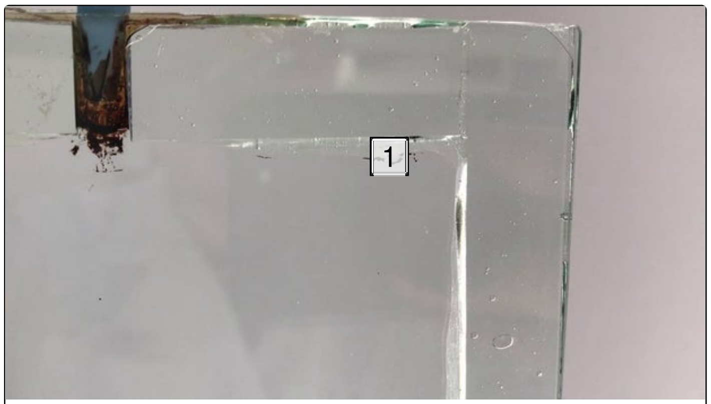
What in our opinion seems to be the most important step of the procedure in the Instructable linked above is that you need to ***leave saltwater in the tank for a while before adding ferrofluid***. My personal, very unscientific, guess to why this works is that salt-crystals or gas bubbles "fill"/smooth out the impurities. As you can tell, I'm no chemist myself, but perhaps someone can help elaborate on this in the comments.

Another, unrecorded, variable that we suspect had a big role to play in our inconsistent results was where in the main beaker containing "boiled down" ferrofluid we placed the pipette for transfer to the samples. In general we aren't perfectly sure that boiling down the ferrofluid is actually necessary, but we've always done it ourselves, so we cant recommend to skip that. If you **do** boil down though, be sure to stir it right up until the point where it's going to be transferred, and attempt to place the tip of the pipette above the bottom, but below the surface of the ferrofluid. In our experience the bottom contains more coagulated/clumpy ferrofluid than the middle does, and at the surface it's easier to pull up air-bubbles.

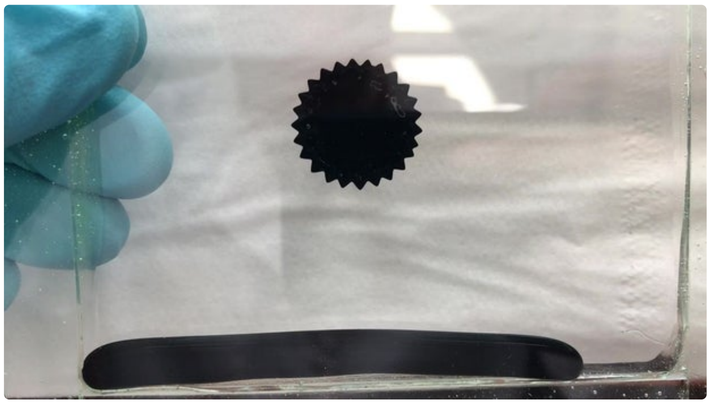
At this point we had received a prototype-tank that was made by our university glass-blower. He made it from two 100x100mm pieces of 2mm thick glass, separated them by a brim of 6mm thick glass and glued together with epoxy. We had already ordered the full-size tank, so the prototype was deemed expendable enough to fill with ferrofluid even though the experiments were inconclusive. It was a bit of a gamble, but since it did work, repeating the experiment was not a high priority for us. We do hope to repeat it in the future to get more consistent results on what works well though, and if we do the results will be posted on our YouTube channel Applied Procrastination.

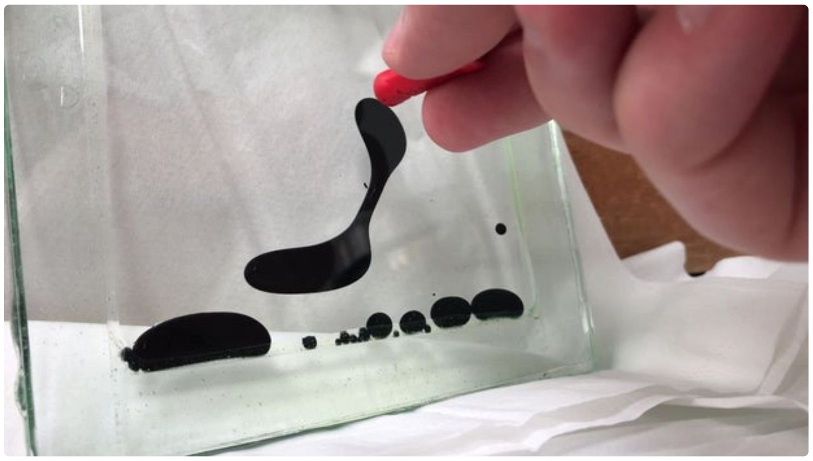
LOCTITE
POWER EPOXY
UNIVERSAL
SETTING 5 MIN TIME
28g / 25ml

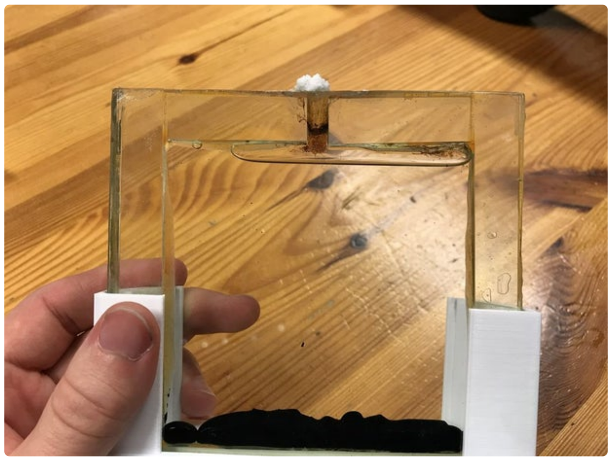


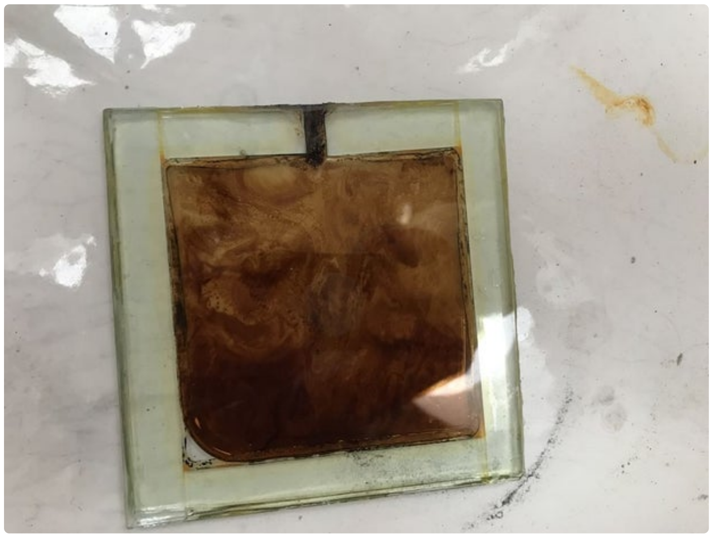


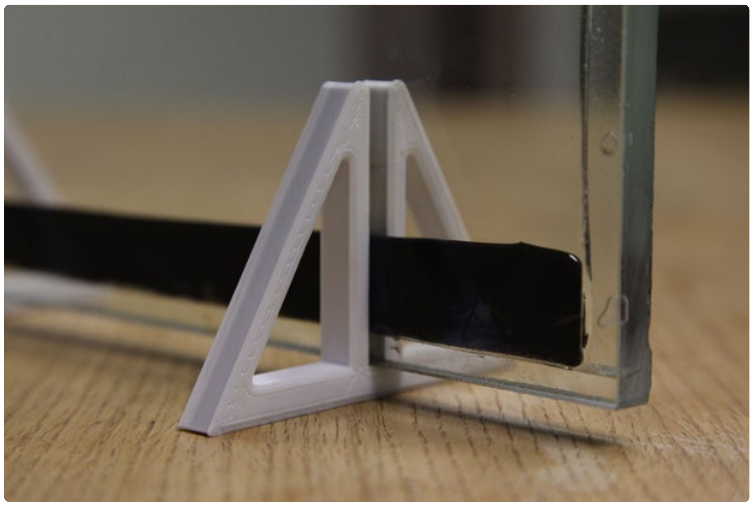
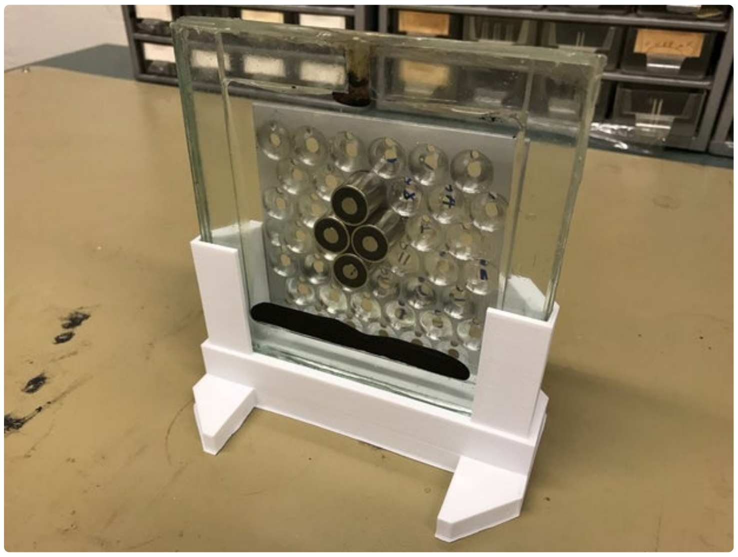
1. These minor stains along the top were the only ones we saw in our prototype









Step 3: Mounting Magnets

The magnets need to be mounted to something, and there are several factors to consider when deciding how to do that. The ones we were most concerned about were these:

Pattern of the magnets

Our intuitive idea was to align magnets in a hexagonal pattern because that would give us the largest effective area, reducing the size of the display. By effective area I mean that most of the area is covered by a strong magnetic field, because there are fewer/smaller gaps. This strategy has a *huge problem* though, and it was not clearly apparent to us when we started out - so our first magnet mount is actually in this hex pattern. The problem with it is that any two magnets that you want ferrofluid to move between need to be of opposite polarity - so that their magnetic fields can reconnect. If two magnets has the same polarity their fields will oppose each other, creating a sharp line of no magnetic field in between them. The ferrofluid can't cross this line unless it already has momentum enough to go past as one of the magnets is turned off. In the hex grid, the opposite polarity magnets from any given position will be heavily leaning in one direction (see illustrations in the photo section of this step). This means that drawing things is a lot harder (not impossible), and certainly more time demanding. The proper way to deal with this problem would be to implement the possibility to reverse the current in the electromagnet, giving it the opposite polarity, but that was a bit too complex for our plans and budget.

