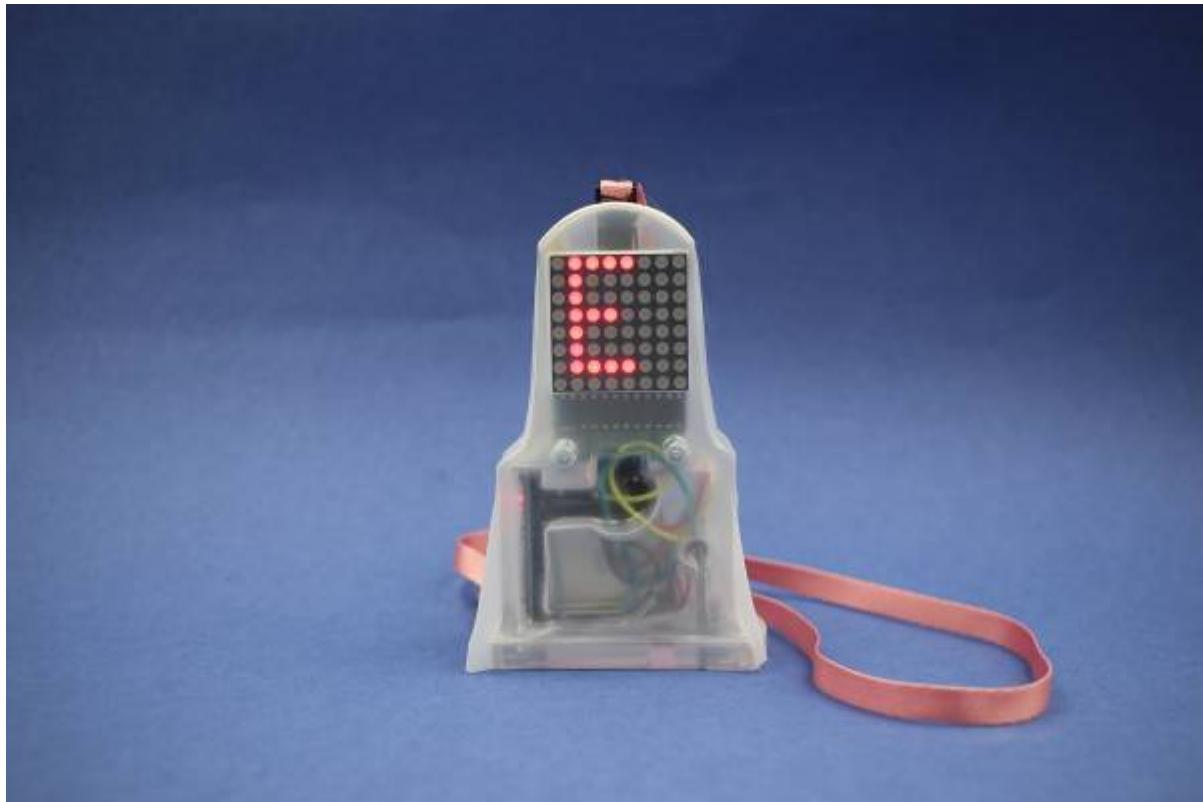


Badgeduino

SLQ Wiki Fabrication Lab 2026/01/18 00:50

Badgeduino

The Edge Jan 2018



Summary

Participants will create a wearable electronic badge. The badge has a programmable 8x8 LED Matrix display with a re-chargable battery, a laser-cut frame, with a weather resistant hand-made vacuum formed shell.

This workshop was initially developed for the Flying Arts program ArtizINNOVATE@theEdge in June 2017, then run at Woodford Folk Festival 2017/2018. This version was adapted to run in-house at SLQ The Edge in 2018.

What is wearable technology/wearables?

Wearables are smart electronic devices (electronic device with micro-controllers) that can be worn on the body as implants or accessories ¹⁾



Image by Pebble Technology [CC BY-SA 1.0 via Wikimedia Commons](#)

Wearable technology should be:

- robust
- reliable
- simple to operate
- small enough to wear

This workshop is designed to be an accessible introduction to electronics assembly.

we have chosen a simple design with:

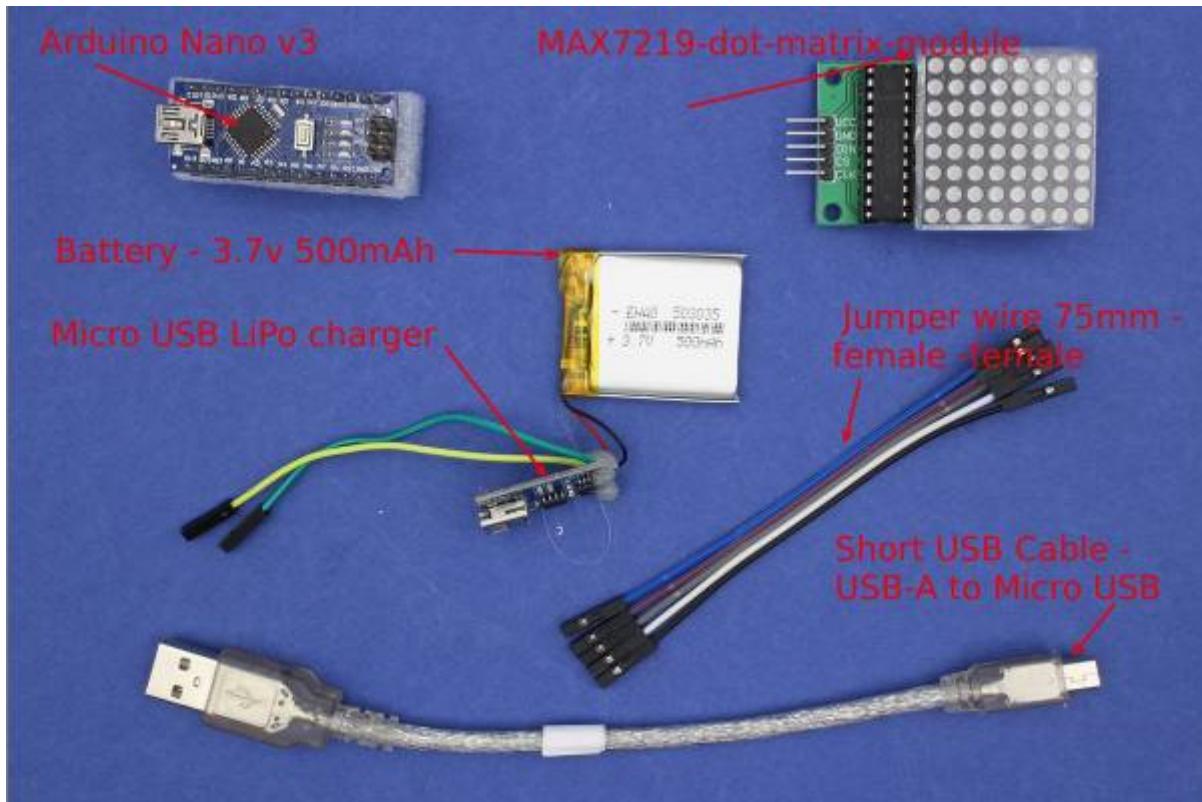
- large components
- no soldering
- robust enough to be taken apart
- can be vacuum formed over and over

Skills Introduced

- high level overview of circuit and components
- case assembly and wiring between components
- vacuum Forming
- coding in the Arduino IDE

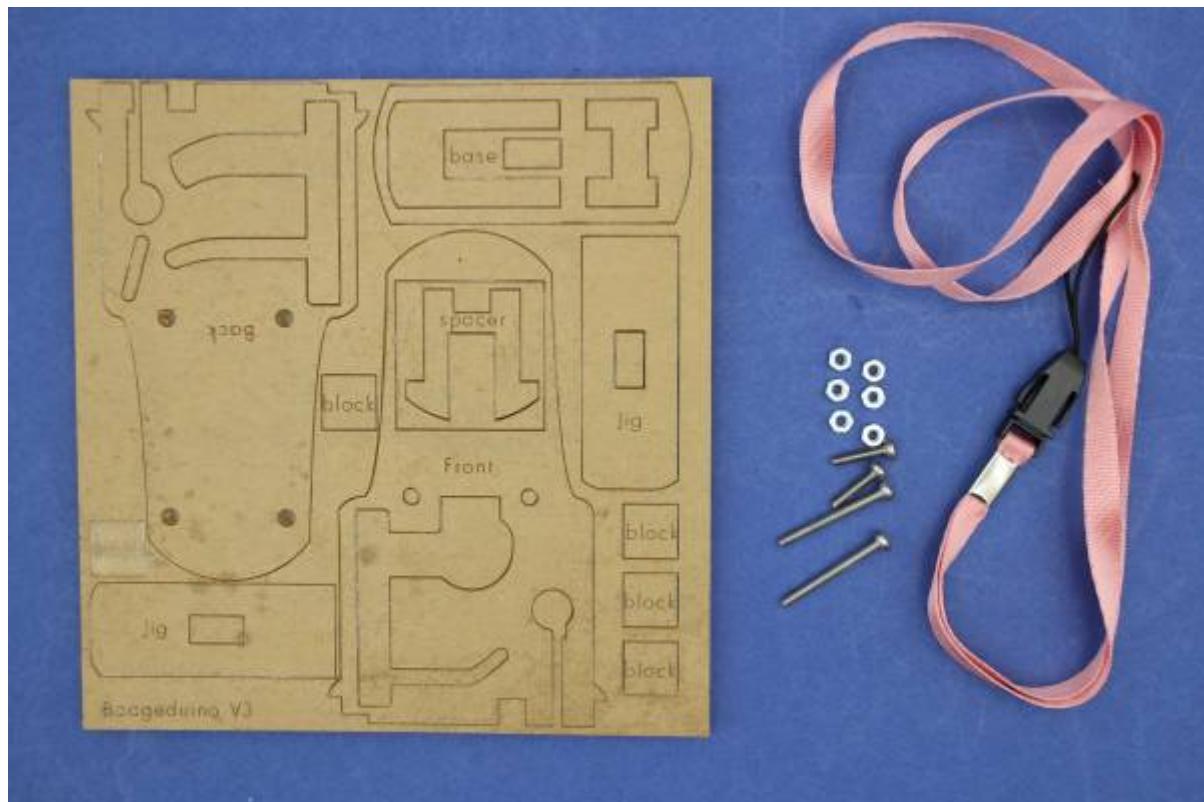
Materials

Electronics



Material	Quantity	Cost	Supplier
Arduino Nano v3	1	\$3.40	AliExpress
MAX7219-dot-matrix-module	1	\$2.10	AliExpress
Lithium Ion Polymer Battery - 3.7v 500mAh	1	\$5.70	AliExpress, \$5.70
Micro USB LiPo charger w/ Protection	1	\$0.8	AliExpress, \$0.45
Short USB Cable - USB-A to Micro USB	1	\$2	Tronixlabs
Jumper wire 75mm - female -female	5	\$0.14	Tronixlabs
	Total	\$14.14	

Case



Case	1	\$4	In-house
m2.5 X 12mm bolt	2	0.02	in-house
m2.5 x 25mm bolt	2	0.02	in-house and Shell
m2.5 nut	6	0.06	in-house
Attachment / Clip / Lanyard	1	\$0.22	AliExpress
TOTAL: \$4.32			

Shell



Material	Quantity	Cost	Supplier
Milk bottle sheets	2	\$?	In-house

Tools and Preparation

Tools

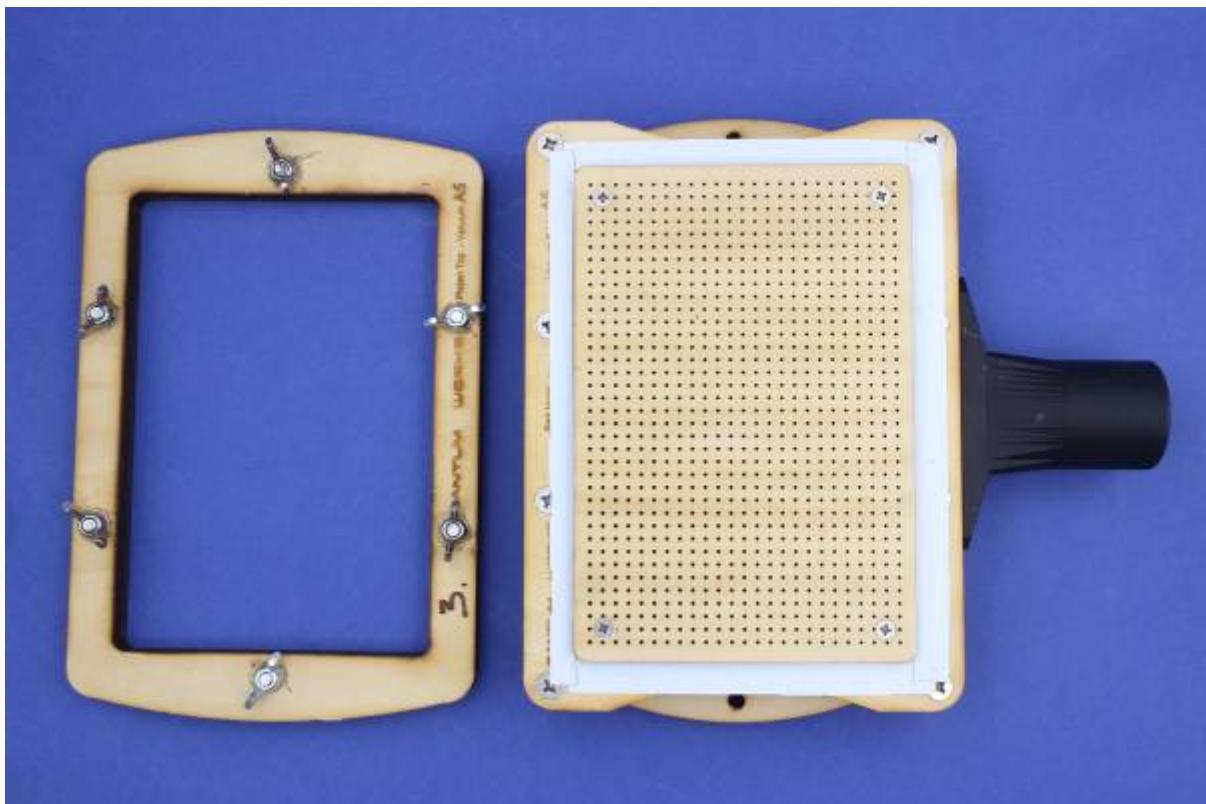
- computer with working USB 2.0 ports
- a working internet connection (or pre-downloaded files)



- craft knife
- scissors
- small Phillips screw driver
- small pliers
- side cutters



- hot glue gun
- hot air gun



- the Edge Vacuum former

Introduction to Arduino Microcontrollers

What is a micro controller?

A microcontroller (or MCU for microcontroller unit) is a small computer on a single integrated circuit.

- one or more CPUs (processor cores)
- memory
- programmable input/output peripherals
- can be mixed signal devices interacting with
 - digital signals
 - analog signals

Why use an MCU?

Microcontrollers are small, low powered and robust, making them perfect for [embedded systems](#) such as;

- medical devices
- remote controls
- office machines

- appliances
- power tools
- toys
- wearables!

What is Arduino?

Arduino is an open source computer hardware and software company, project, and user community ^{[2\)](#)}

- the hardware is based on the Atmel 8-bit AVR MCU
- the software uses the Processing IDE, with a simplified version of the C++ language.
- open source has led to the creation of a huge range of
 - clones,
 - compatible devices
 - peripherals
- a strong community means
 - “someone, somewhere has solved the problem”
 - we can run this workshop using and adapting existing resources.