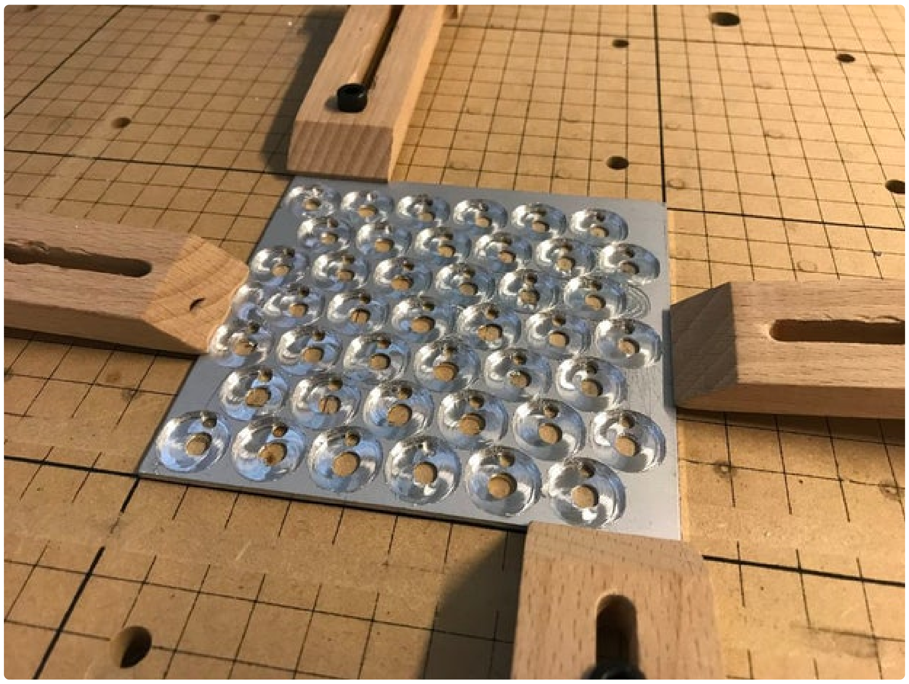
We therefore went for the, more common, Cartesian "square grid". Here the opposing polarity electromagnets align along the x and y axis, making it a tad simpler to program and animate. We have not regretted this decision.

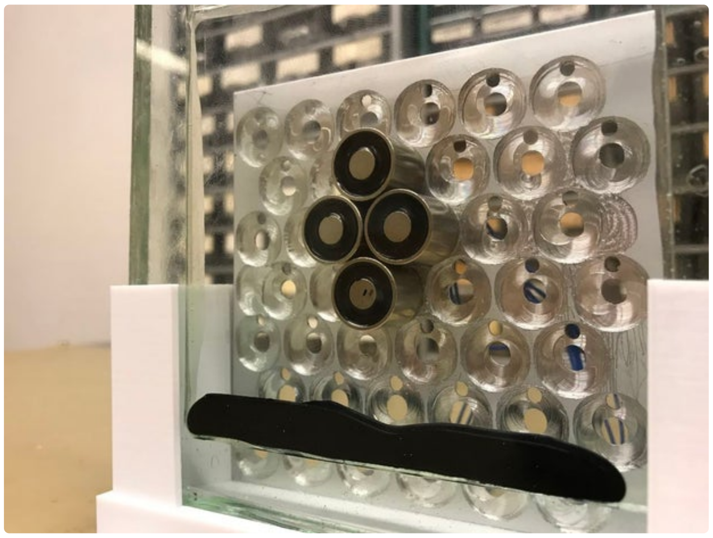
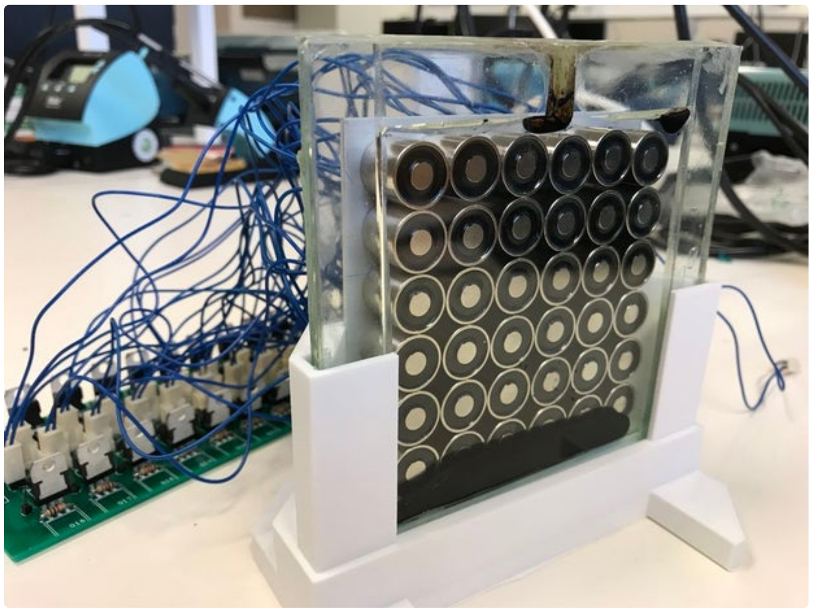
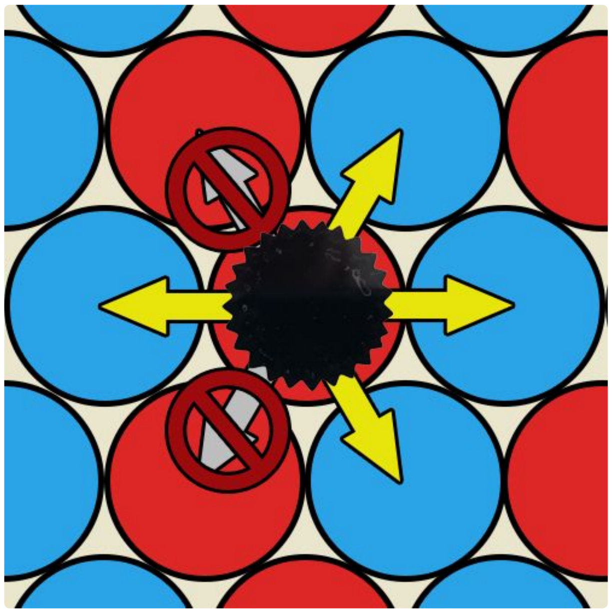
Material of the mount

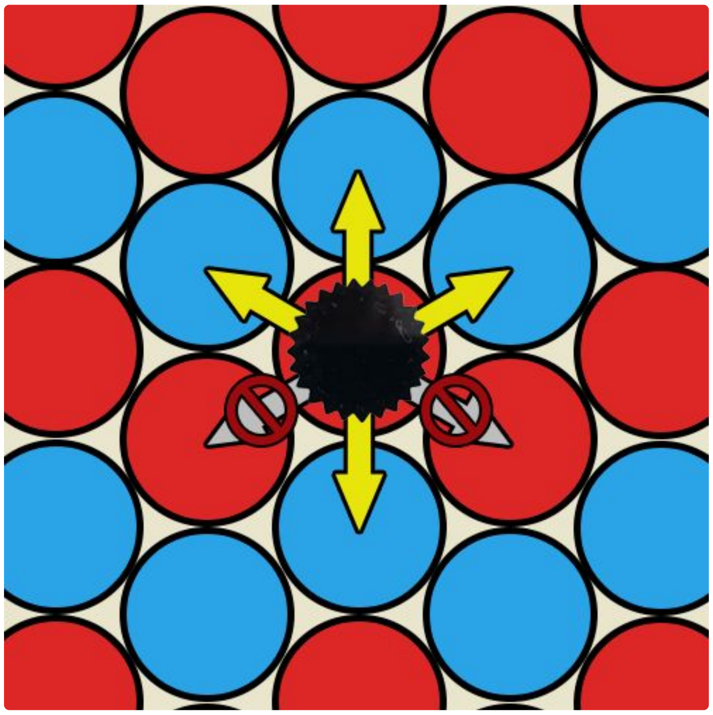
When choosing the material we would mount the magnets to we did not know exactly how hot these magnets would become. Our concern was that a magnet in the middle of the grid would be surrounded by so many other hot magnets that the epoxy could start to melt or even worse, something could catch fire. We therefore wanted to mount the magnets to something metal, which would act as a heat-sink. We actually considered buying a custom made heat sink that we could mount the magnets directly to with screws, but decided not to do it in the end because of budgetary restrictions. The budget actually also stopped us from getting access to a high quality CNC router where we could make the final sized mount, so we ended up using two pieces of laser cut acrylic instead, which turned out to work just fine. The magnets do get somewhat hot, but not incredibly. And were currently running only on passive cooling, no fans or anything. That can always be added in the future if we feel the need for it.

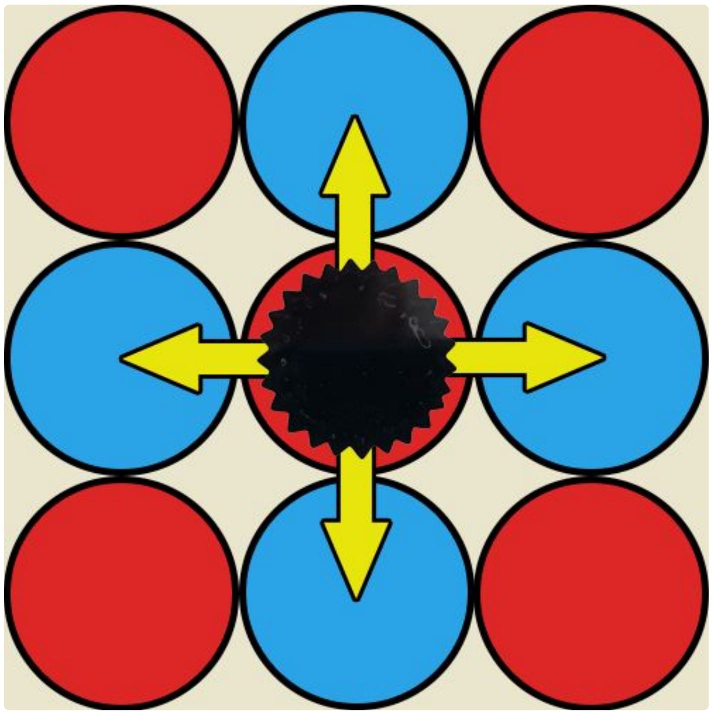
Accessibility and ease to replace

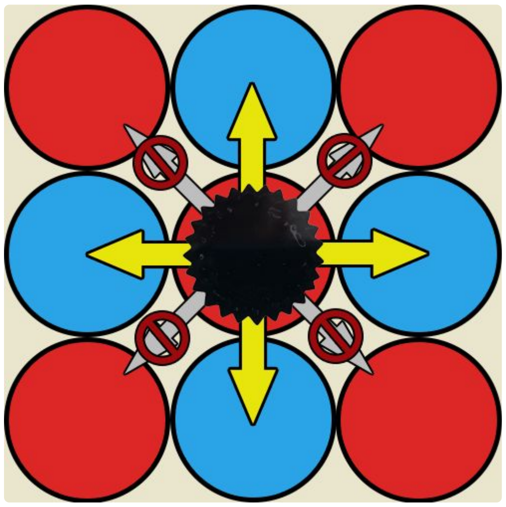
Another thing that can be done in the future is to get the aluminium plate CNC'd, therefore we've attempted to make the laser cut stencil in such a way that the magnets can be removed from it without cutting off the connectors we've mounted to the cables. Because mounting those was probably the largest time-consumer of any manual labor we had to do during this project.

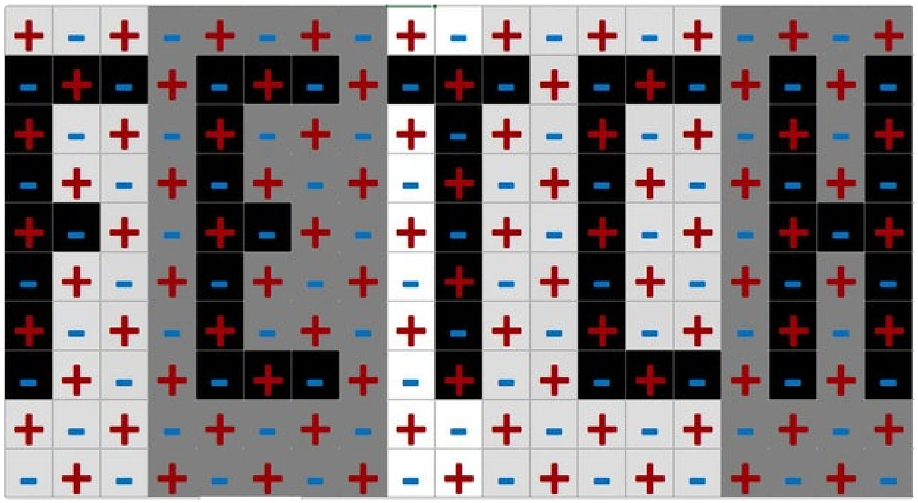


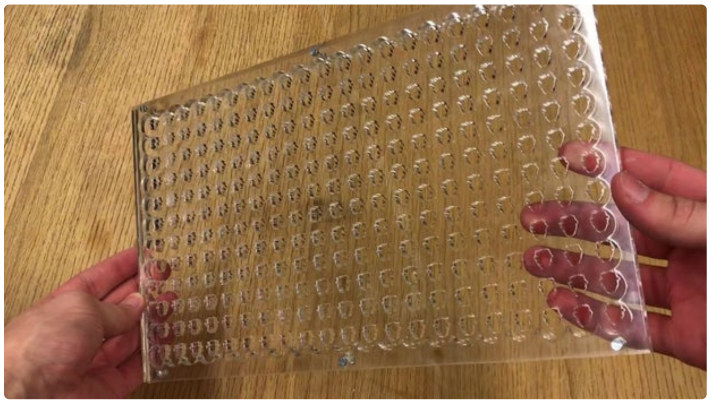




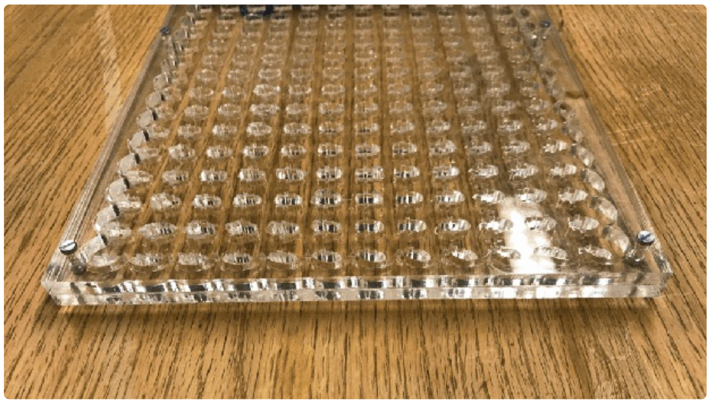










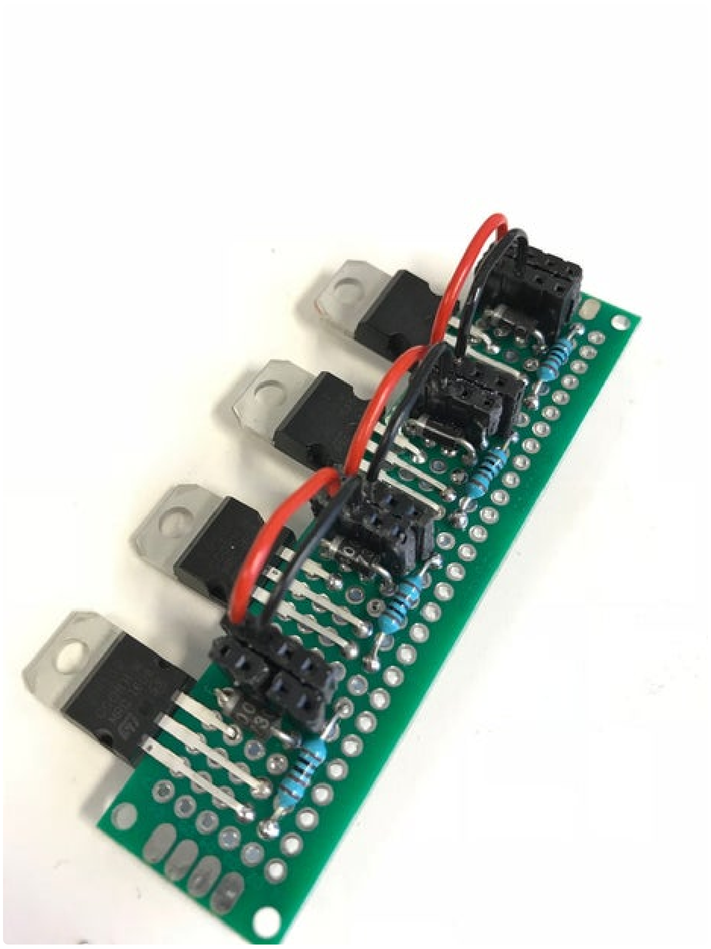


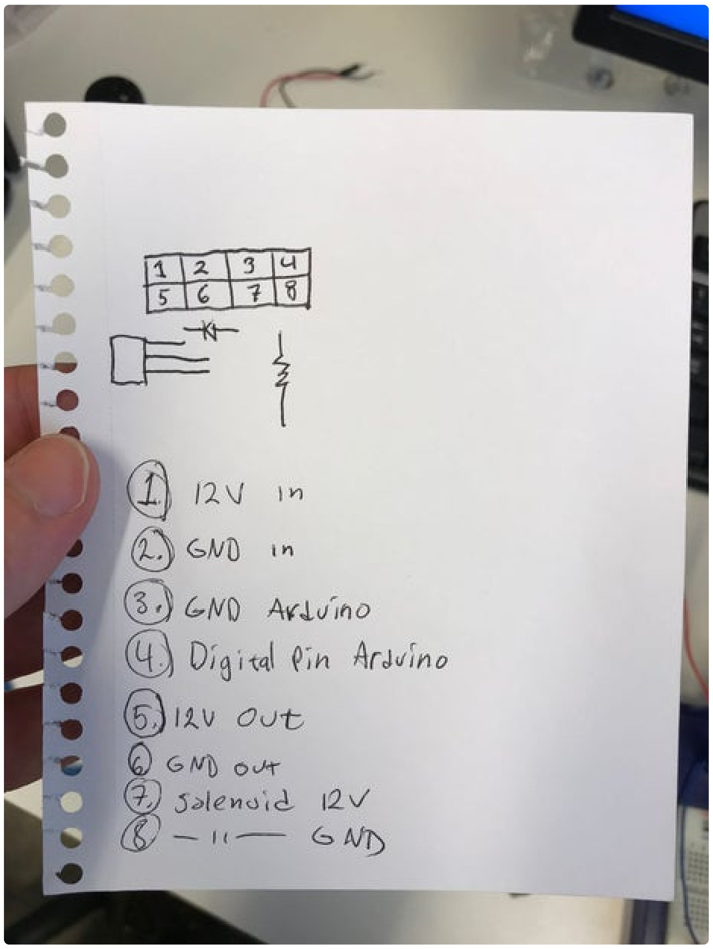
Step 4: Power Distribution to the Magnets

Distributing enough power to each magnet, and doing that individually (as stated in the design requirements), is the biggest electrical challenge in a project like this. Since they're operating at 12 volts we're gonna have to be dealing with quite a bit of current. As the pictures here suggest, we clearly overspecced our first few prototypes in terms of necessary heat dissipation - using huge clunky components. In retrospect it is obvious that these components would never have been able to fit within the form factor we were aiming for. They did however provide a great starting point for us, they were simple to solder and deal with, and definitively provide some value.

In the end we replaced all the clunky big transistors with a few tiny little ULN2004N solenoid-driver ICs. We have noticed in retrospect that these are not even the best choice within that series of the specific manufacturer we used, but they do the job well enough.

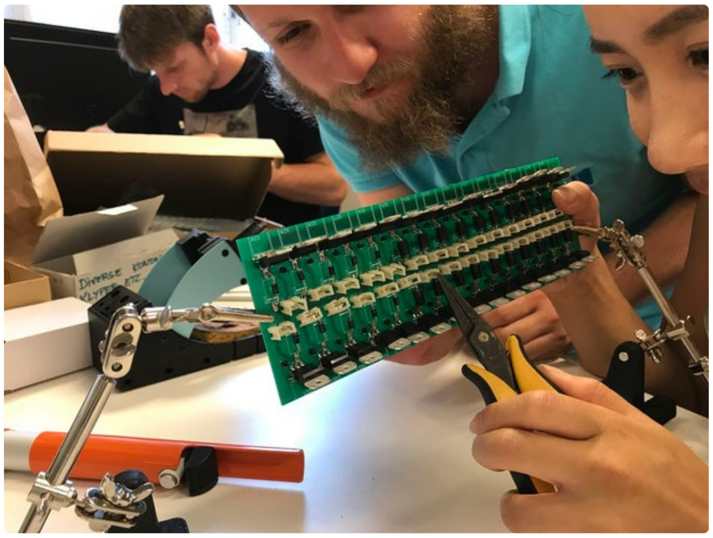
We are currently working on a redesign of the electronics, and will hopefully have them done within a few months, but with a pandemic ravaging the globe, who really knows...? The next iteration will tackle some big problems like size, fit, and cable management. It will also tackle some smaller problems by generally improving the capabilities of Fetch. These will be discussed in the final step of this Instructable.