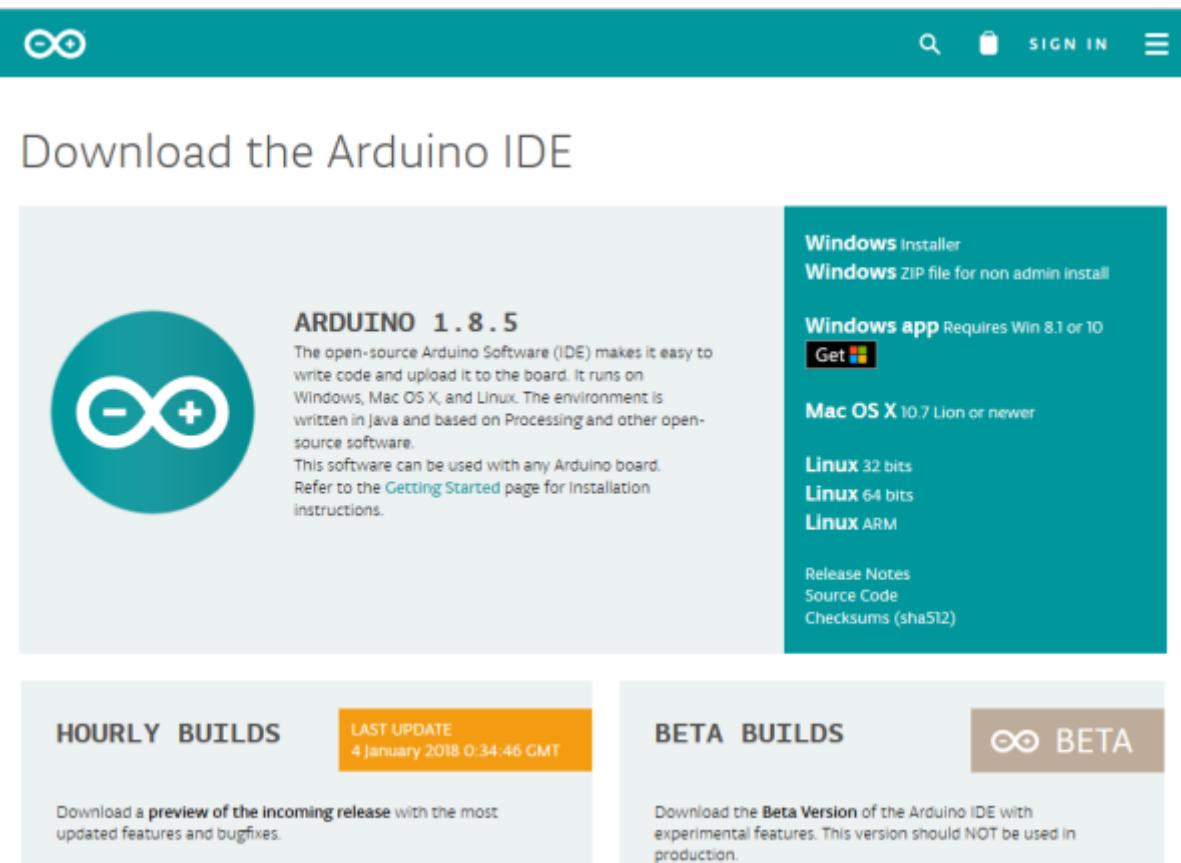
Assembly and Testing

Its time to get started!

Step One: Connecting and Testing the Arduino Nano

Download the Arduino IDE

Download the latest version of the [Arduino IDE](#).



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

HOURLY BUILDS LAST UPDATE 4 January 2018 0:34:46 GMT

Download a [preview](#) of the incoming release with the most updated features and bugfixes.

BETA BUILDS **BETA**

Download the [Beta Version](#) of the Arduino IDE with experimental features. This version should NOT be used in production.

Select your operating system and when prompted choose “Just Download”.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **21,306,305** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 **\$5** **\$10** **\$25** **\$50** **OTHER**

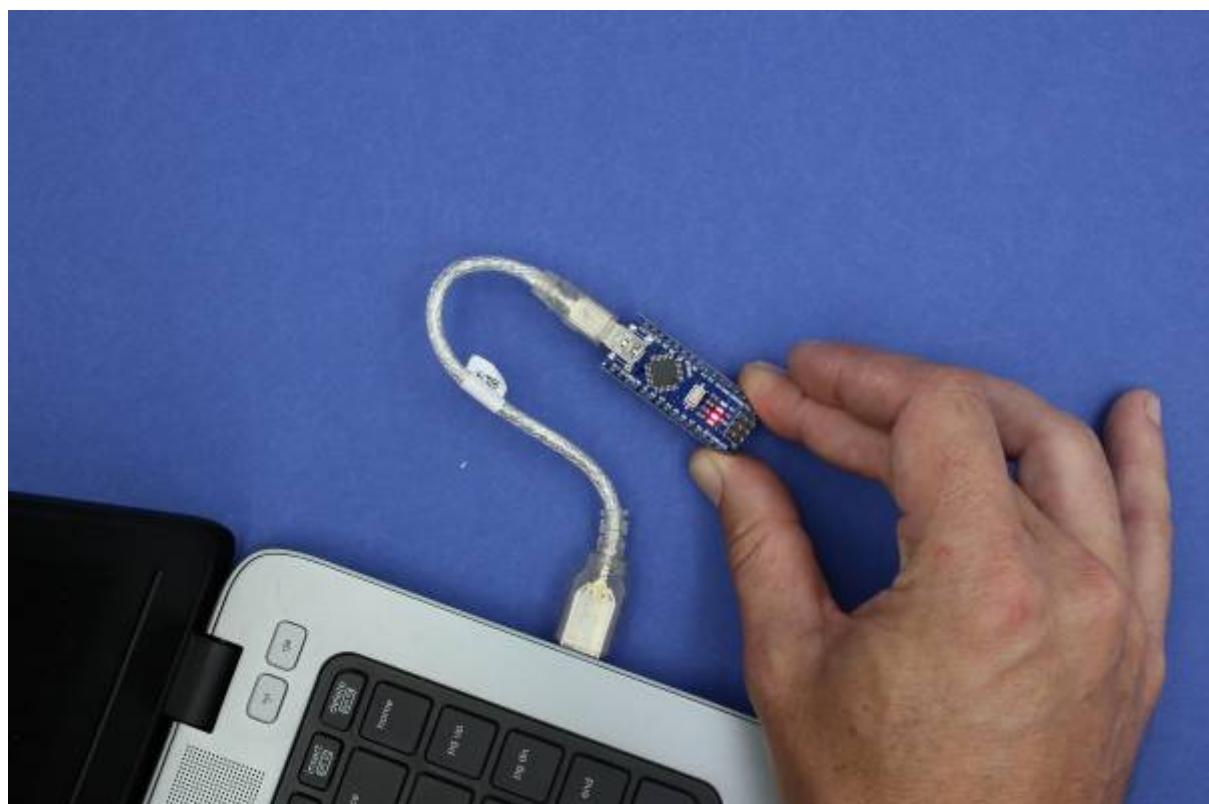
[JUST DOWNLOAD](#) [CONTRIBUTE & DOWNLOAD](#)

If you are using a genuine Arduino with the FT chipsets you won't need to download any drivers on PC and MacOS Sierra or later. If you are using a knockoff Arduino with the CH340G chip you will need to download the [drivers](#) for Mac.

Follow the instruction and install the program on your computer.

Connect Your Arduino

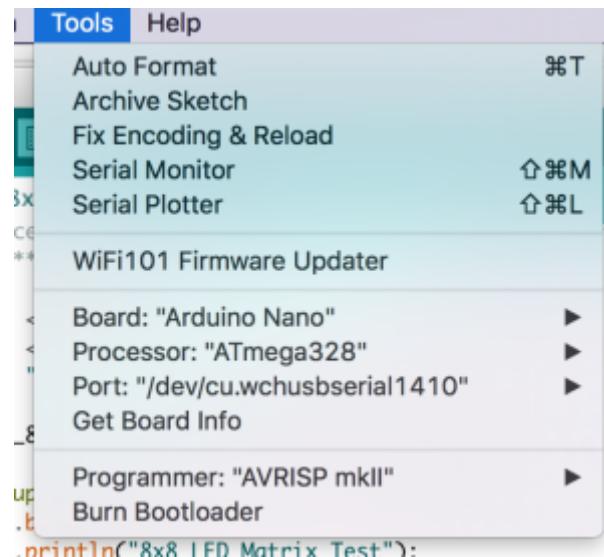
A red and then a green LED will light up on the Arduino, indicating that it's receiving power (for some Arduino's this may just be a single red light).



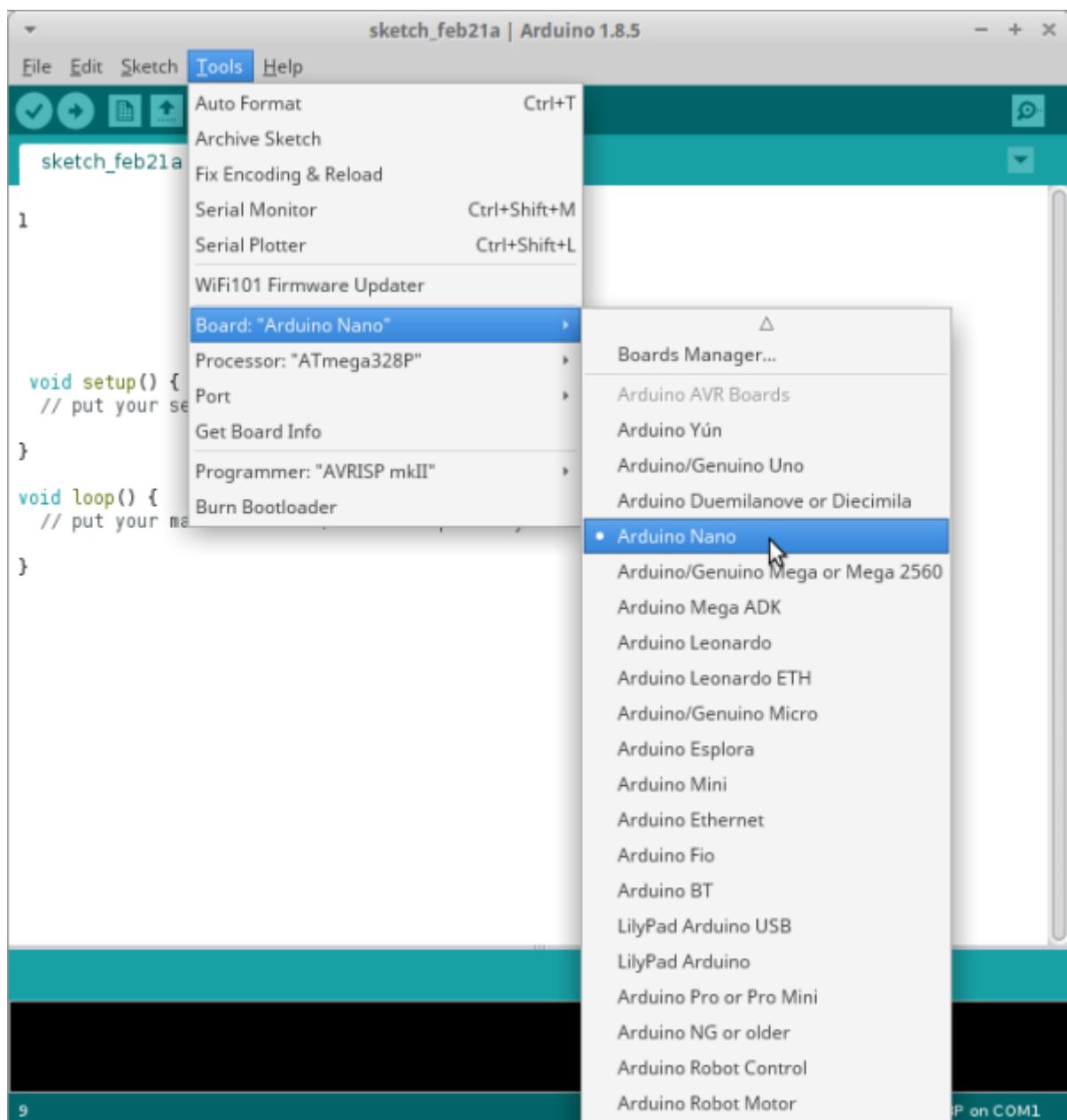
Now launch Arduino IDE.



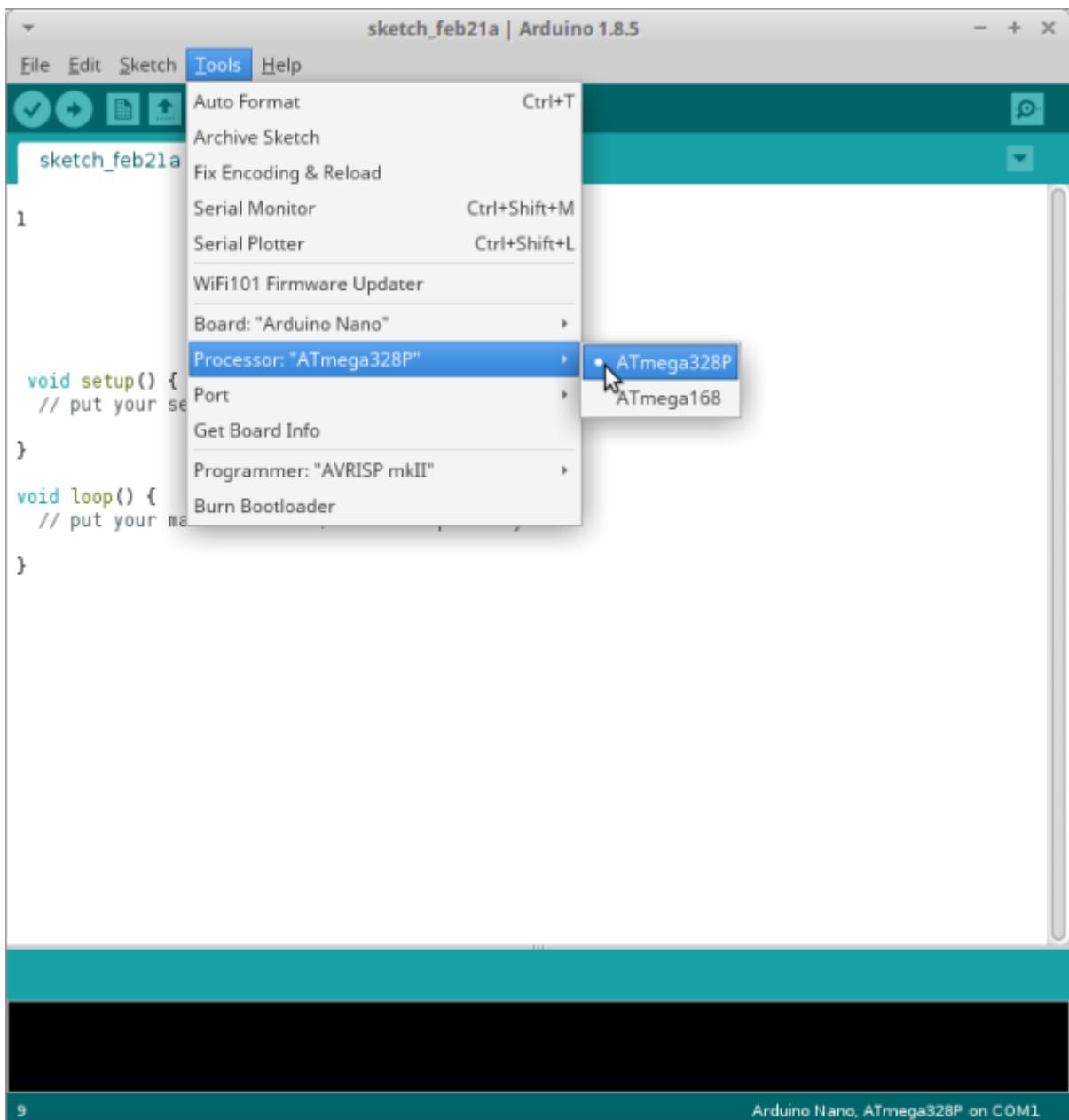
Go to the *Tools* menu and change the following:



- **Board to Arduino Nano**



- Processor to ATmega328



- **Port** to USB/Serial (different depending on OS and Chip)
 - If you're having problems finding a USB port you most likely need to download the correct drivers.

Once again exact OS (and version) as well as the chip make a big difference.

Once that is done, run "Get Board Info" in the same menu, this should bring up a small popup with some information.

As long as you're not receiving an error in the console, you have successfully connected your Arduino!

Test with the Blink Example

Another step you can take to test that everything is working is to locate one of the basic example projects and upload it to the Arduino.

- go to File → Examples → 01.Basic → Blink
- click on the verify button (Check mark) and let it compile.

You should get a message at the top of the console telling you when it's done (this should take no more than a few seconds).

- once this is done, without any errors, click the Upload button (Arrow).
- this will upload the instructions to the Arduino and it will say "Done Uploading" when everything has worked.
- your red LED on the Nano should now be blinking.