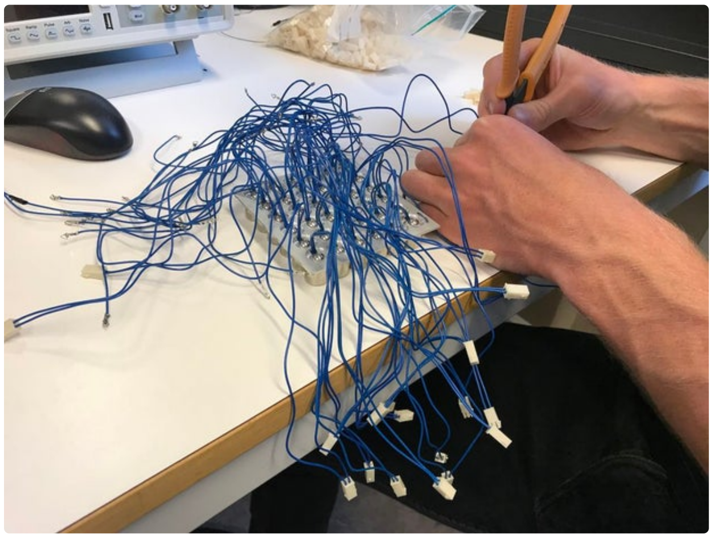
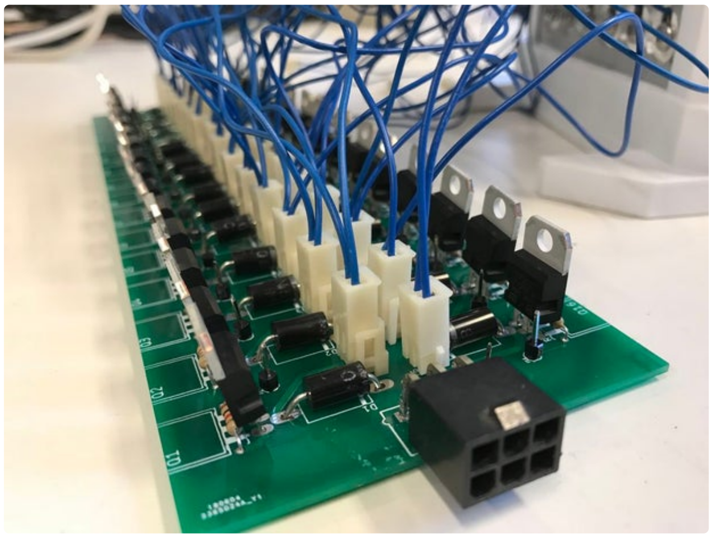
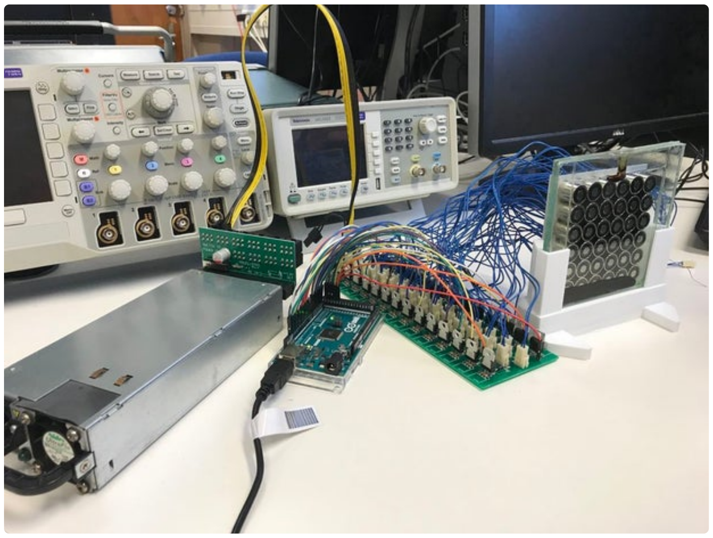
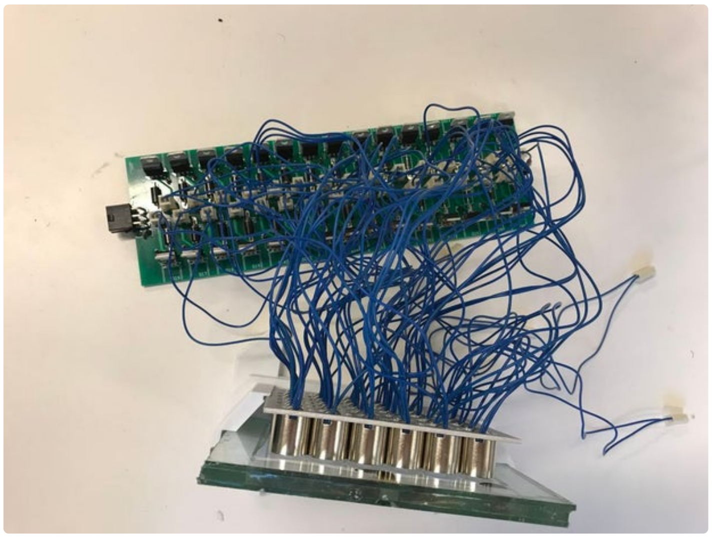
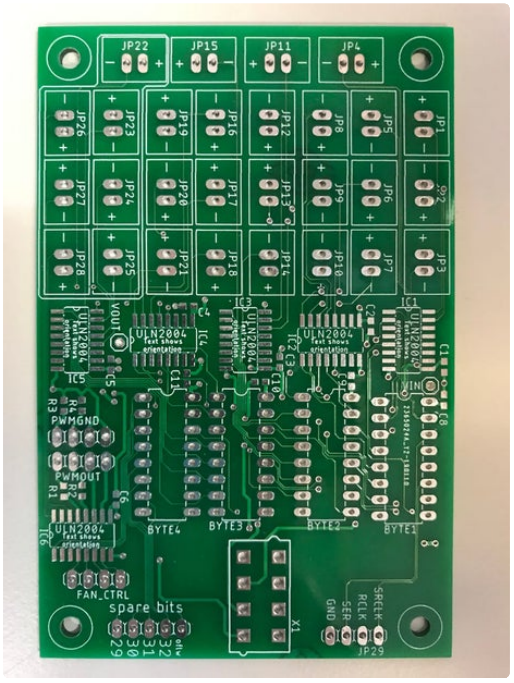
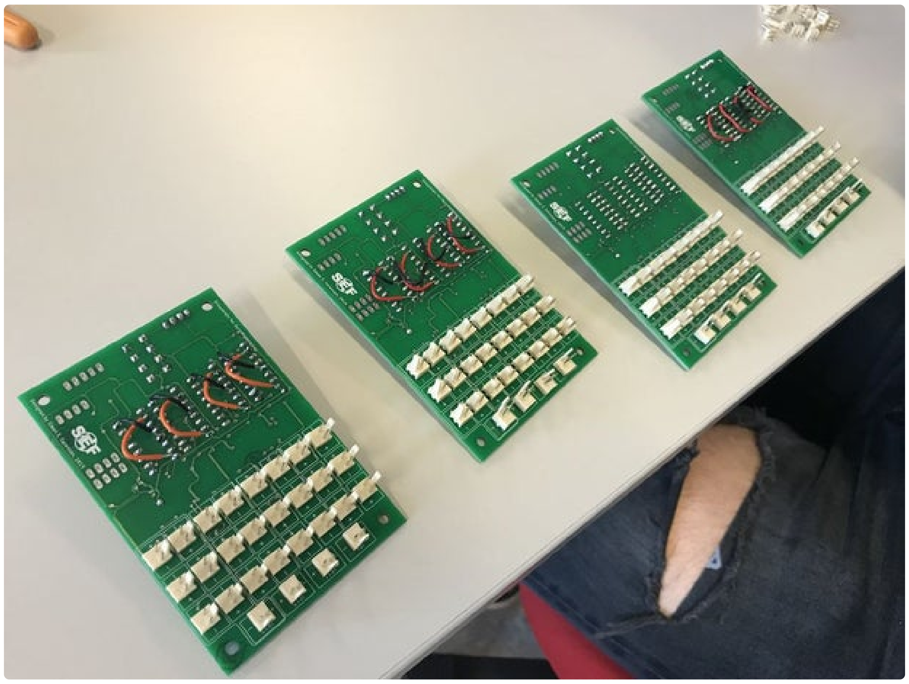


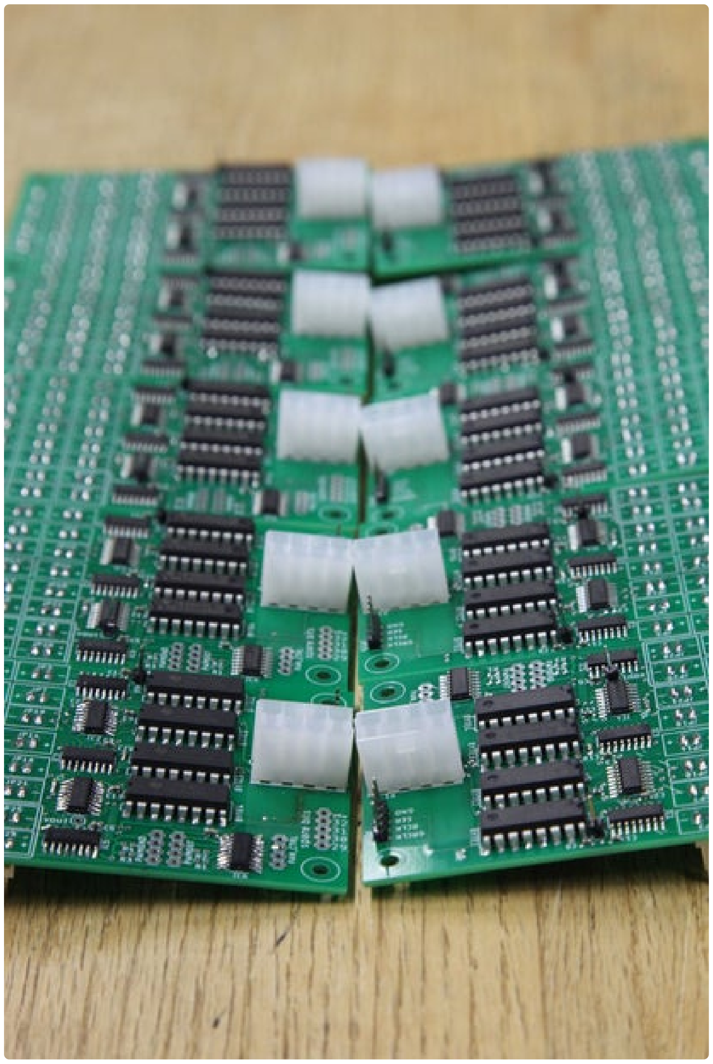
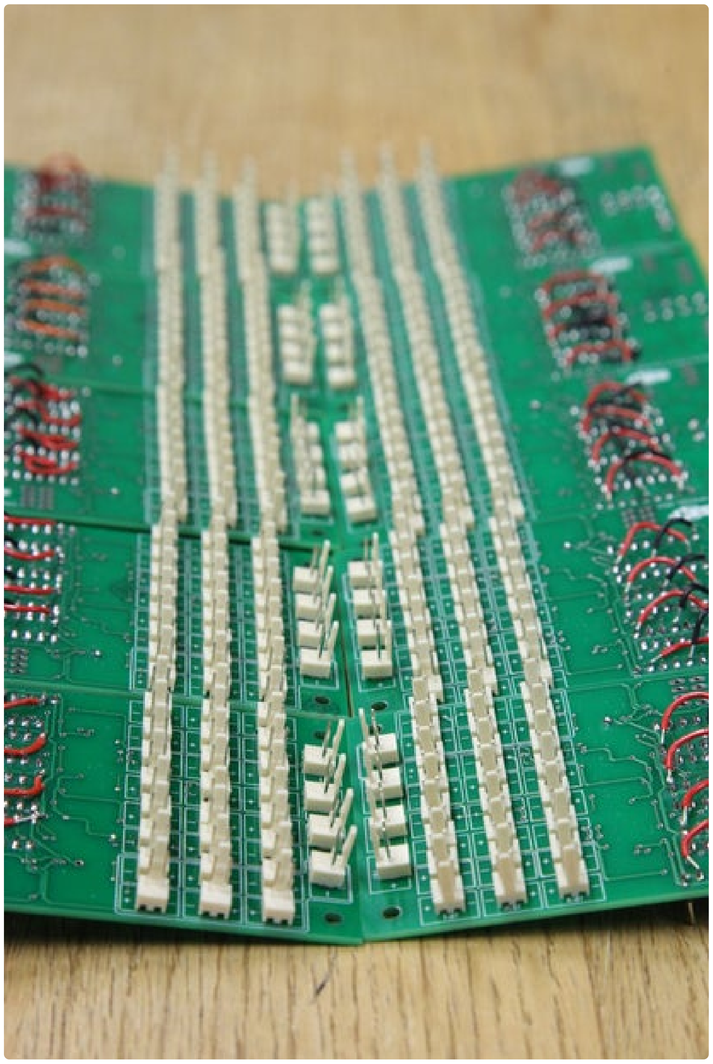