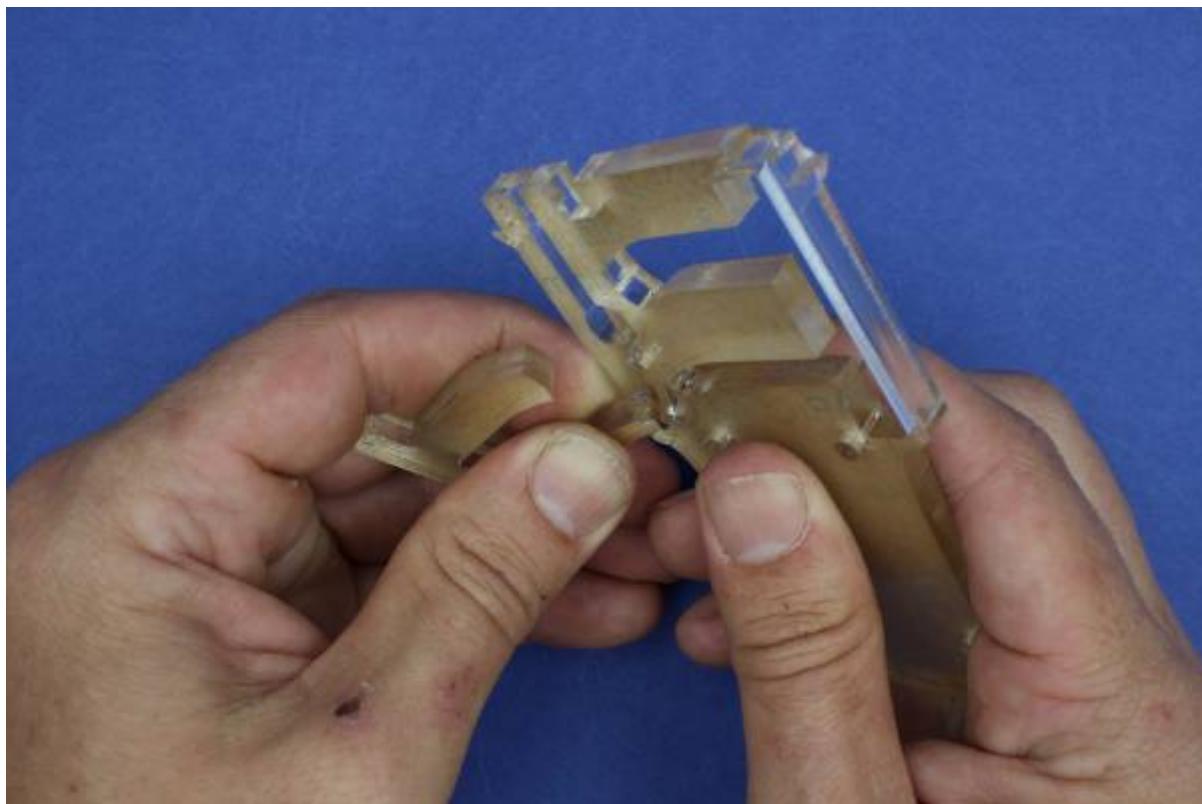
One second on, one second off.

- uploading scripts to your Arduino works!

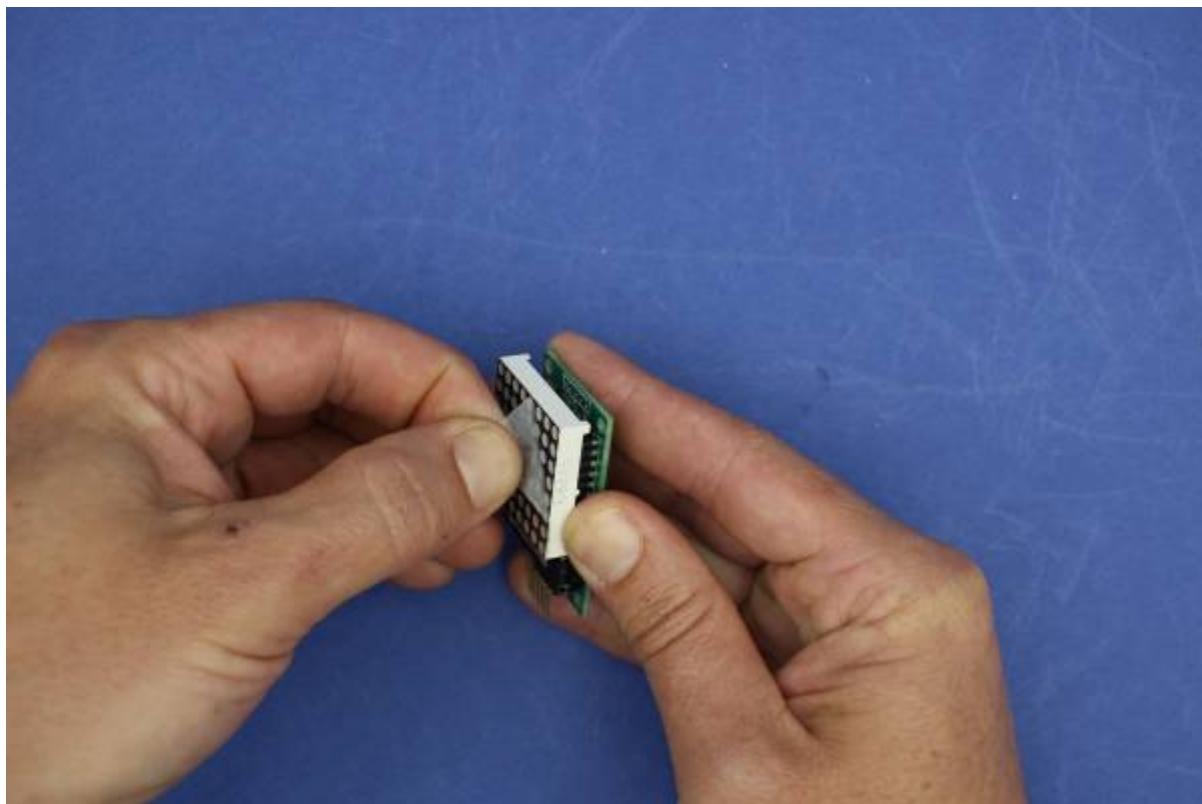
Now we are ready for the real projects.

Step Two: Assemble the LED panel and Nano

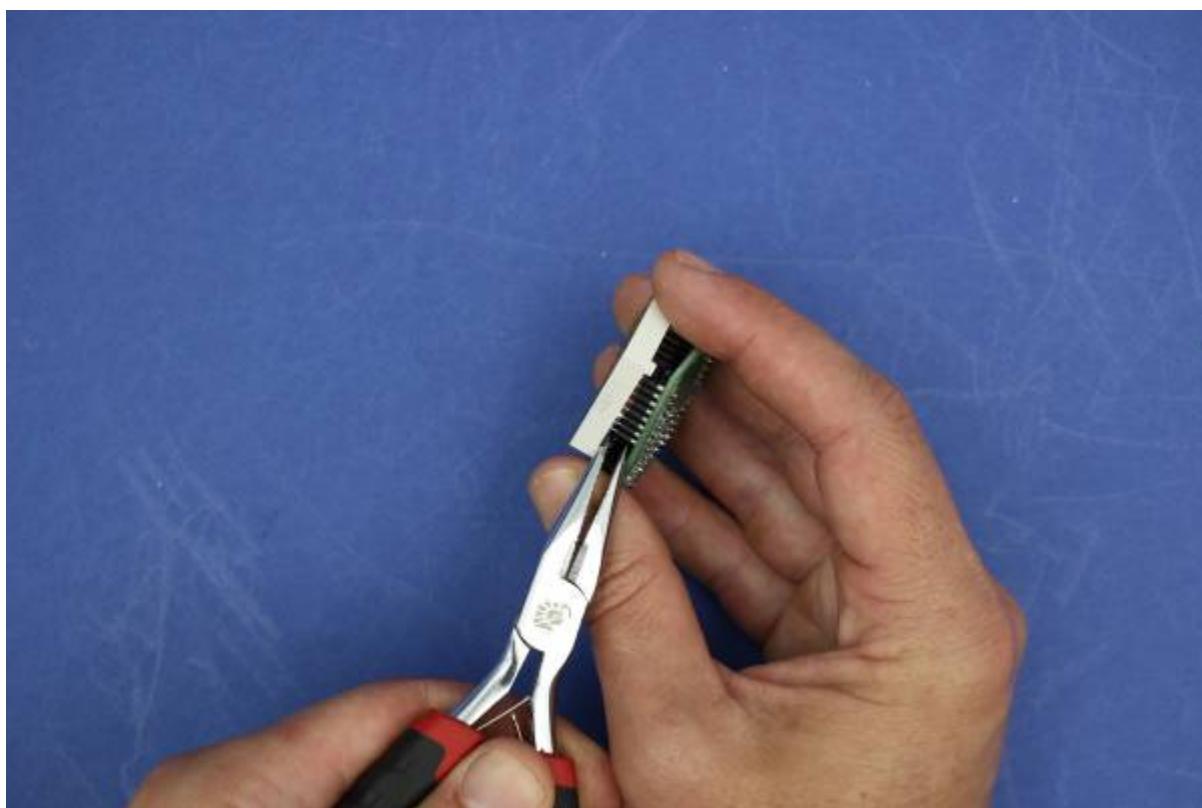
First let's peel the paper and pop out our case parts



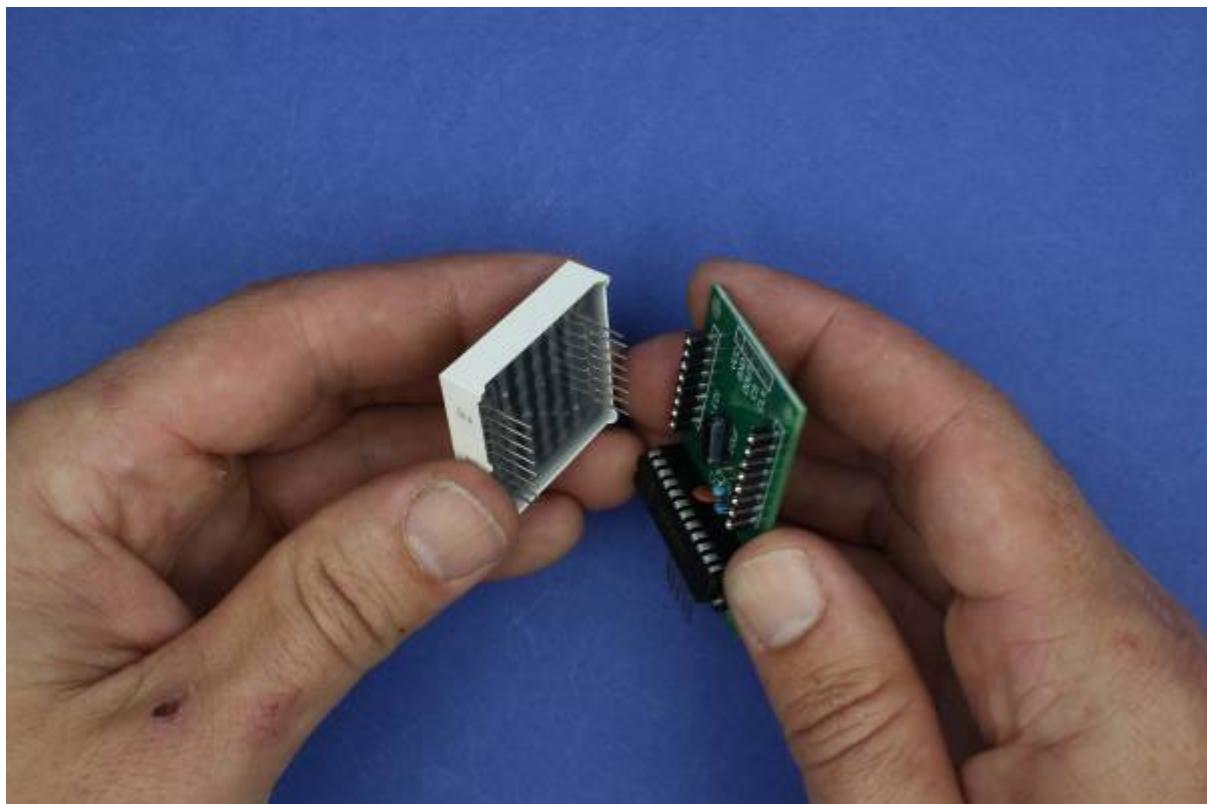
And peel off the plastic cover on the LED panel



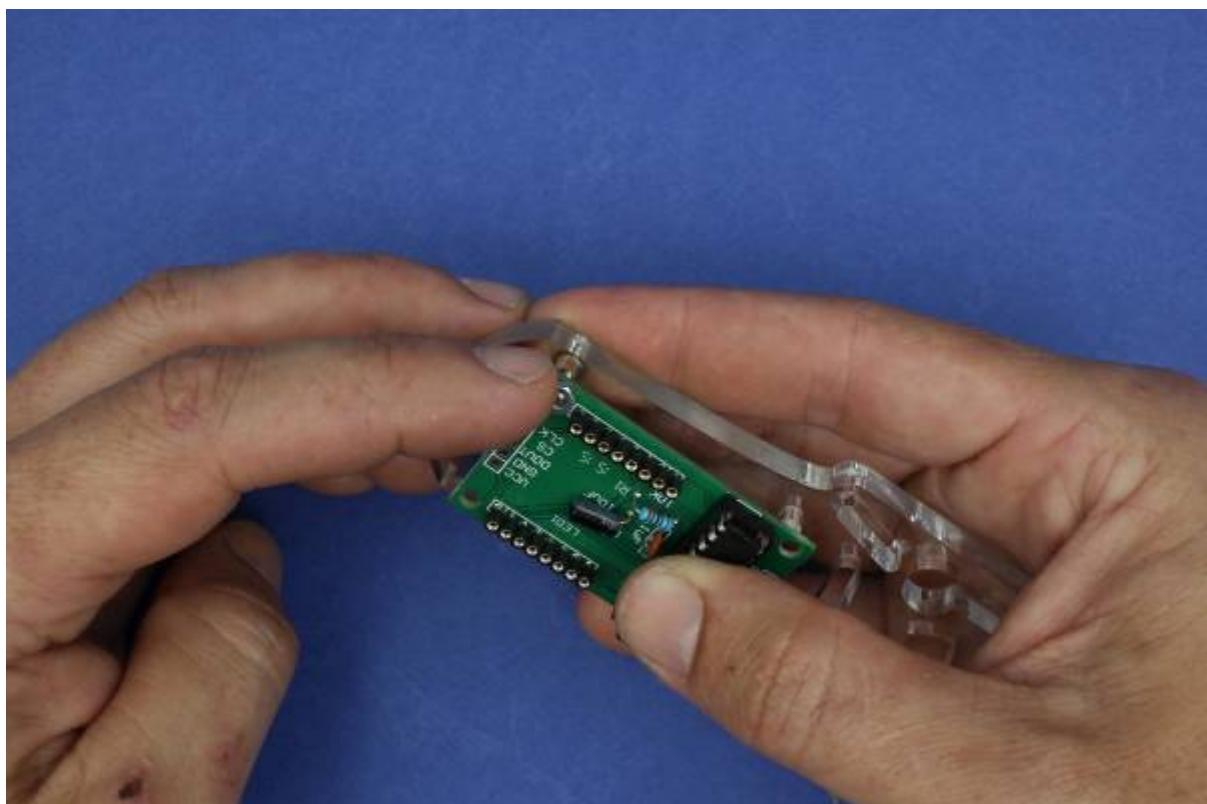
Then take the LED panel off its board. You can use the back of the pliers