1	2	3	4
5	6	7	8

1. 12V in
2. GND in
3. GND Arduino
4. Digital Pin Arduino
5. 12V Out
6. GND out
7. Solenoid 12V
8. — 〃 — GND








JP22
JP15
JP11
JP4
ULN2004
PWMGND
PWMOUT
BYTE4
BYTE3
BYTE2
BYTE1
FAN_CTRL
spare bits
X1
SRCLK
RCLK
SER
GND
JP29


Step 5: Assembly

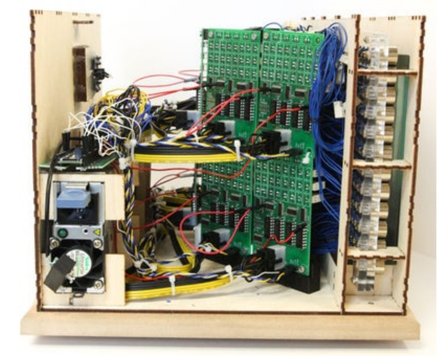
As a few other places, the assembly is also showing signs of a low budget. We really wanted a metal finish, and hope to be able to do that in a future version, but laser cut plywood also does the job. We aren't product designers, not even mechanical engineers, so while the design itself is a little rough arount the edges, we are generally quite happy about how it works.

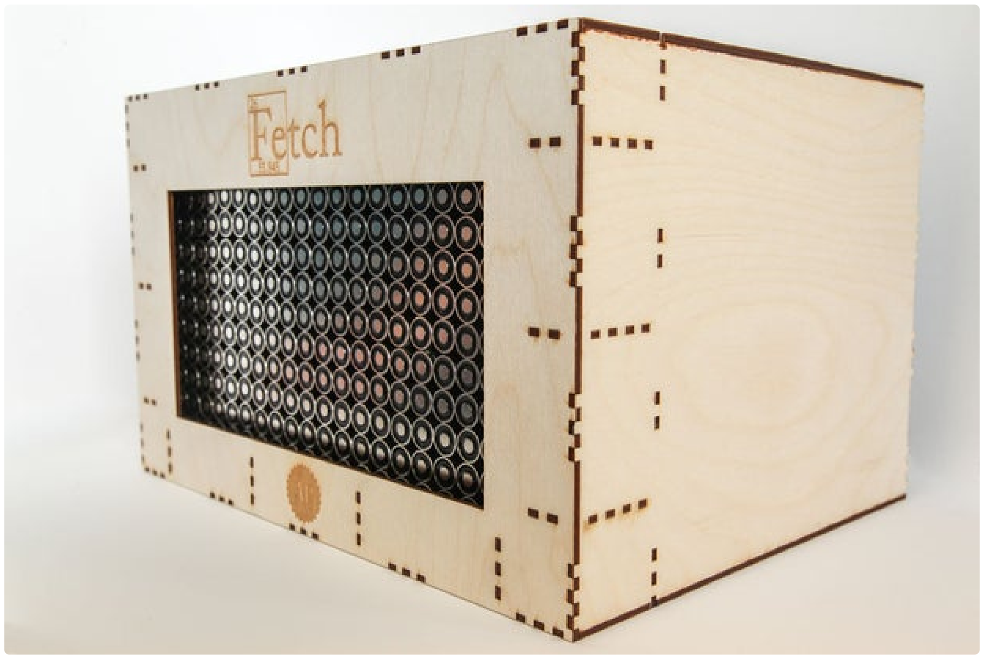
The biggest design problem is that the tank is not really properly secured by the enclosure. If the front-facing wall of the enclosure was to fall off, the tank would follow. We therefore sealed all joints with wood glue in an effort to prevent this from happening.

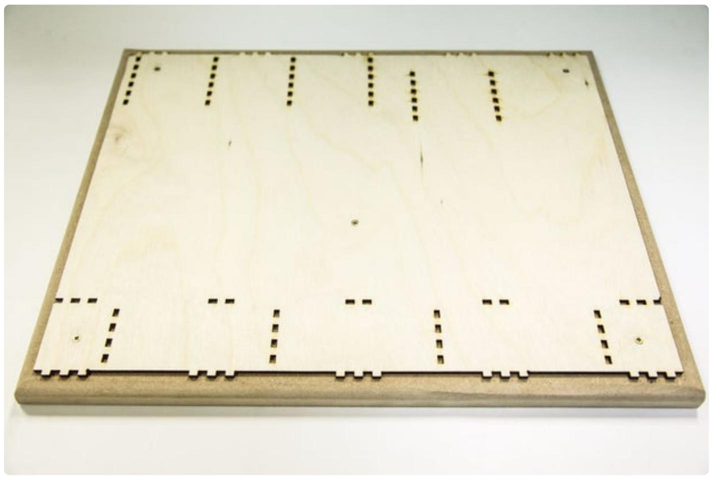
Another problem was the assembly process itself. I guess mechanical engineers learn this early on, but for us a concept like: "Think about how the pieces will come together" is foreign, and did not occur before we were actually doing it. Safe to say, we suffered from it, but it did come together which was all that mattered to us!