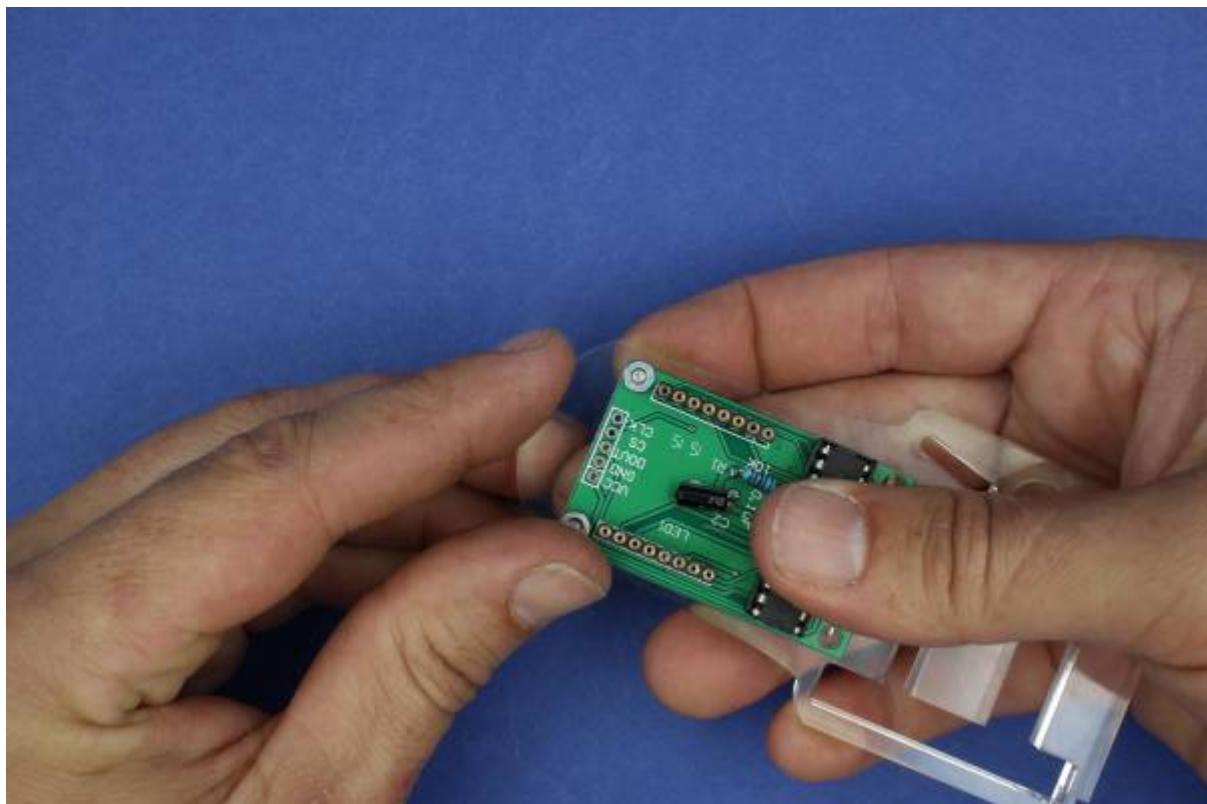
Be gentle and try not to bend the pins



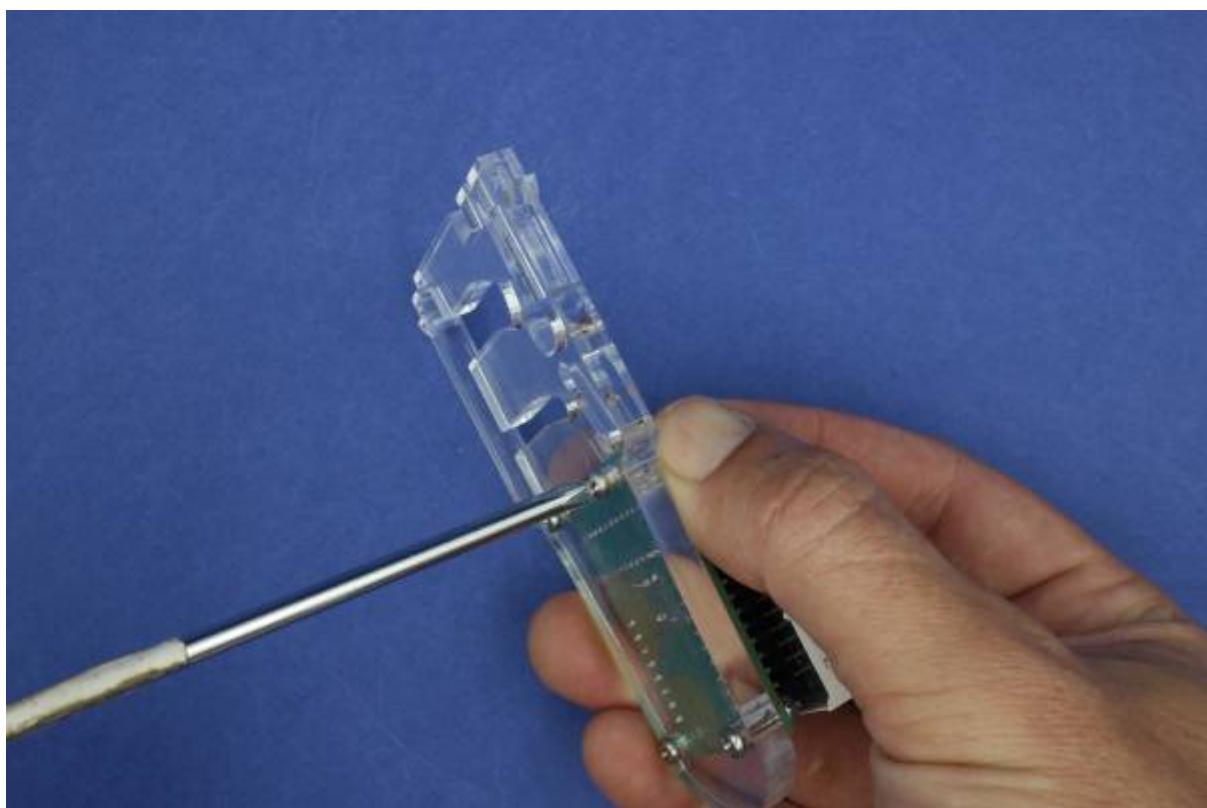
Attach the LED board to the case back panel using the four m2.5 bolts and nuts. The two 12 mm (short) screws are for the top



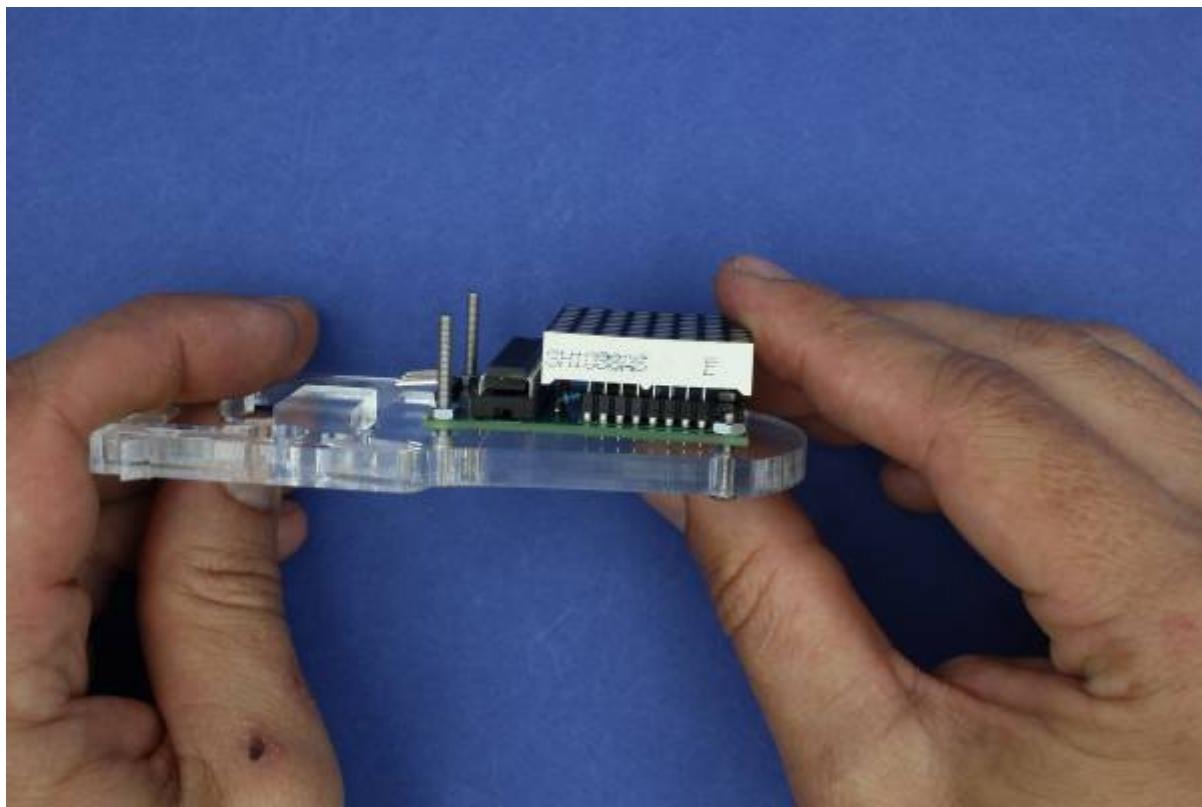
Then the two 25mm bolts for the bottom



Tighten gently with the screwdriver

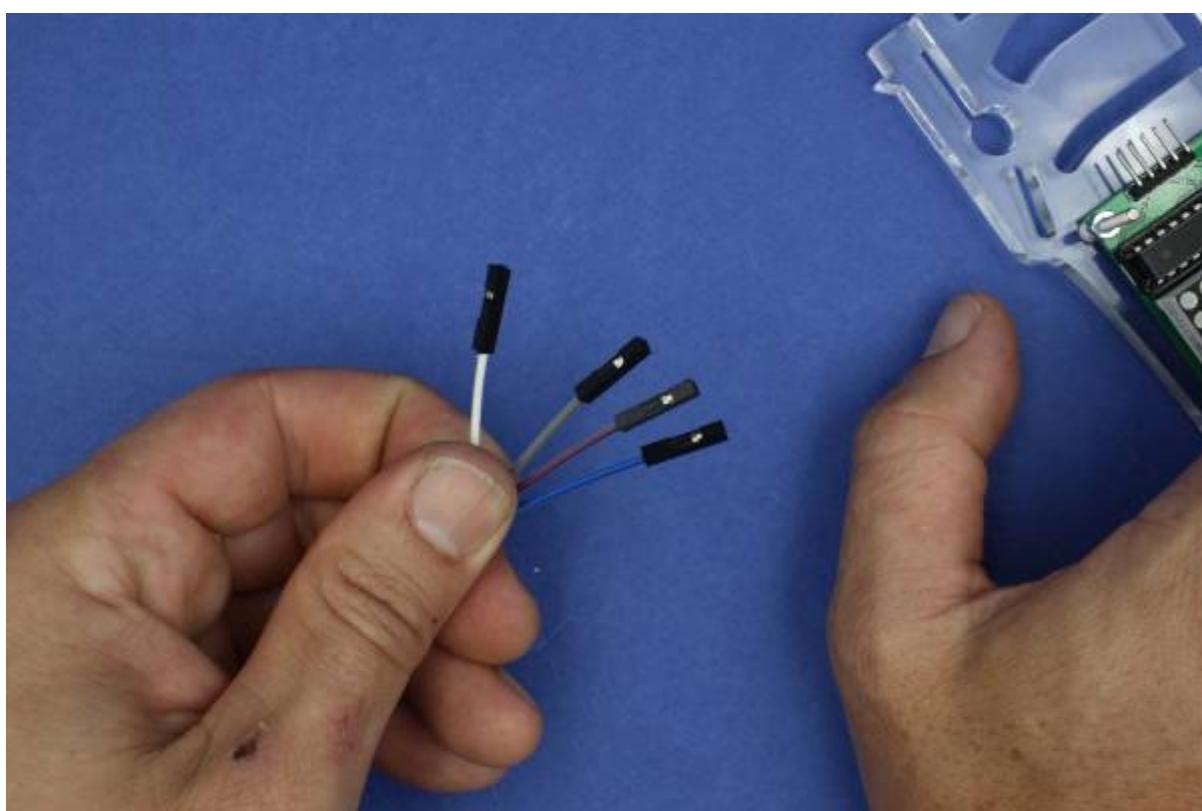


Pop the LED panel back on. Check it is the right way round, and the panel is level



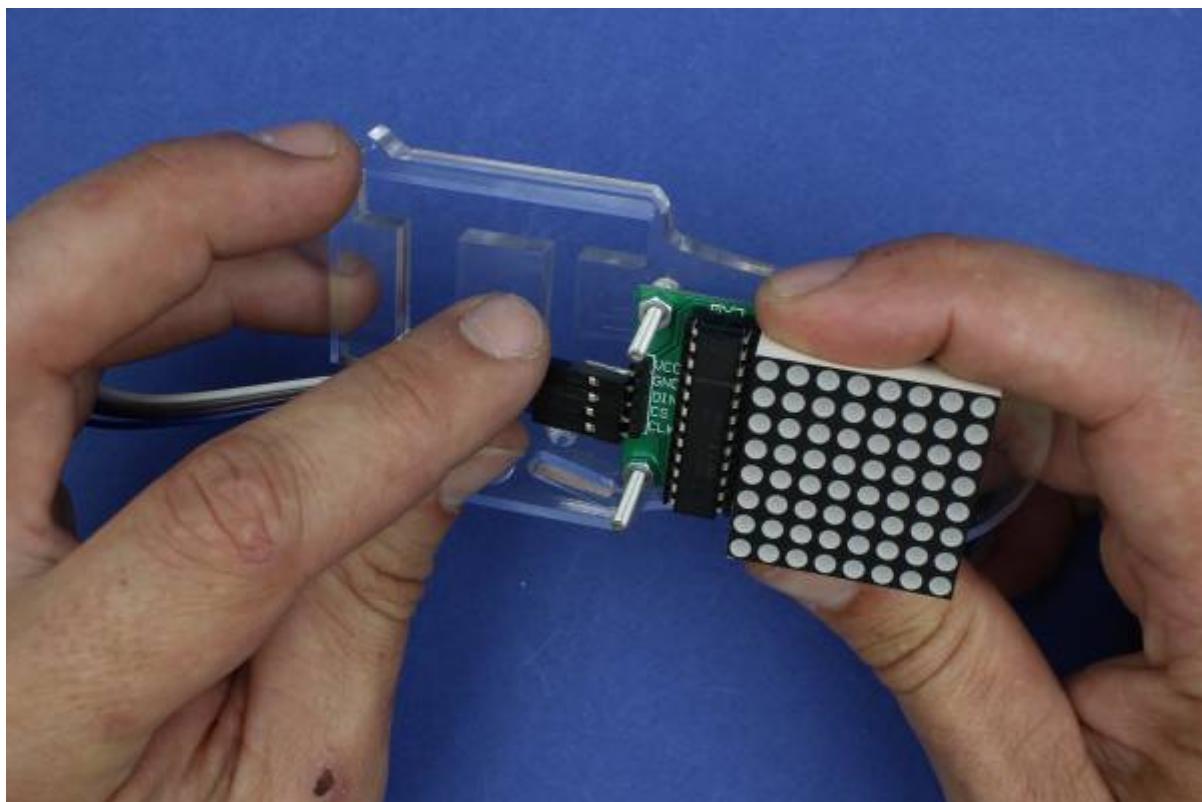
We're going to connect five jumper leads to the Arduino from the LED board

- detach one wire from your stack of five
- separate the rest out a little on each end

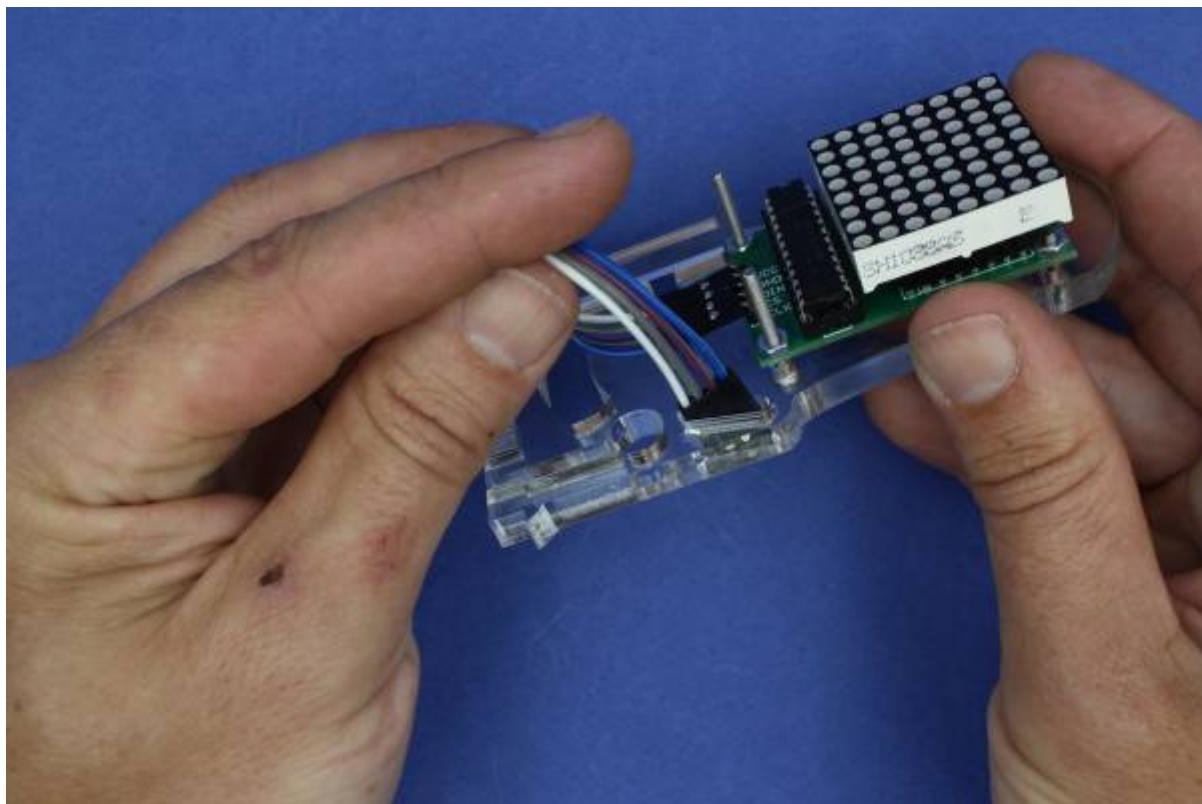


Attach the wires - from bottom to top:

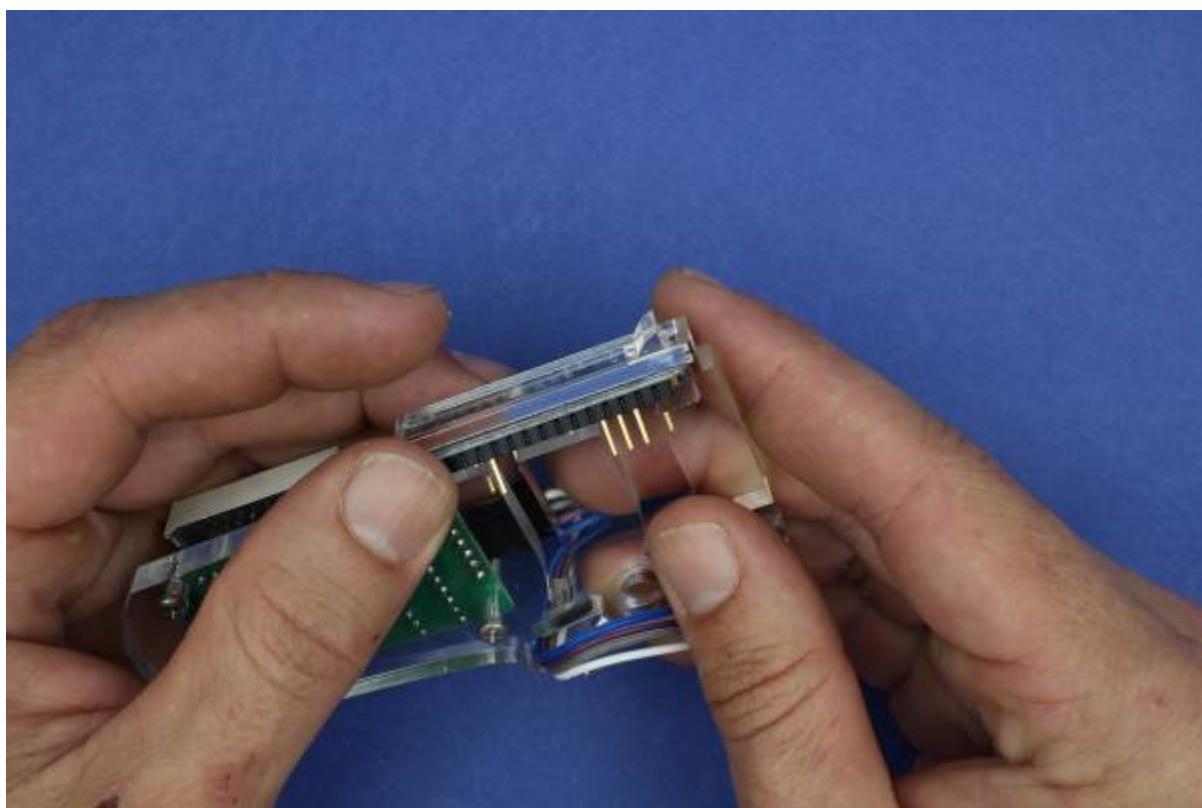
- SCL/CLK (System Clock)
- DIN(Data In)
- CS (Chip Select)
- GND (Ground)



Feed them through the slot



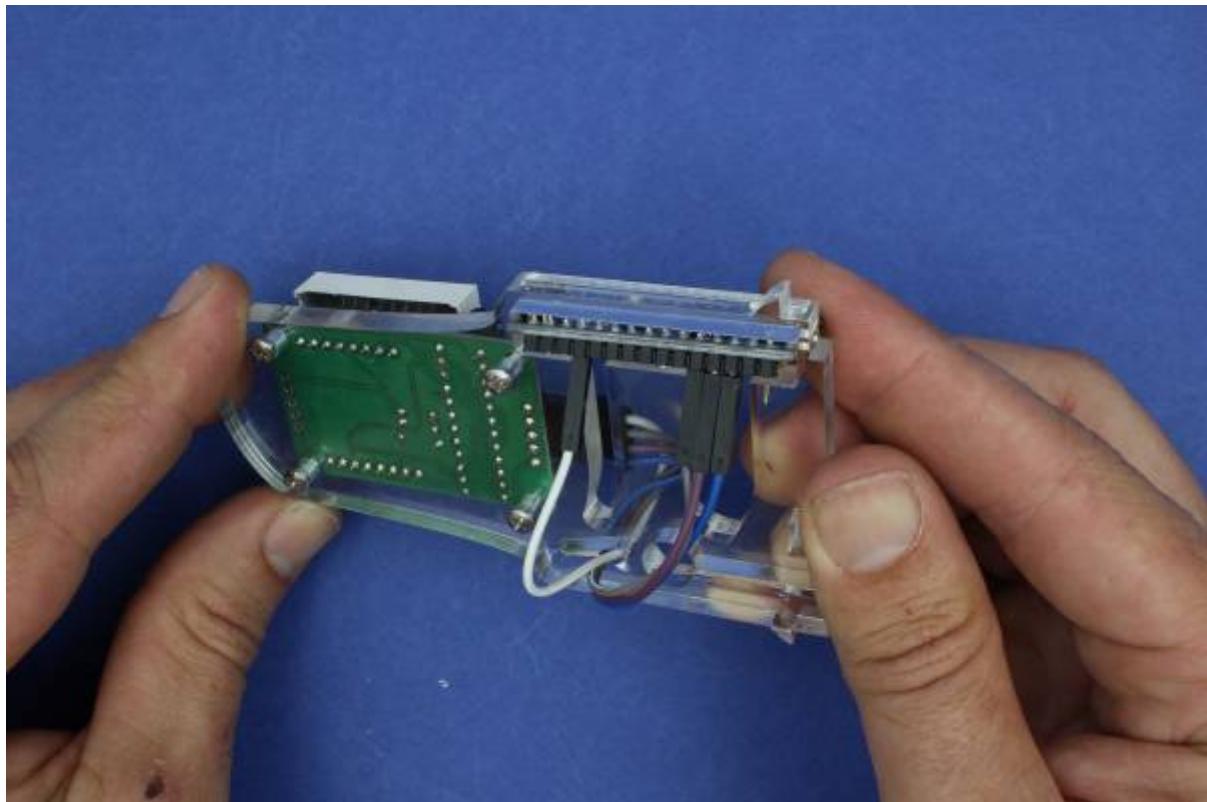
Flip the case round and place your Nano in the slot



Attach:

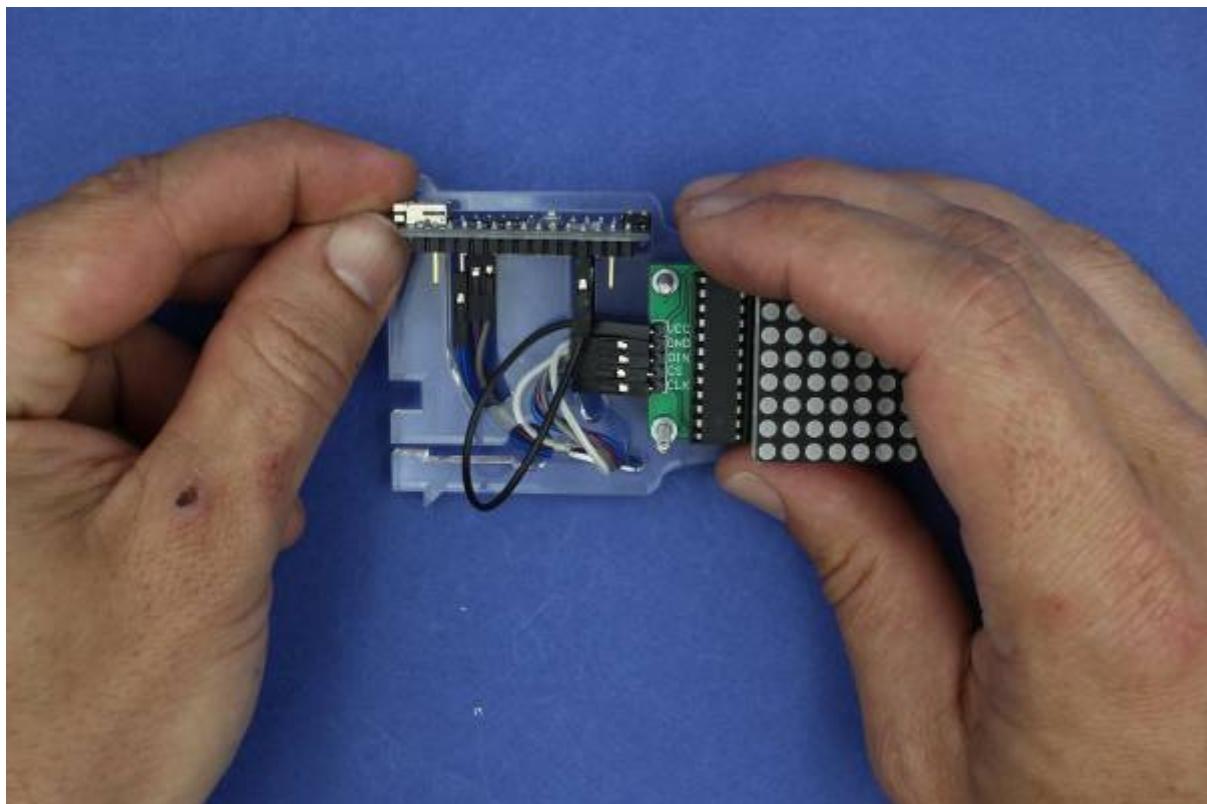
- SCL/CLK (System Clock) → D10

- DIN(Data In) → D8
- CS (Chip Select) → D9
- GND (Ground) → GND



Now back to the front, and attach

VCC (Voltage In) → +5V



Plug it into your computer - time to test our blinky lights!



Step Three: Using the Library and Project

Now its time to check our Arduino and LED are talking to each other.

Download and install the Library

In order to make the LED display a *lot* easier to work with we'll need to install a custom version of the MaxMatrix library (found in the [download](#) section below)

- go to Sketch → Include Library → Add .ZIP Library
- just find and click the downloaded file and it will be added to your project automatically

Download and open the project

[Below](#) you will find the project files compressed into a .ZIP file

- extract the content wherever you'd like it and open up the *wearables_matrix_ino* file
- the .INO file format should be associated with Arduino and will launch automatically
- alternatively you can open Arduino and select the File → Open option in the menu

Note: You won't be able to run this script unless you have downloaded the library above.

Upload the Project

It is time to test your project file in your badge. Upload it the same way as the blink file. Your badge should say Free Bear Hugs.

Step Four: Fit the Battery

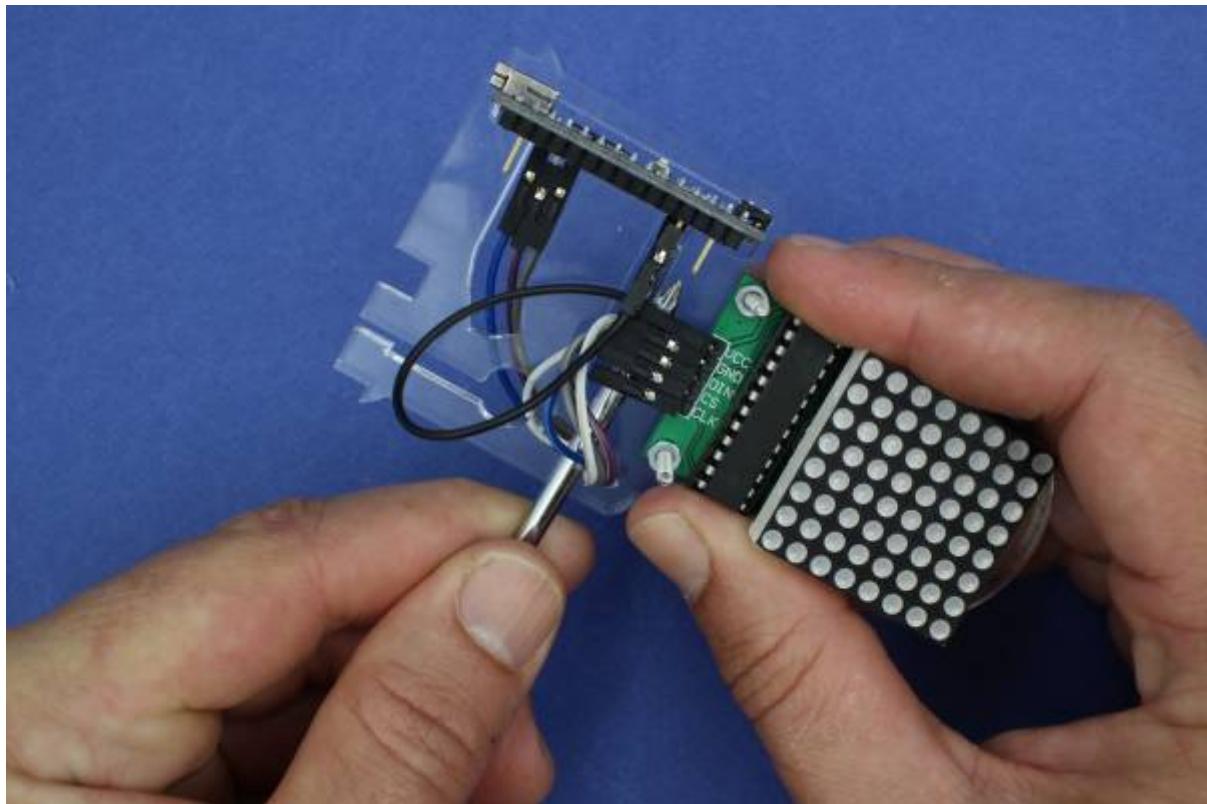
It is time to fit and connect the battery. Both the battery and the charging board have protection circuitry but must always be handled with care.

Make sure you take care when handling the battery.

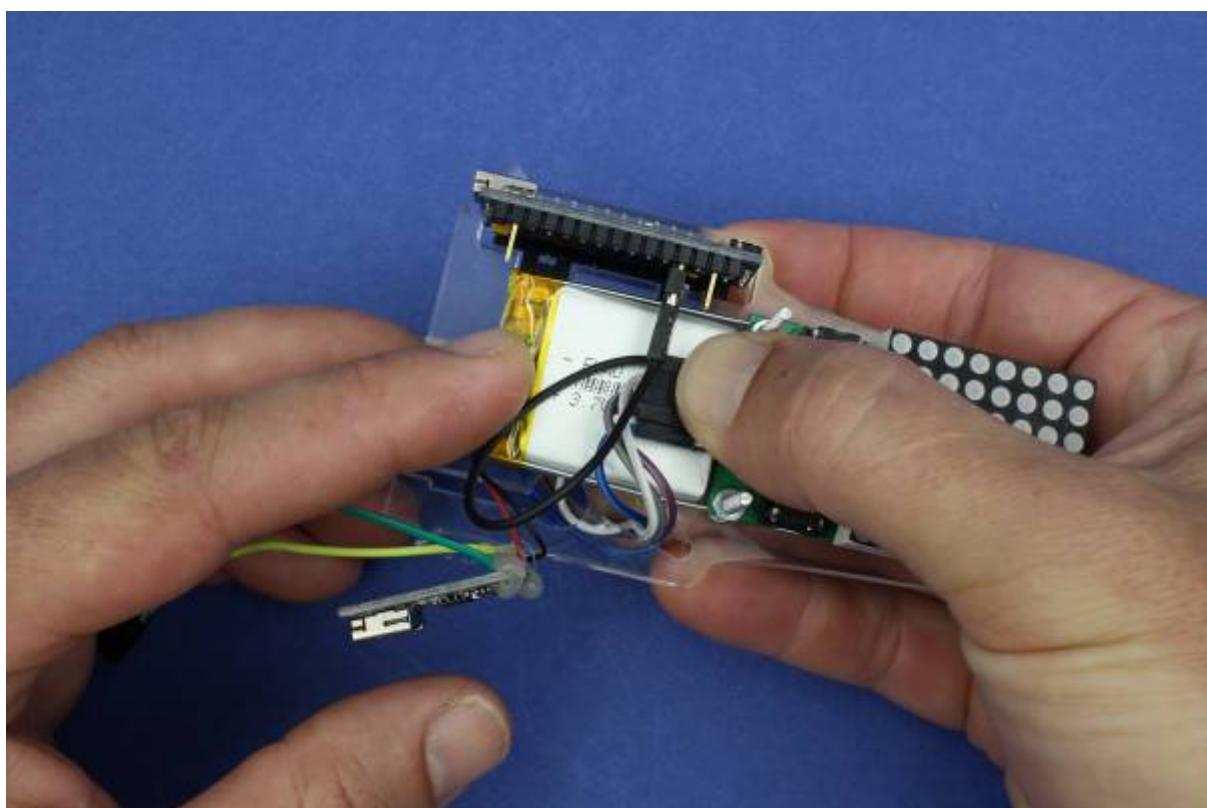
- check that the RED wire from the battery is connected to B+ and the BLACK wire is connected to B-
- do not let any conductive surfaces or wires touch. (This will short the battery which makes it unusable)
- if this is not correct please notify the facilitator immediately!