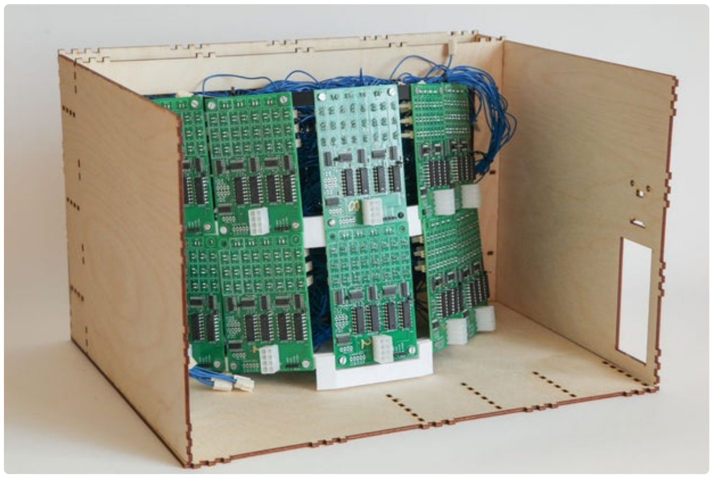
Wiring

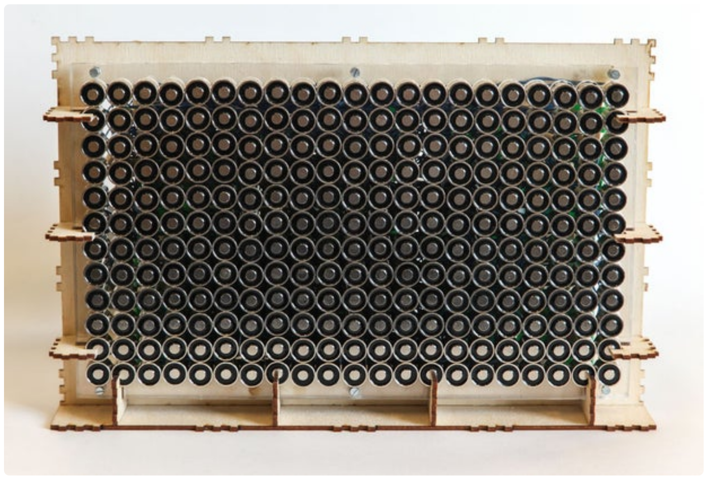
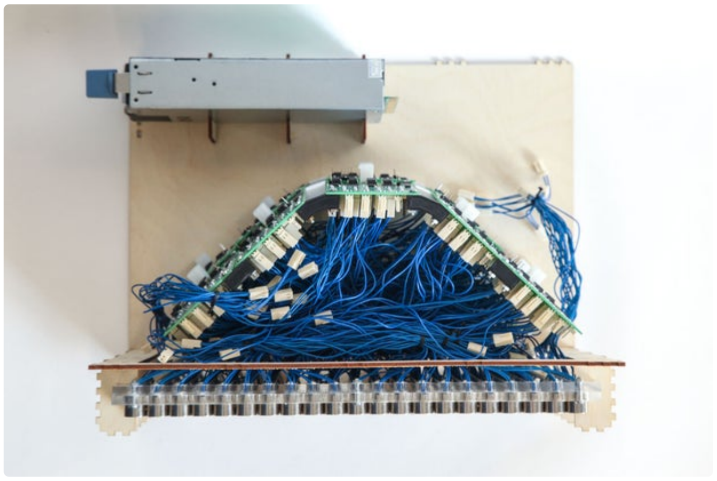
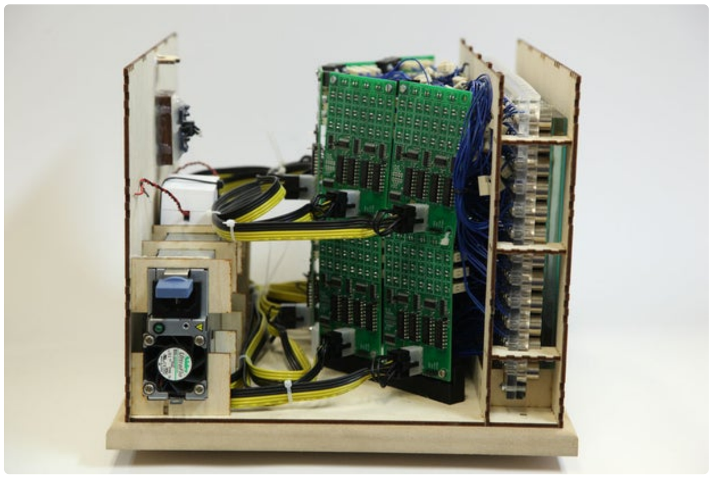
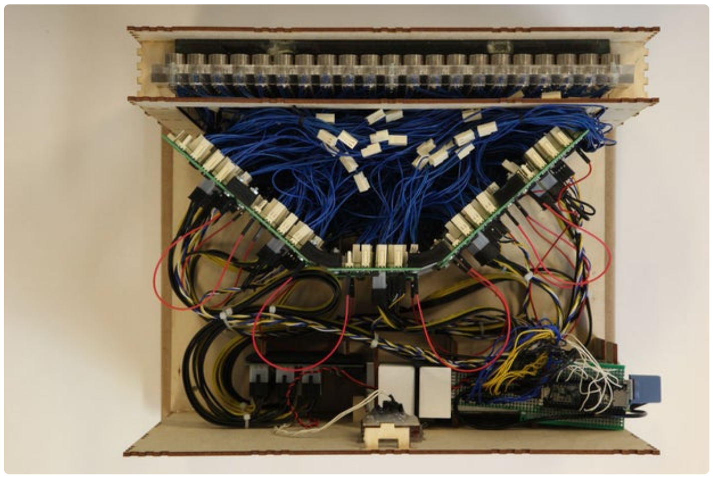
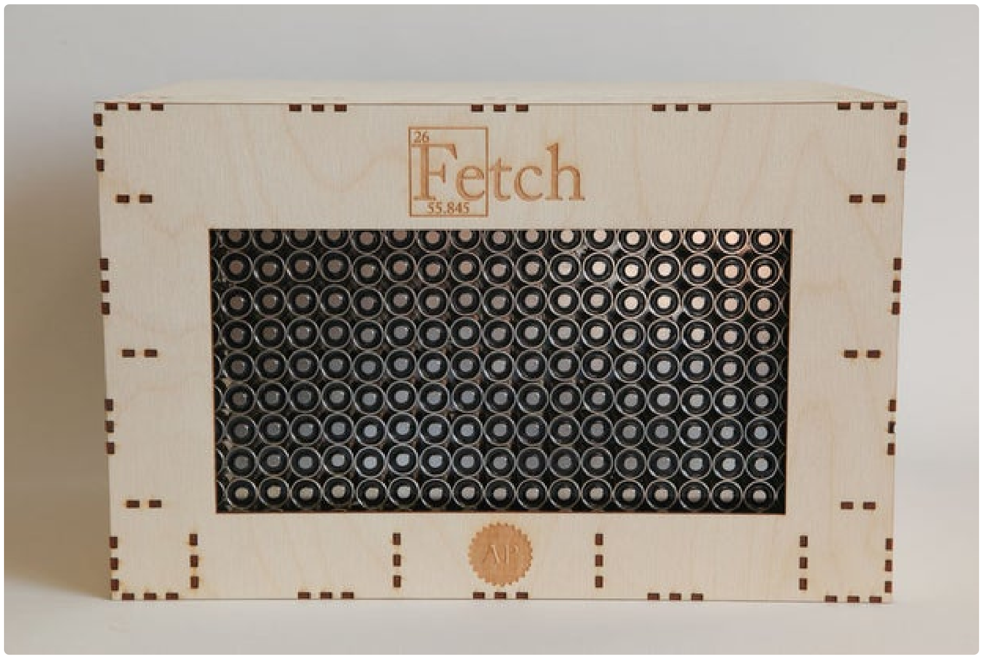
The power distribution PCB that attaches to our PSU has a mechanical on/off button on it that needs to be pressed after plugging in the cord. This led us to place the PSU in such an orientation that this button could be reached through a hole with a stick (basically a button extender). In the end we realized that it would be a lot easier to solder a wired button in parallel to the button we wanted to press, and mounting the extended button on the casing. As a result of this late change, the power cables are now pushing against the back wall of our case even though they don't need to.









Step 6: Animating

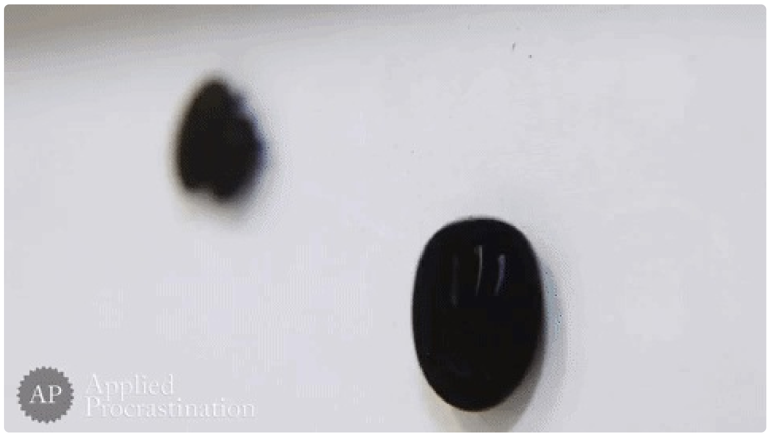
We were initially planning to make our own animation program that suited our needs. But we quickly discovered that there are already some very good programs out there that are specifically designed for something else, but still perfectly suits our needs. One such program is Aseprite. It's one of the go-to editors for pixel-art artists, has a strong community and scripting possibilities. Additionally, the development is pretty transparent, so even though the license is not fully open source, all code is available on github and it's free to use if you can compile it yourself. They even provide the instructions on how to do so. There are a few positives from their semi-open approach:

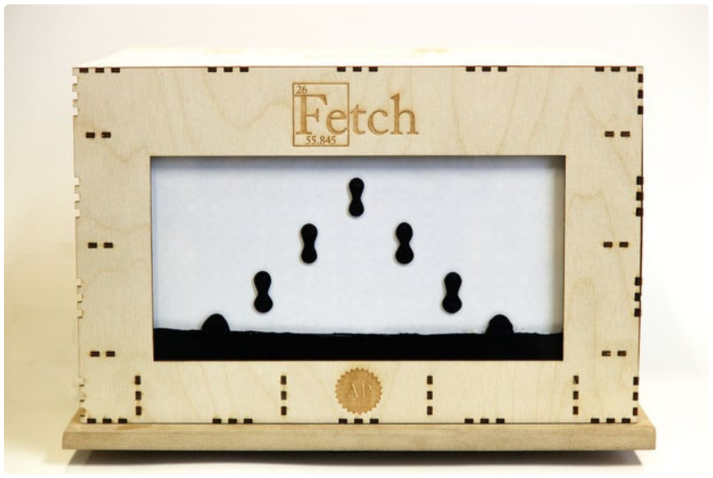
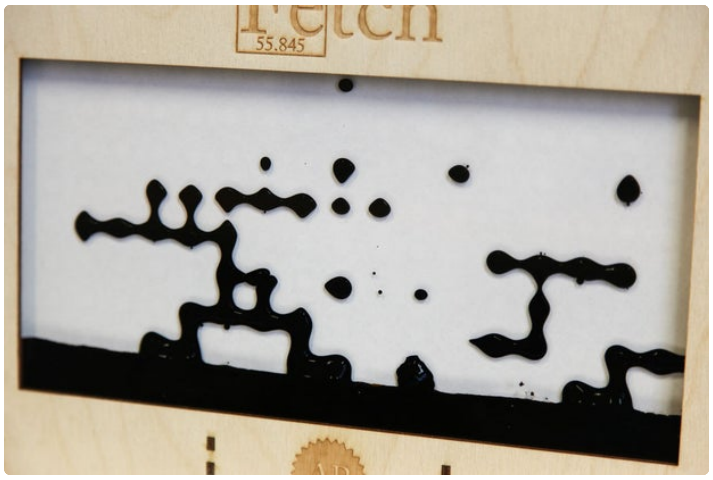
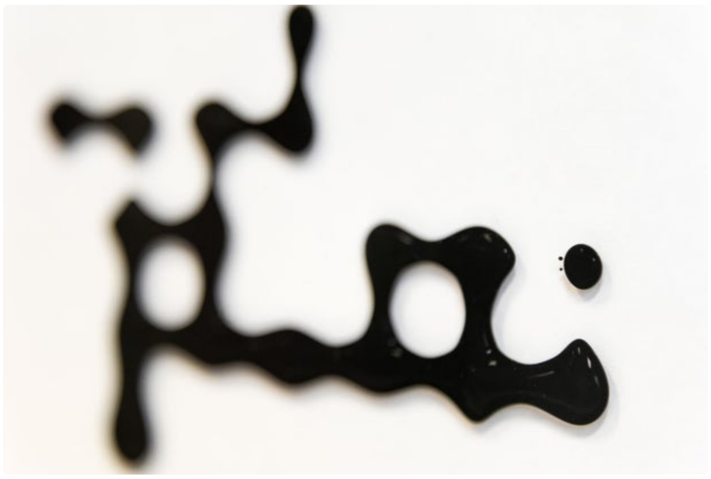
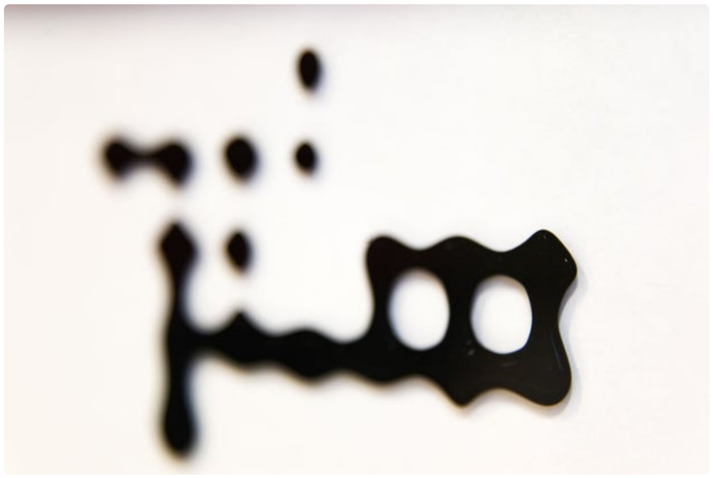
1. There are well developed APIs for common programming languages. Most important to us is knowing the internal structure of their filetype ".aseprite"
2. The owners of the program has funding to actively develop new features and fix bugs, meaning the program gets better by the day. I encourage you to pay for the program or donate even if you consider yourself capable of compiling the source yourself. We did so after testing it out for a while and making sure we wanted to continue using it.