the components will need to be rewired or swapped out

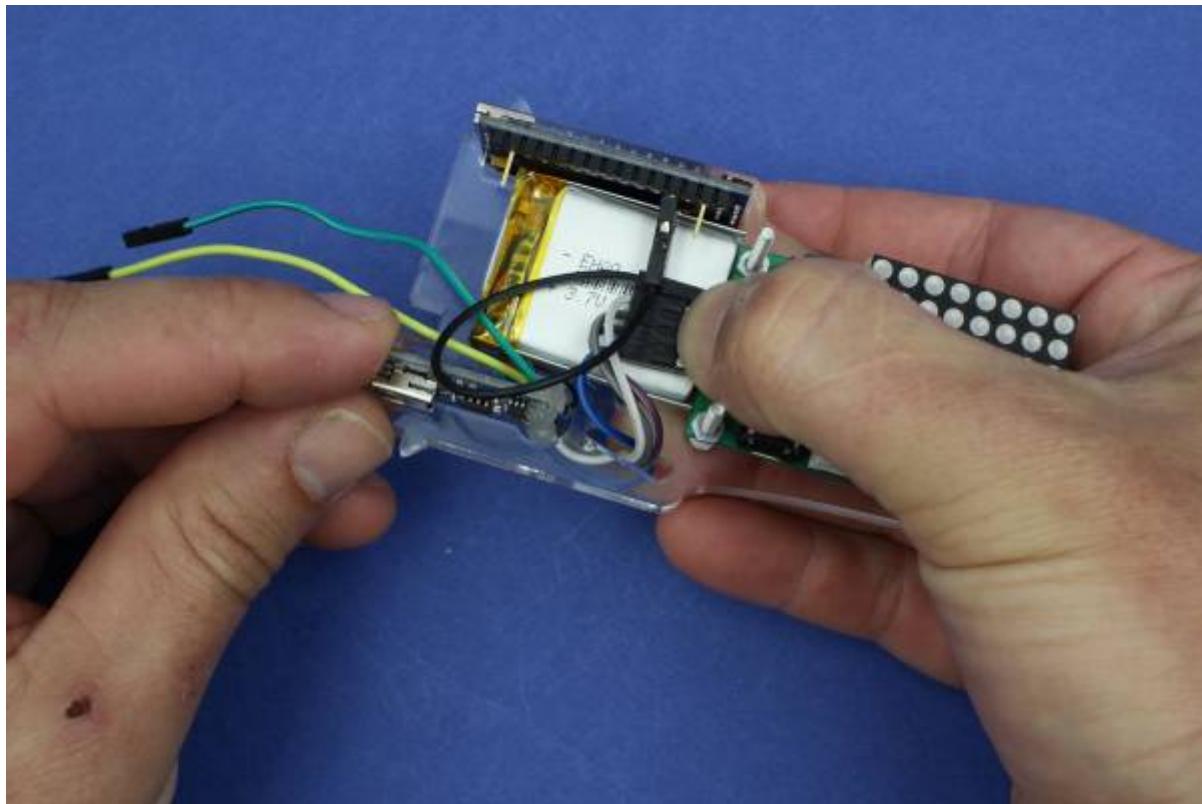
First make some space under the pins of the LED circuit board



Slide the battery in and hold it in position with your thumb

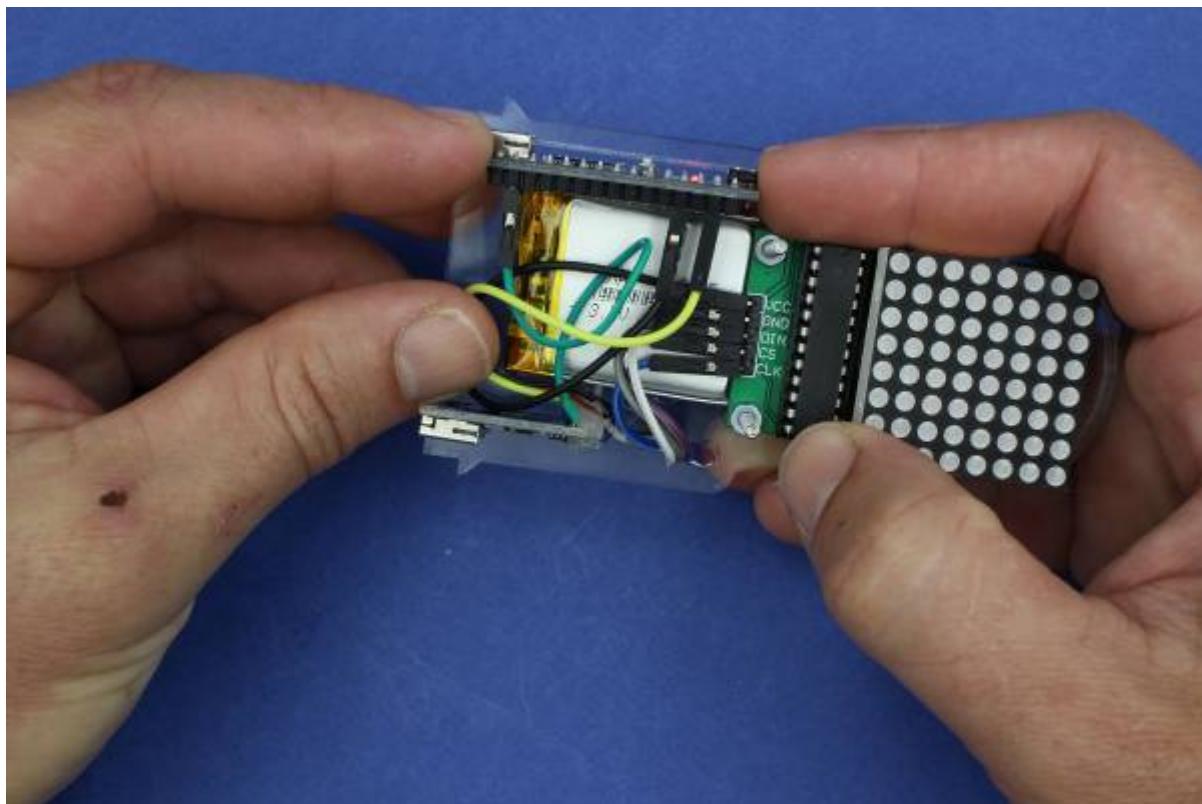


Slide the USB charge board into the slot



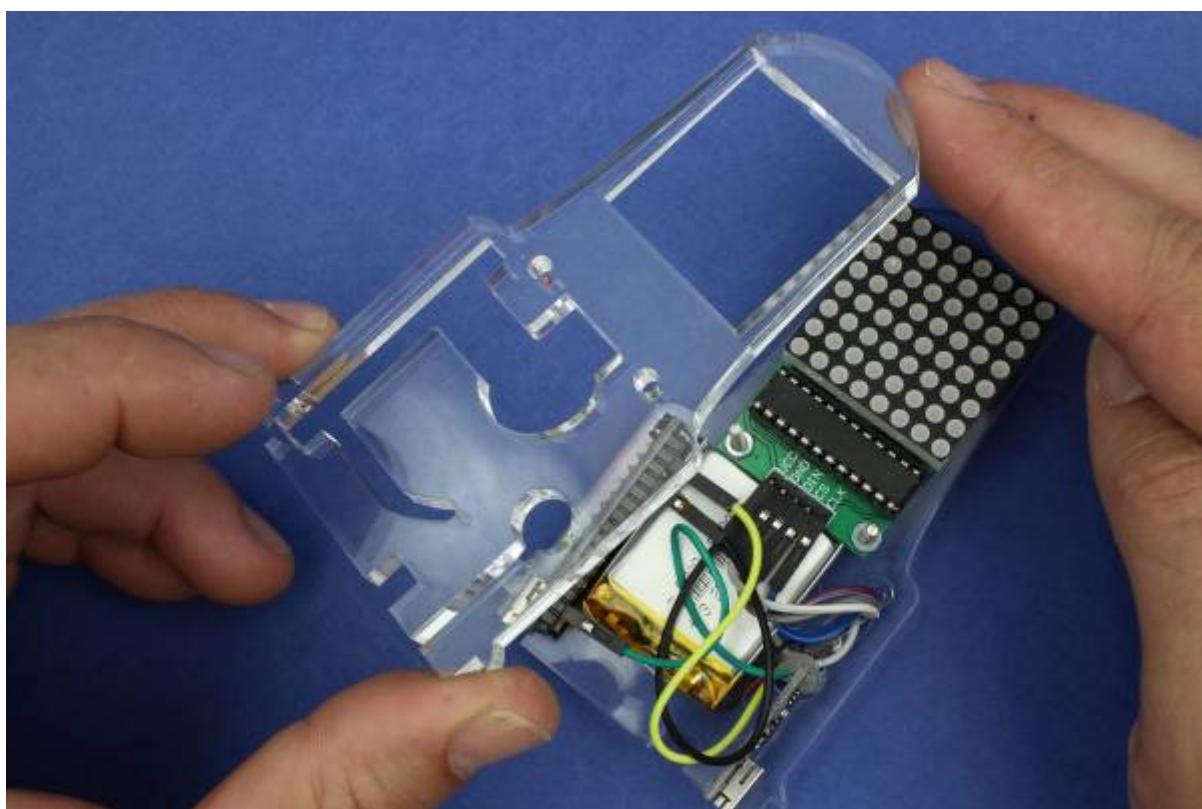
Connect:

- battery - (negative) to Arduino Ground (GND)
- battery + (positive) to Arduino 3V input (3v3)
- your Arduino will light up, and if your battery has enough charge, so will the LED screen

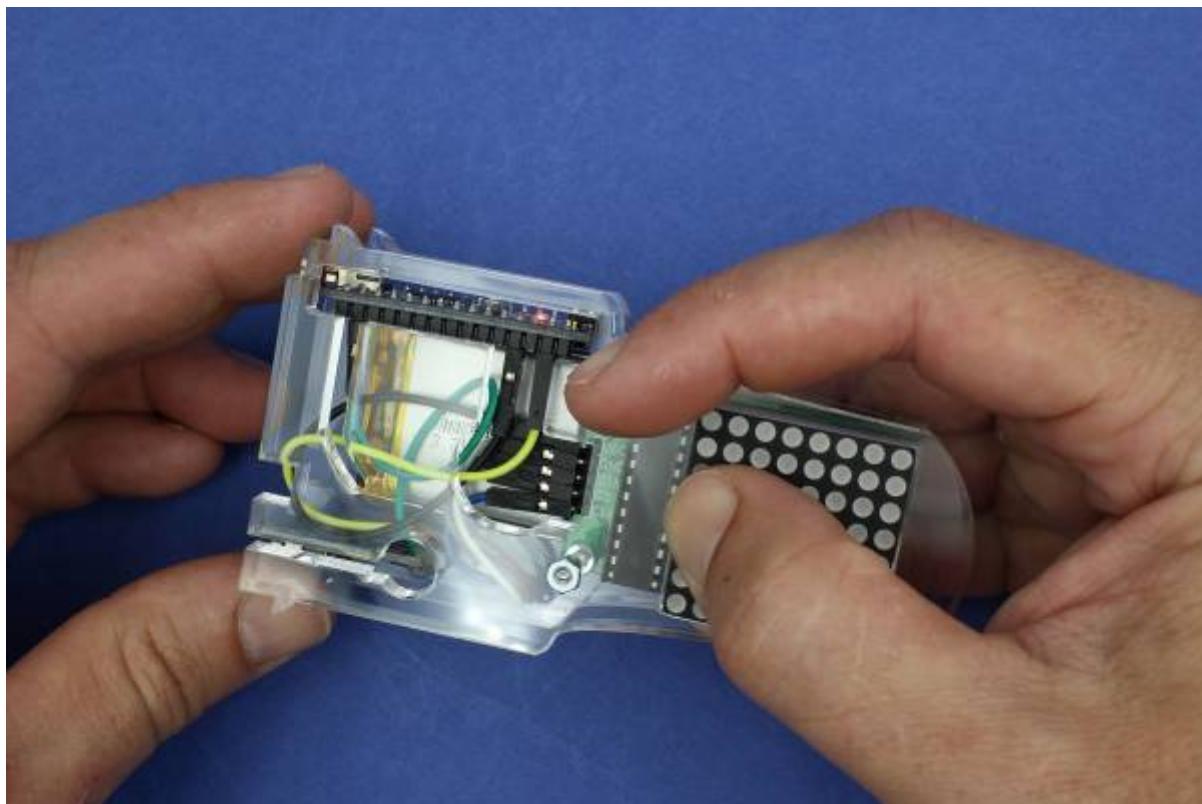


Step Five: Assemble the Case

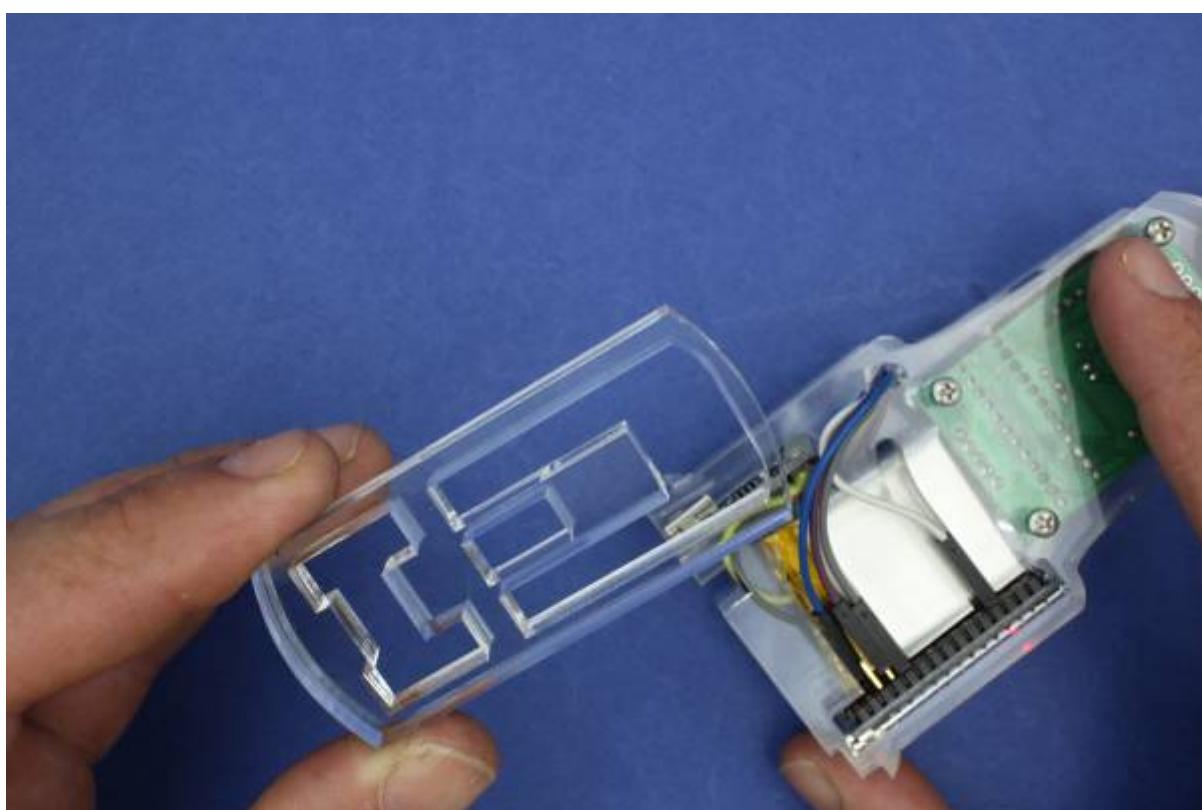
Grab the case front



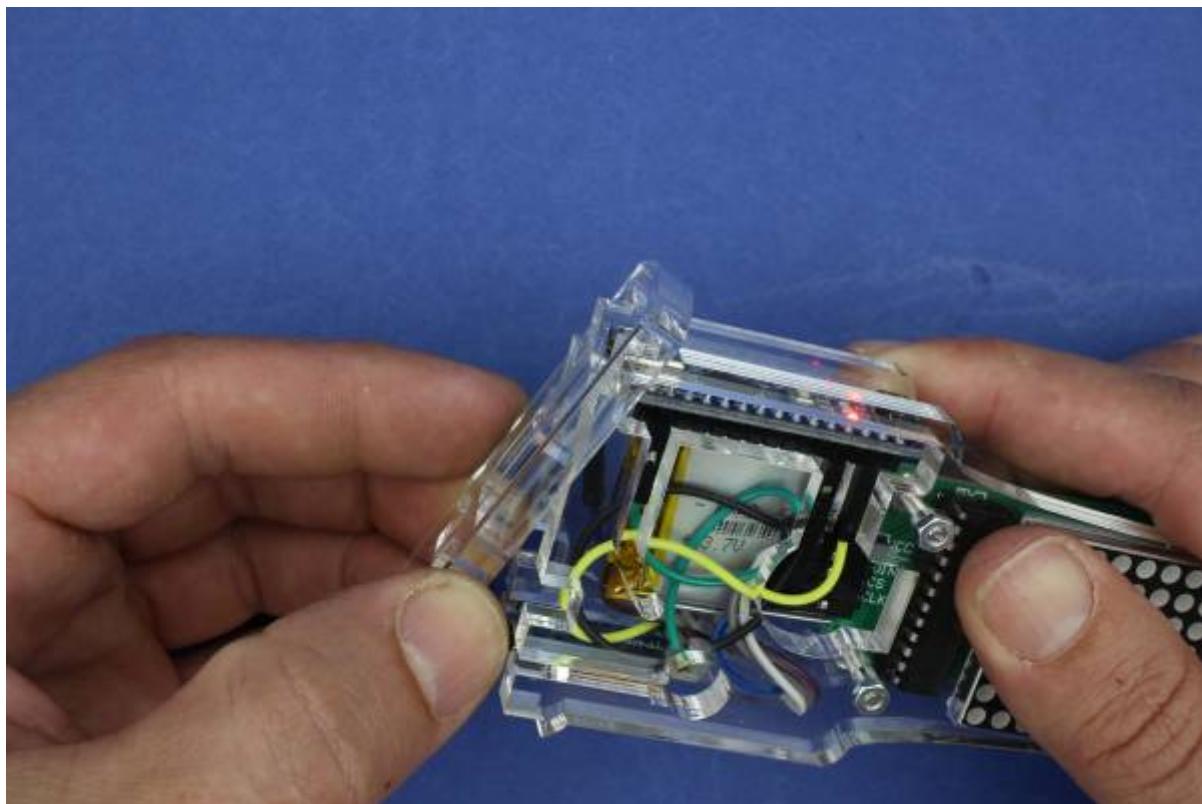
Use the two remaining m2.5 nuts to fix the front of the case



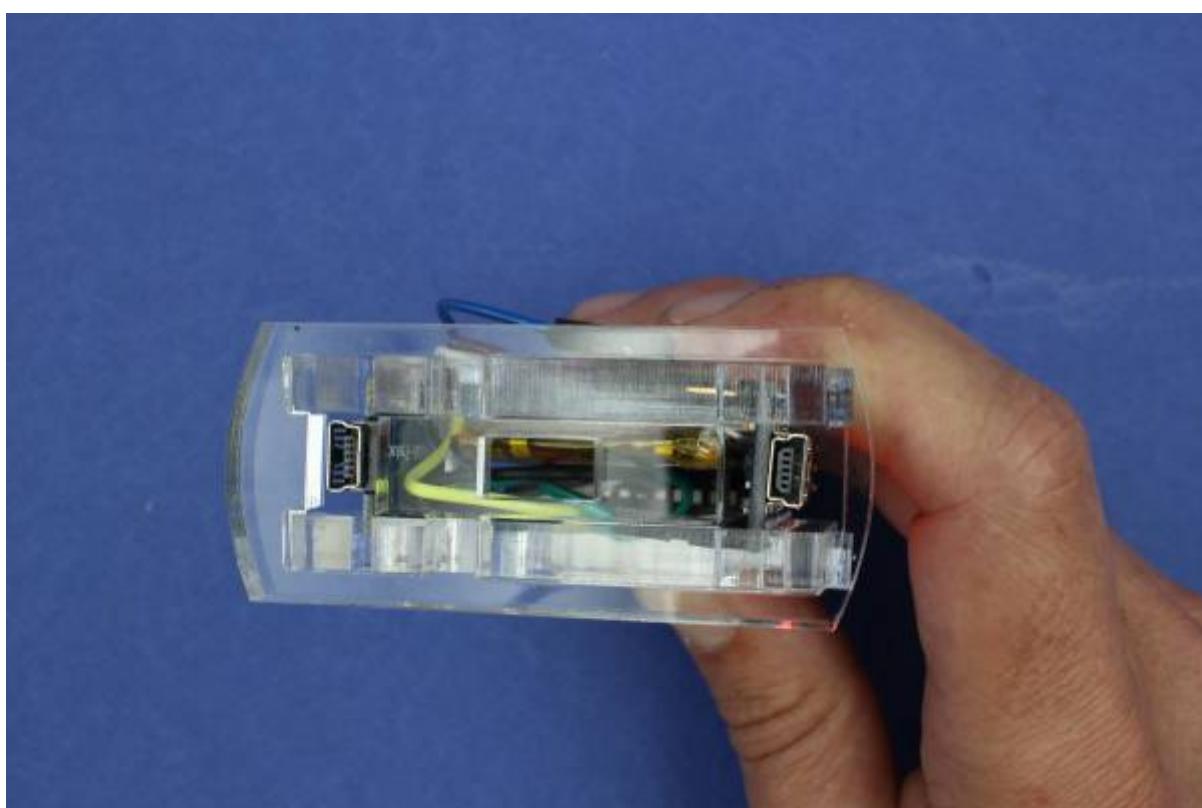
Grab the case bottom



Line up the Arduino end first



Then slide the case so it sits flat

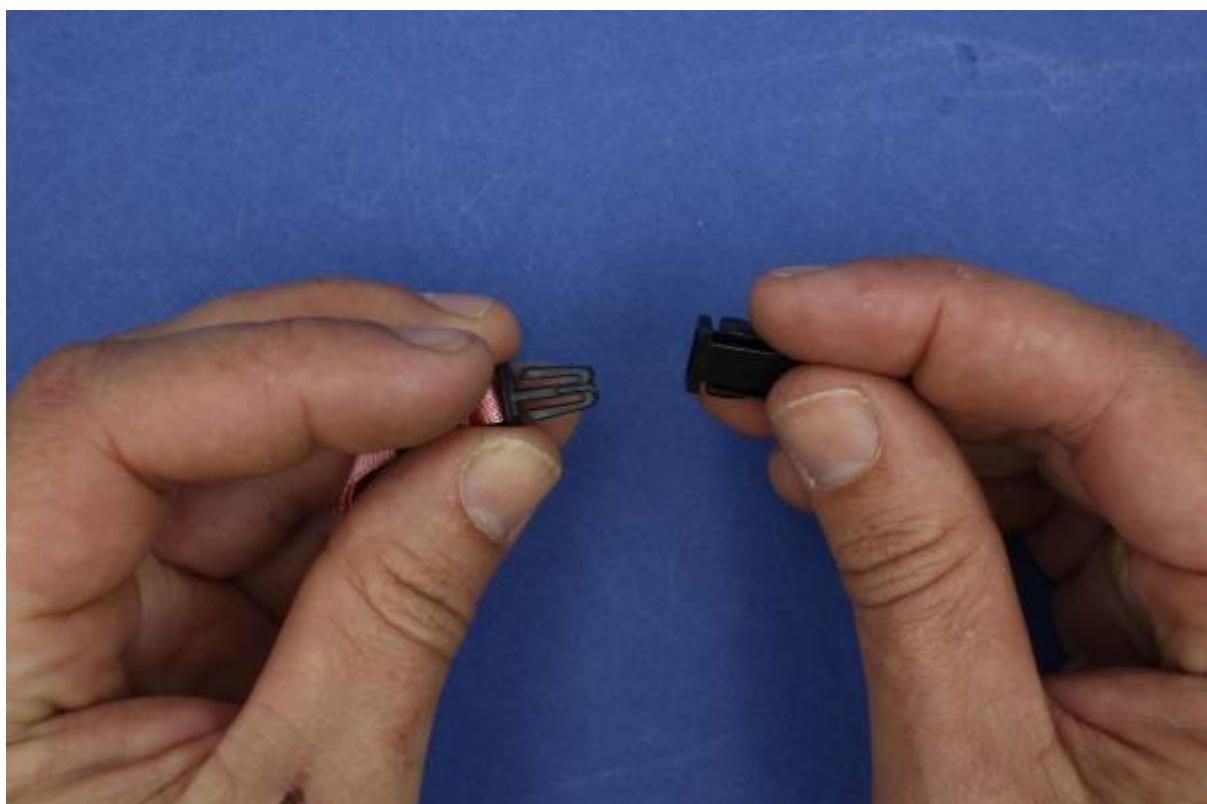


Insert the Lanyard

Its time to hot glue the lanyard clip into place.

The lanyard will not click out after this, it will no longer be removable.

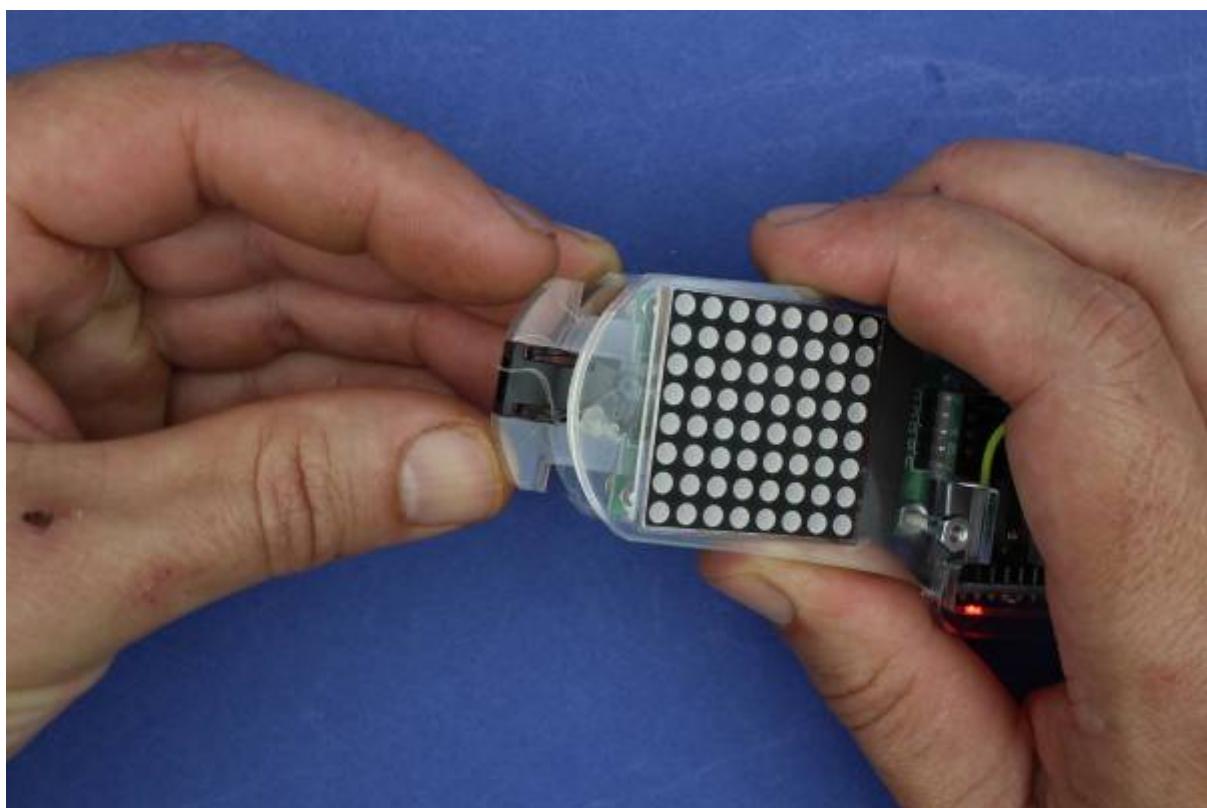
First remove the Lanyard clip from the necklace



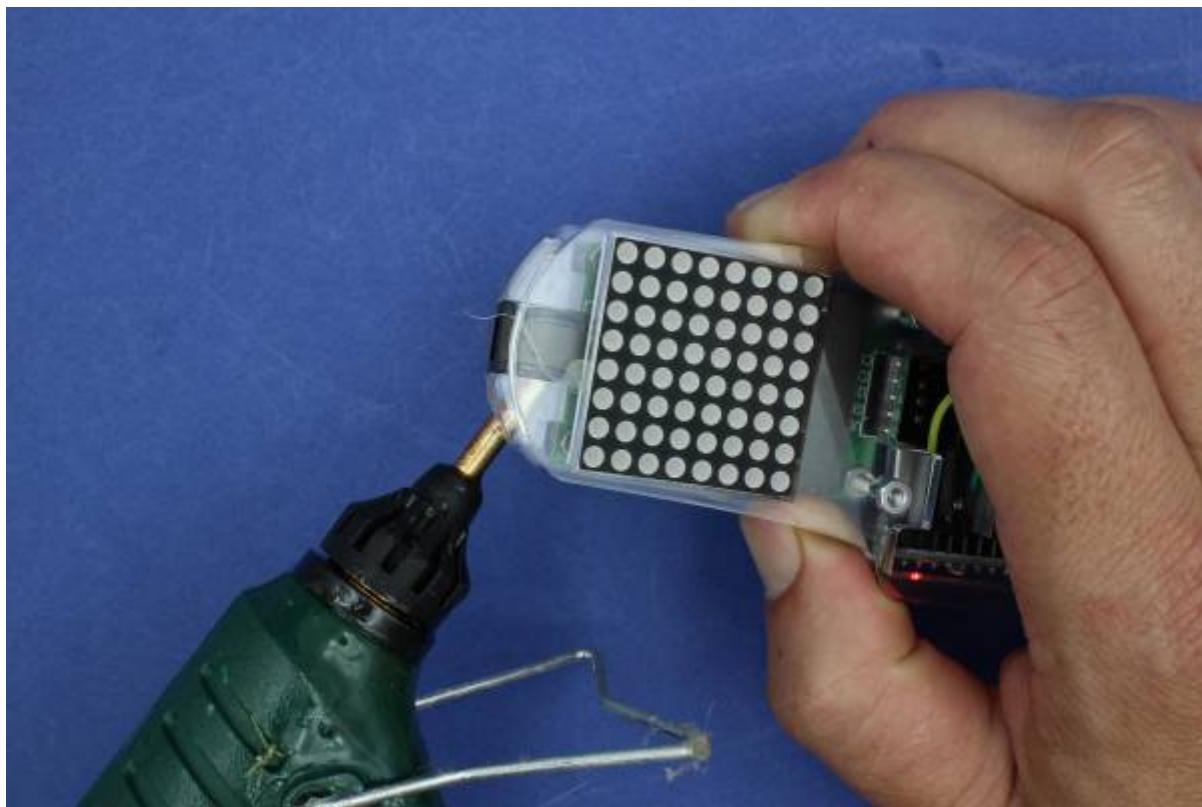
Hot glue the lanyard clip into the spacer