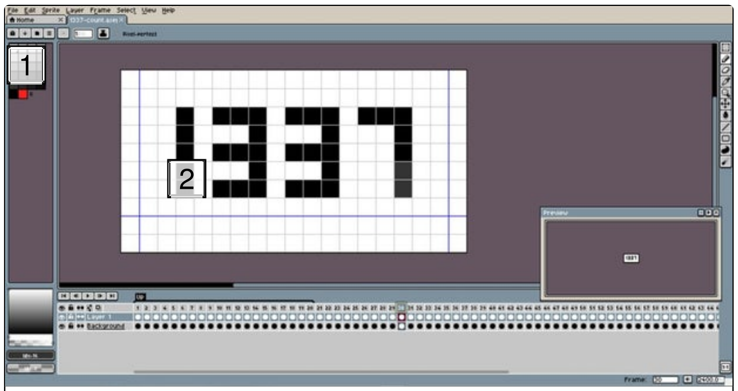
Our workflow is currently based on an external python script that converts the .aseprite files to a two-part file type that we've developed ourselves and found convenient to use. The two parts are a data part and a header part. The data is stored in binary, and the header is stored in a human readable ASCII format for easy configuration and changing how an animation is displayed. We will update our own filetype in the future, but not before we're certain that we're satisfied with the amount of data contained in them.

These "Fetch specific" files are then stored on an SD card and inserted in the system. A useful feature we want to add in the future is remote upload via for example Serial, Bluetooth or WiFi.

The ferrofluid display itself currently runs on a Teensy 3.6 with this open-source firmware we've written. The most urgent changes we're planning to make in the firmware involves holding more information in the Animation and Frame classes (like origin points and current location on the screen) to make programming games and dynamic animations easier. Right now we're pretty much stuck to showing pre-made animations. These changes will also need to be reflected in our file-type.




26
Fetch
55.845
AP

Fetch
55.845





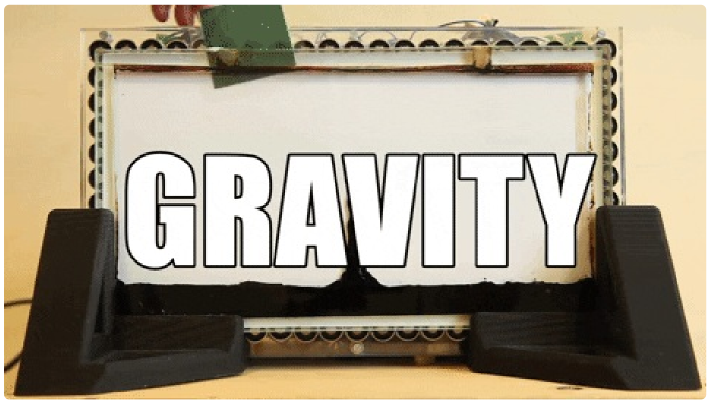
1. We're using colors to encode the duty cycle of the PWM. Here we have a resolution of 20 values an "off the charts" value (red) that's used as reference when drawing the animations but ignored by our parser.
2. The two bottom pixels are colored differently because we want less holding force here. Gravity does more than enough to fill these pixels since they have a straight line above them.

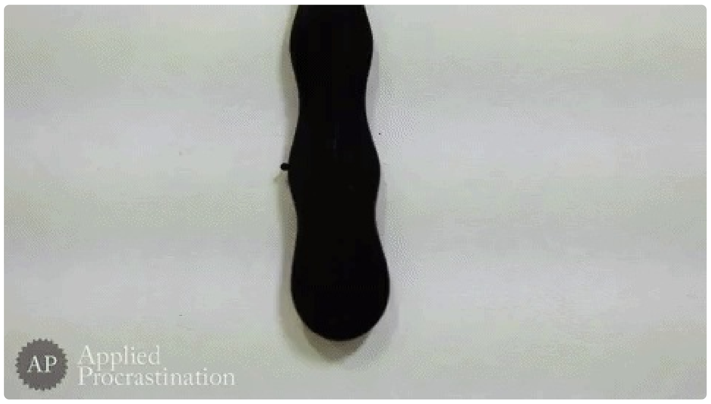
Step 7: PWM - Pulse Width Modulation

We stated earlier that we want our display to be completely vertical, and that we don't want to "cheat" by using a mirror at an angle. This means that we have to deal with gravity. Any column of magnets that are "on" at the same time will have a connection of ferrofluid between them, allowing the ferrofluid to "fall" towards the bottom pixel. The way we've counteracted this is by implementing Pulse Width Modulation in software giving us the option of applying less force on certain magnets. We use this to give less force on magnets that already get help from gravity as described above, and max force to the magnets above them. This has certain limitations though, most notably that the electromagnets we're using don't have much overhead to begin with - so the dynamic range between the largest and the smallest blob we are able to make is small.

We do not currently have any specialized hardware for changing the duty cycle of a given pixel, which means that we have had to write PWM in software. This turned out to be a challenge to do on the Arduino Mega we used during prototyping, but after optimizing the shift-out sequence to our shift registers and upgrading to a faster microcontroller it works very well. That being said, we are currently developing our next revision of circuit boards for the control of electromagnets with specific PWM hardware, making the speed increase introduced by the Teensy reduntant. Upgrading the microcontroller to a Teensy 3.6 turned out to be a necessary change anyway though, because with the Arduino we didn't have a way to store animations in our system. The Teensy 3.6 has a built in SD-card slot which has already proved itself very useful.

https://youtu.be/ZXVIQhwBu1w

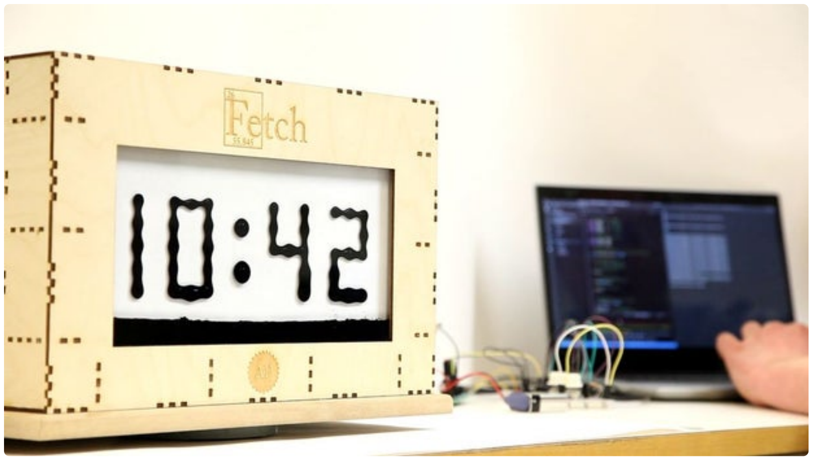


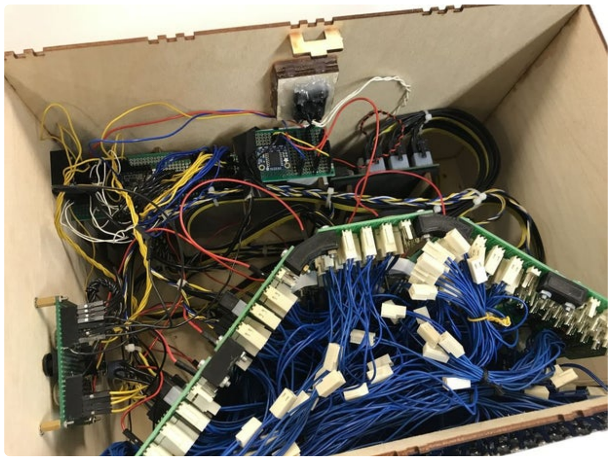


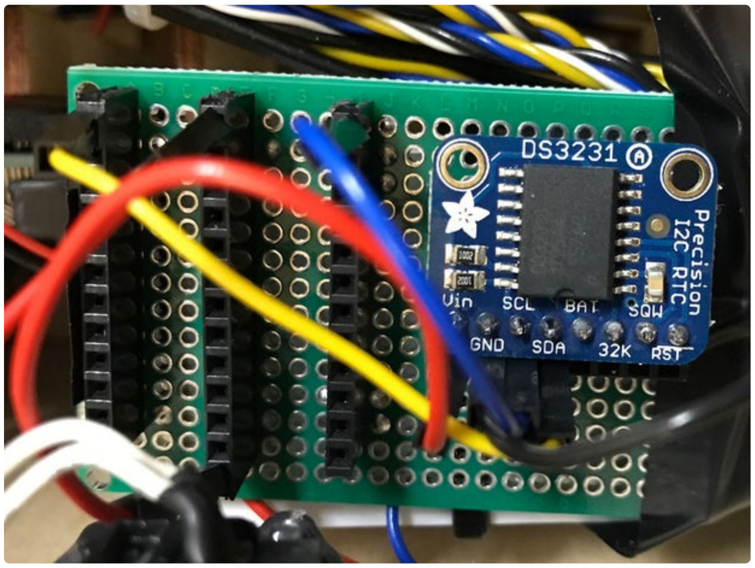
Step 8: RTC Module for Clock Functionality

To make the clock capable of keeping time we've added a DS3231 precision "Real Time Clock" (RTC) from Adafruit. After some minor code adjustments, we now have a working "clock mode" on our display. The implementation is pretty straight forward at the time, but we plan to make it more robust. The way it currently works is that when the display is operating in "clock mode", it will show the time with one minute resolution. Every time a minute has passed, the display will load a new animation from the SD-card and display it.

The display already had a few different modes that were implemented to make life easier when showing our display off at conferences. We currently use an unmarked 4-way switch to change modes, which is obviously not a very good solution, but we will fix that very soon. Unless priorities are extremely willing, the fix will probably be tape and permanent marker...







Step 9: "Don't!"

When we started out we had an idea that we could use a two-glass window for the tank. If that worked we would have an easy way to acquire high precision and durable tanks. We managed to get hold of a full size window for free, and even though it wasn't the correct format we could use it to test whether it worked at all. So what we did was to first drill a hole along the seal, then fill it with water. Since that worked and it didn't leak over the next couple of days, we changed out the water with vinegar. This is a trick used in the Instructable about ferrofluid that has been linked several times above which is supposed to clean the surface of the glass so ferrofluid won't stick. What we instead saw was that these windows have a plastic film on the inside of the glass which started to disintegrate after a day or two. But more importantly, it made us realize that the pieces of glass are separated by an aluminum border. And the vinegar helped it rust... So, don't use windows as your tanks, and don't let any metal in your container except what's in the ferrofluid itself. Rusty displays don't look good.

Another "don't" that may sound obvious when we say it is: don't connect your powersupply the wrong way around. Unless you protect your circuitry against this, it will burn. Oops.

Step 10: Next Generation (V2 Circuit Board)

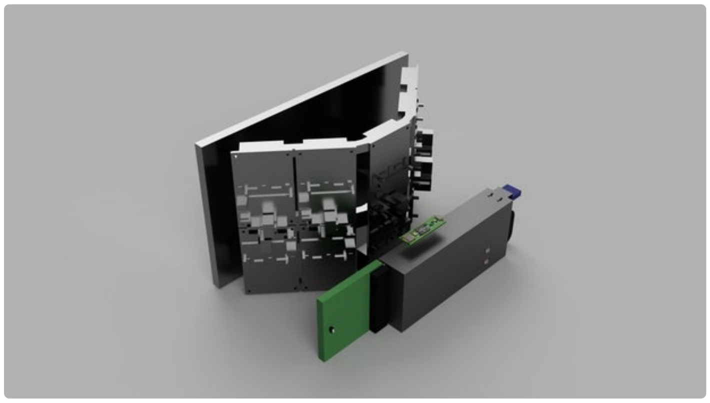
There are a few things that have been bothering us with our current solution for the electronics. One of the main problems were covered in the step about Pulse Width Modulation (PWM), which is the fact that we have to do PWM in software. In our next version we will implement PWM in hardware, offloading the main processor of this task.

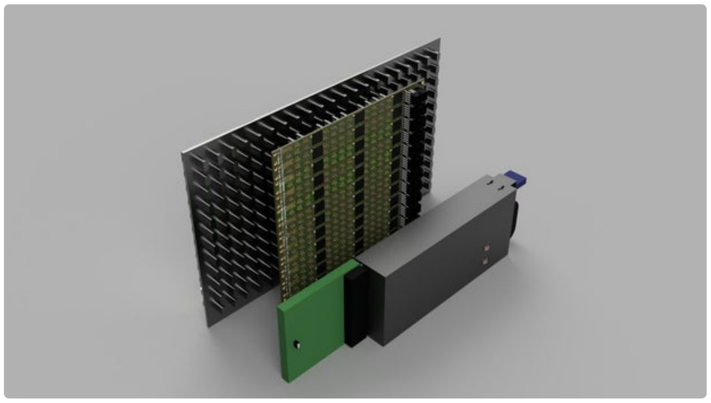
Secondly, we will have a single board per row, and we won't have any overhead on each board. This means that we will have 12 boards with 21 connections each, instead of 10 boards with 28 connections each.

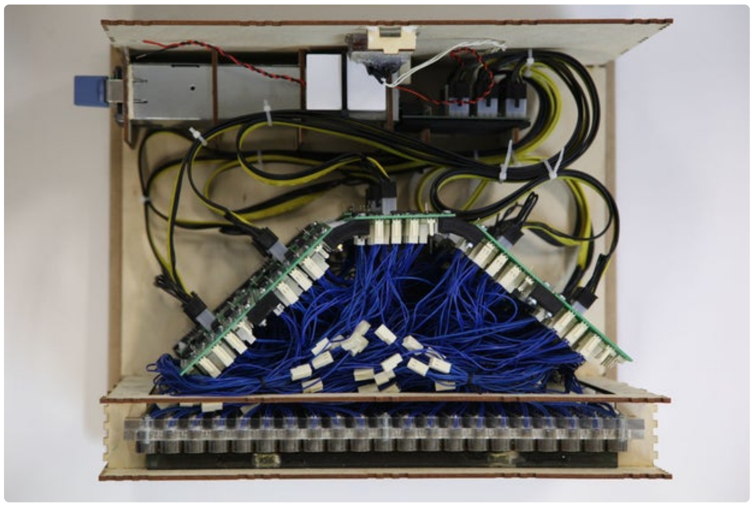
Another problem, which is the main reason for our bulky form factor, is that we forgot to account for the length of the wires from the electromagnets when designing the previous version of the boards. It's much easier to program fast code when each board controls its own row, and a board placed all the way to the left is too far away from the magnets on the right to be nice and flush. Therefore we have had to place our boards in sort of a "horizontal dome", taking up a lot of unnecessary space.

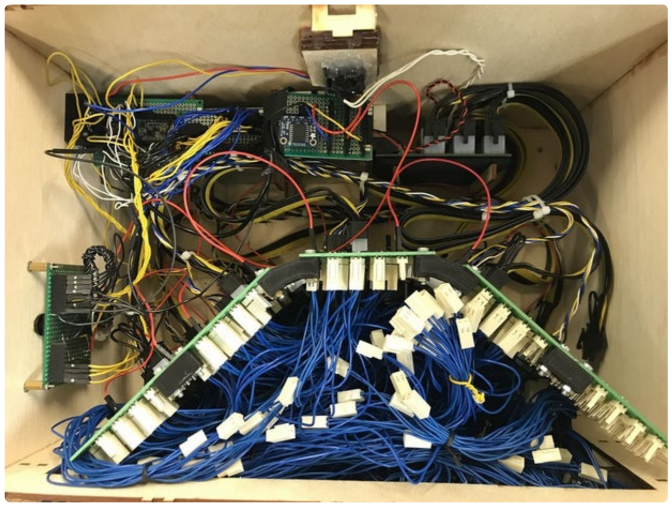
The final improvement we will do is to make sure we are able to daisy-chain our I2C-bus, drastically reducing the amount of loose wires inside the chassis.

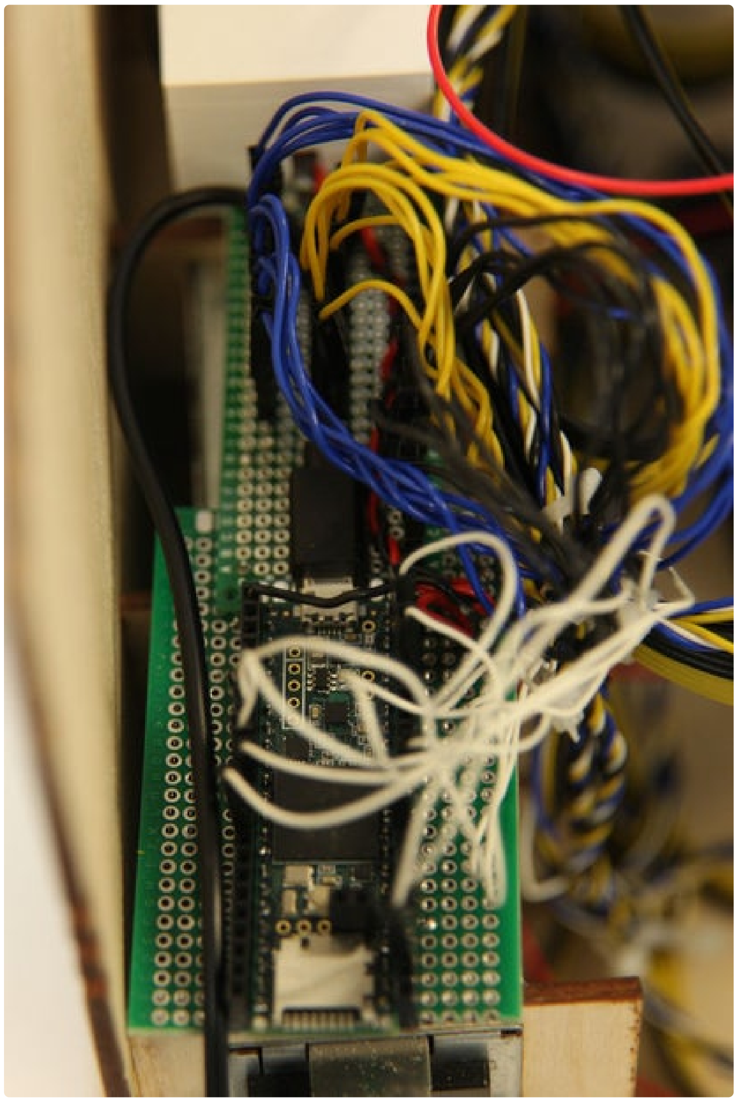
Ultimately, we would like to make a motherboard for all these row-controllers, eliminating the need for a whole bunch of wires. But such an improvement is unlikely to happen in a while, and will certainly not happen in our first revision of the electronics.











Ok, I'll admit it... I want the clock too.

But on the video game front, what about a single-screen platformer (blobformer?), in the vein of [Not-Yet-Super] Mario Bros? Another candidate (if less entertaining to watch) might be something like BSD's "robots" (or the early Mac "Daleks") - although I'm not sure quite how to differentiate the "player" blob from the "robot" blobs.

We have a theory that it will be possible for the player to remember which blob they're controlling because it will respond to their commands. It's obviously not the best UX, and it will have a learning curve, but with the technical limitations of this display it's the best we can do.

You mentioned 12V power, but how much current does each magnet require? The Jian website says 1.4W and also says an incredible voltage range, which don't really jive. Is it really 1.4W at 12V = 120mA per coil?

I think the voltage range is just written that way to indicate that they will make magnets according to customer specs, but the most common voltages for these magnets are 12V and 24V.

We have never bothered to actually measure the average current draw, because we are more concerned about the peak current draw and didn't have equipment suited for measuring that. Therefore we instead went for a solid overspecification of the power supply (1200W to the theoretical ~400W needed). This "trade-off" wasn't really a trade-off because it literally had zero downsides compared to for example a 500W ATX PSU for a "regular" desktop computer. The formfactor of the 1200W PSU was better for our purpose, the price was better, the power rating was better, and it was more "hackable".

With all this said, if you already have a 500W PSU lying about and want to use it for powering magnets, it will probably work for this specific use case. Keep in mind that there will never be a real use-case where all magnets are being used at once. The only thing that could make that happen would be a programming or user error (so, I'm not saying it won't ever happen).

Wow. Just... wow!

Very impressive. The desktop version will make you rich. I really appreciate that you took the time to explain your thinking...as important as the brilliant idea. I have been waiting a long time for someone to come up with a good reason for ferrofluid.

Nice job on your project

I have a project running in this Clocks contest, but I voted for yours. This is an amazing and well elaborated project. I would love to see some kind of (inverted) Tetris running in this display.

Love your Fetch-A-Sketch! Absolutely brilliant and really well executed and explained. Are you sure that fluid isn't actually alive?

a very cool idea for a clock! but i dont think i could make it

Great work. Well done. Excellent idea.

The aluminium border of the double glass contains zeolite particles to keep the glas fog free(dew point below -60grC) when it is cold outside. Zeolite is a clay with lots of active aDsorption site. This react with the vinegar, hence the reaction you saw. Nice project BTW, and I love the term "Applied Procrastination"

I must have missed this reading through this, but how much Ferrofluid did you use, as well as where you got it?

We bought 2 x 250ml bottles of EF-H1 by Ferrotec from this supplier: https://www.czferro.com/products/ferrofluid-bulk but that has proven to be WAY too much :) I think we have used roughly half of one 250ml bottle (maybe a bit more) but that includes a lot of experimentation and goofing around. 60ml would probably have been more than enough for the display alone.

I completely forgot to mention that we use EF-H1, although it is mentioned in the Instructable we link to about handling the ferrofluid itself. I've added a link to the supplier we used in the "supply" section now, thanks for letting me know!

Thank you for your reply, and thank you for updating.

THATS SO SMART-- also, the display will last forever ish (the bulbs don't burn out

I love it - so interesting looking :D