Slide the lanyard clip in behind the LED screen



Hot glue it in place

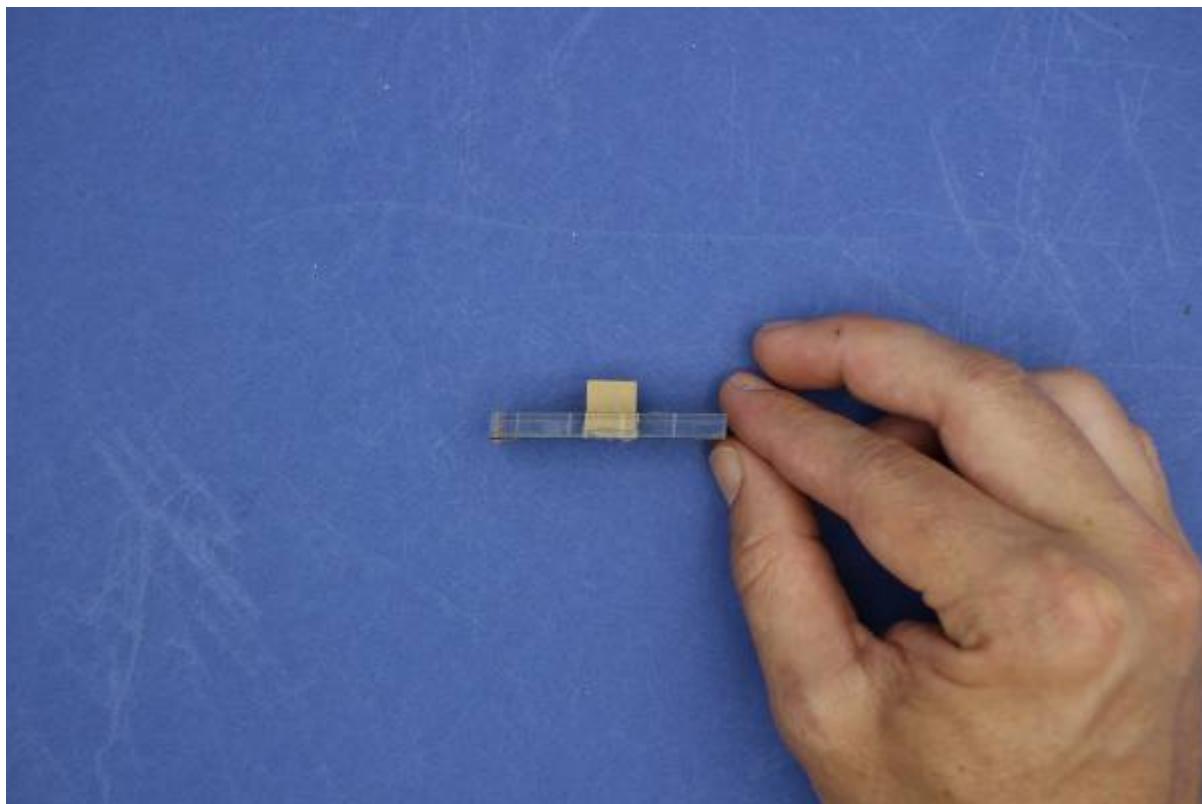


Step Six: Vacuum Forming

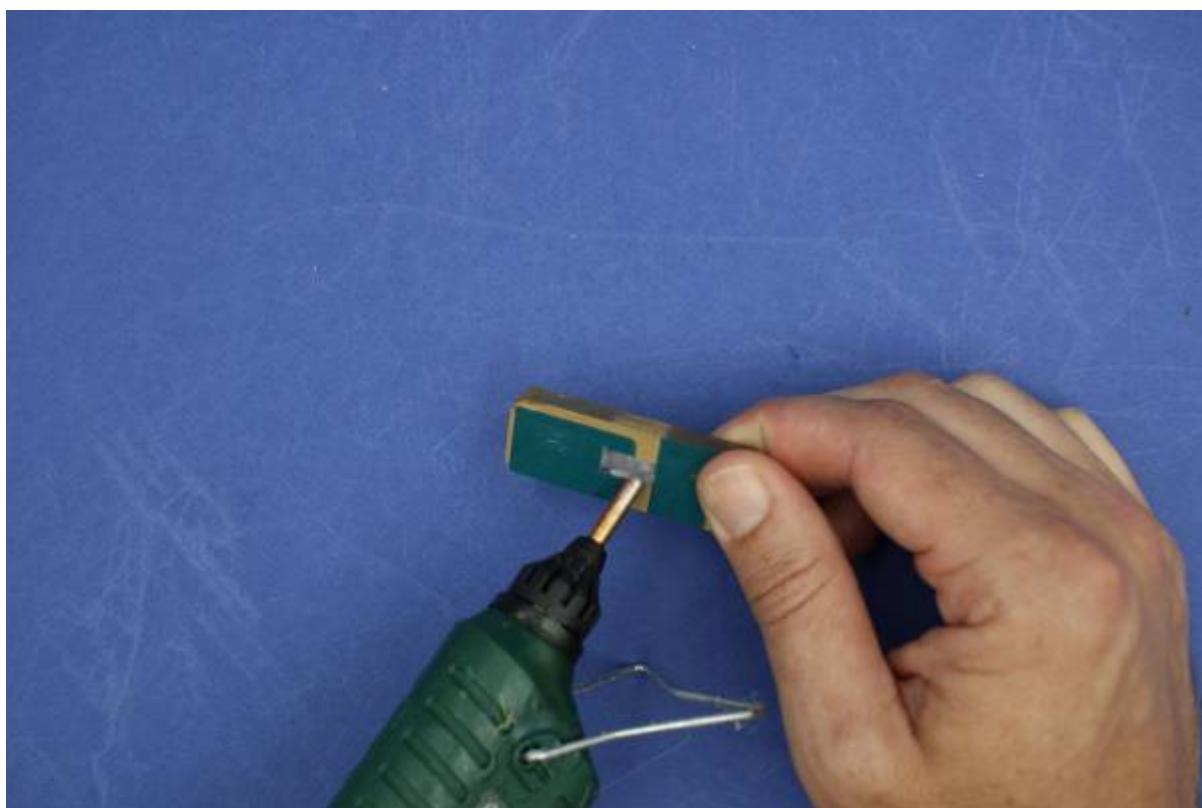
Vacuum forming is an almost instantaneous process, so we need to get all set up before we start, and make a jig to hold the case in place

Making our Jig

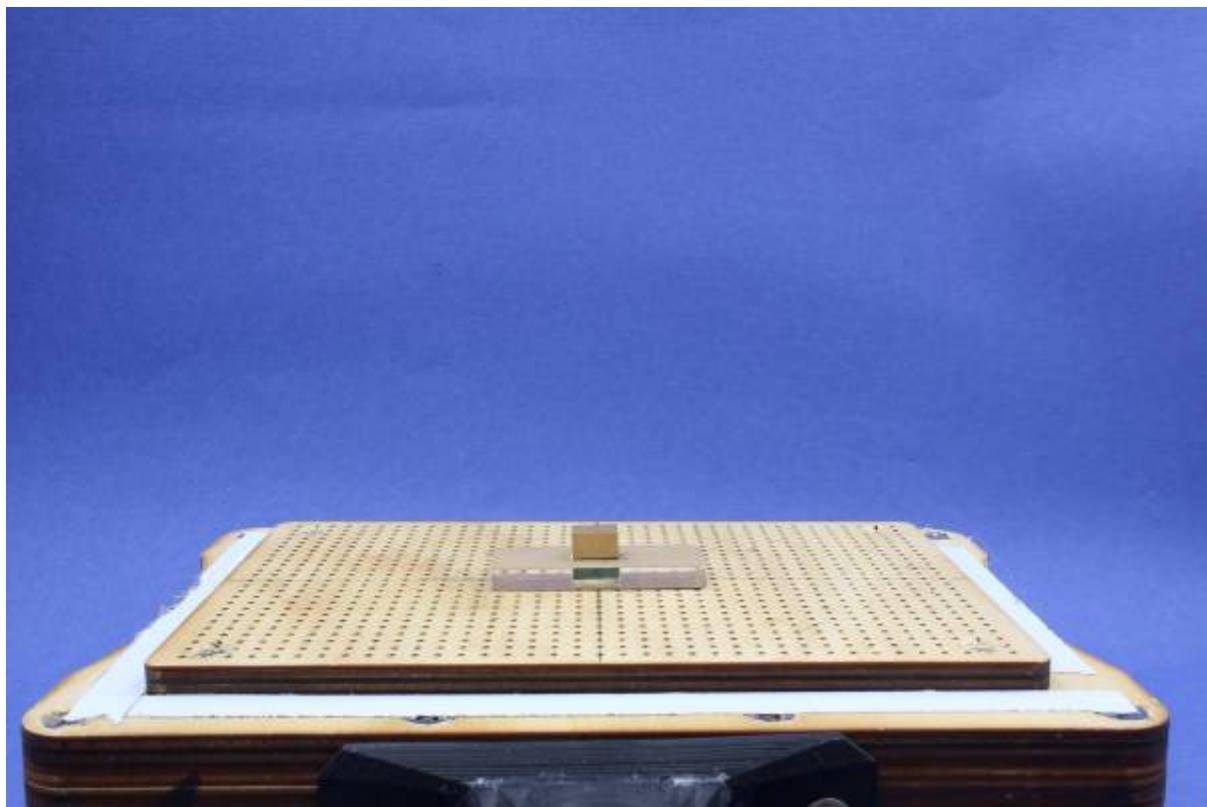
Slide one of the blocks into the jig piece



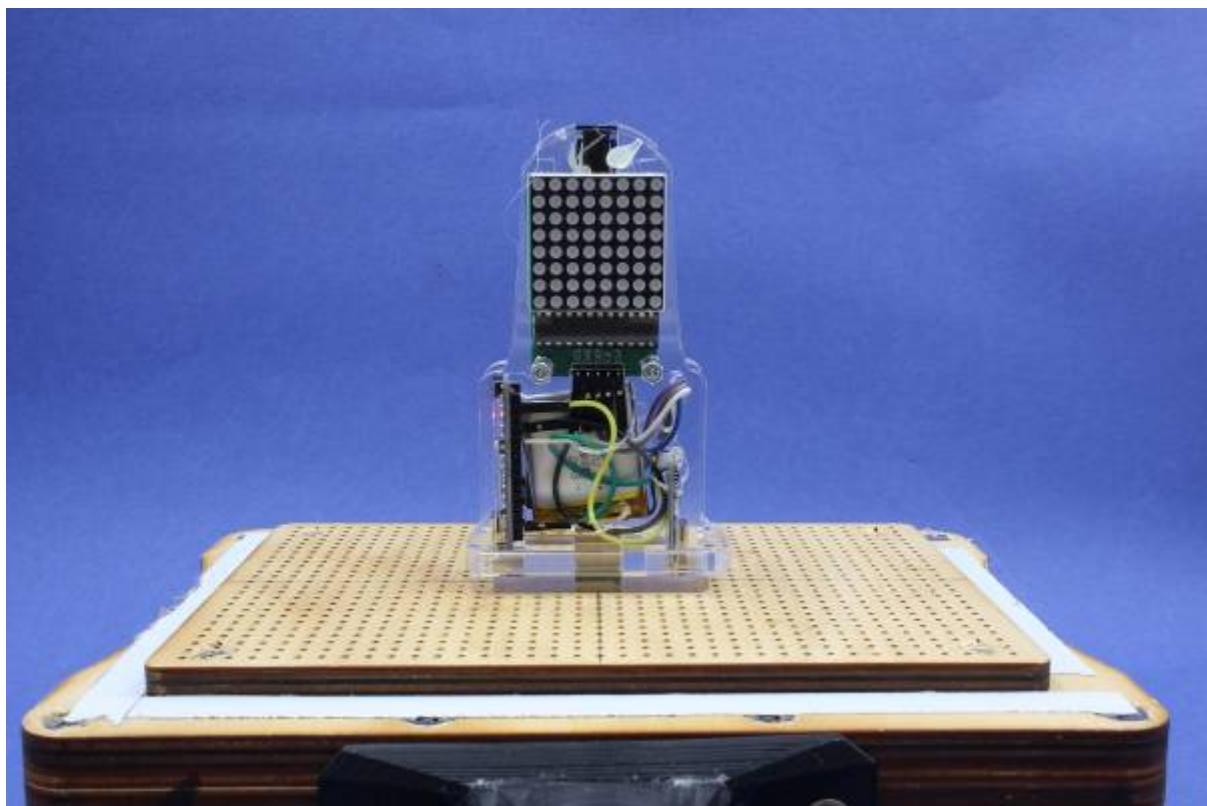
Hot glue it into place



Line up your jig in the centre of the vacuum former plate

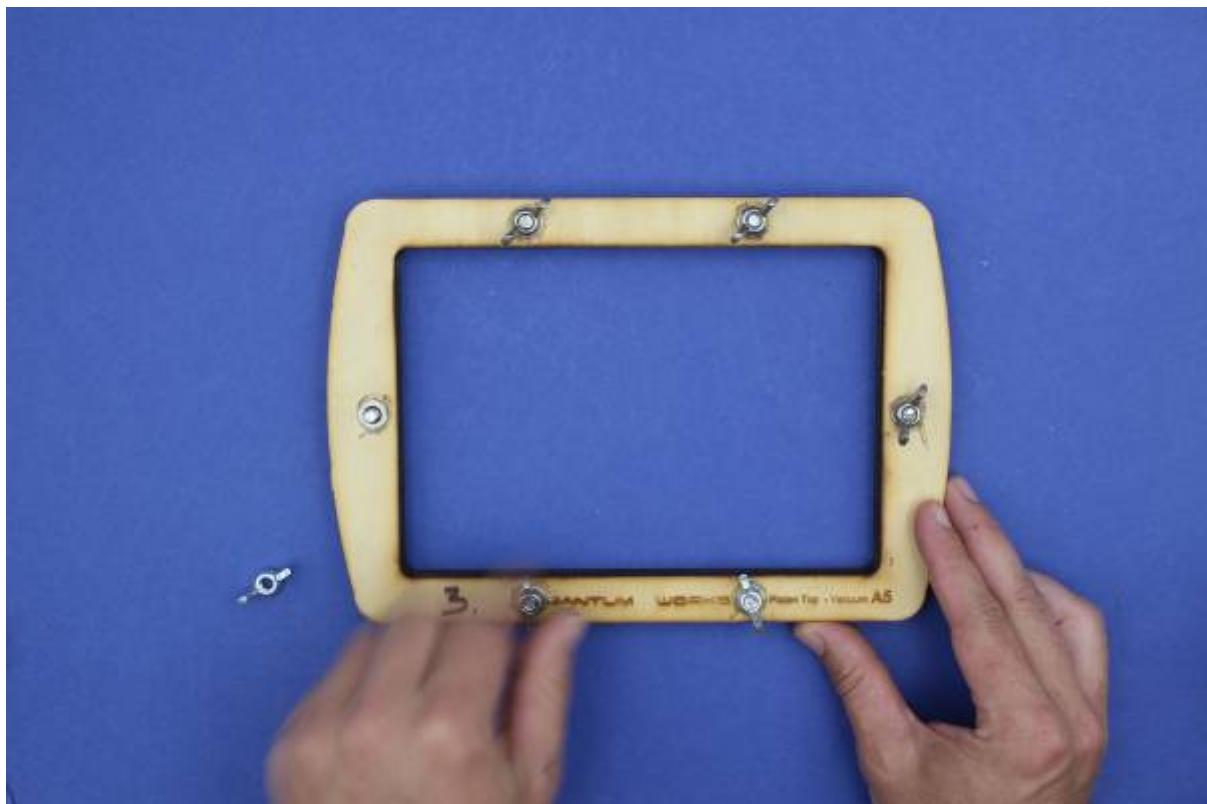


Carefully place your badgeduino on top



Prepare your Plastic

Unscrew the wing-nuts on the vacuum former frame



Fit your sheet of plastic in the frame



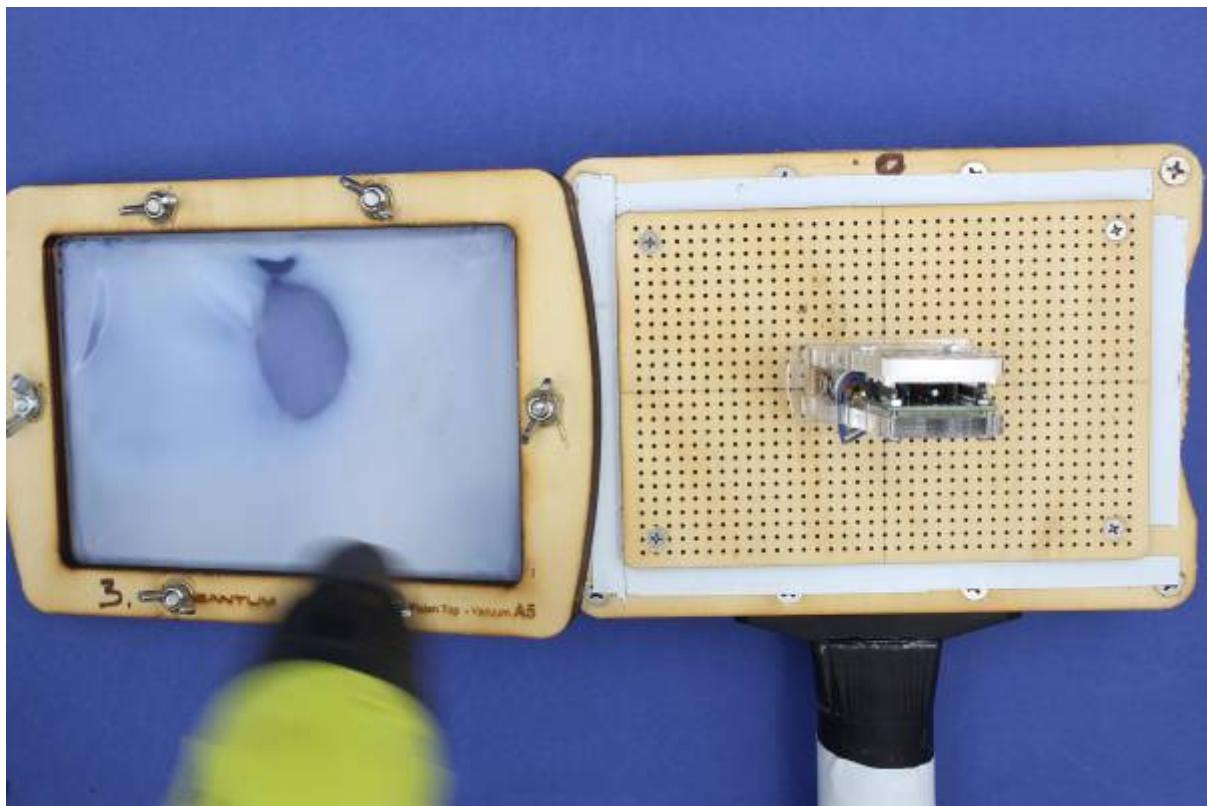
Tighten the wing-nuts firmly



Begin heating your plastic with the hot air gun



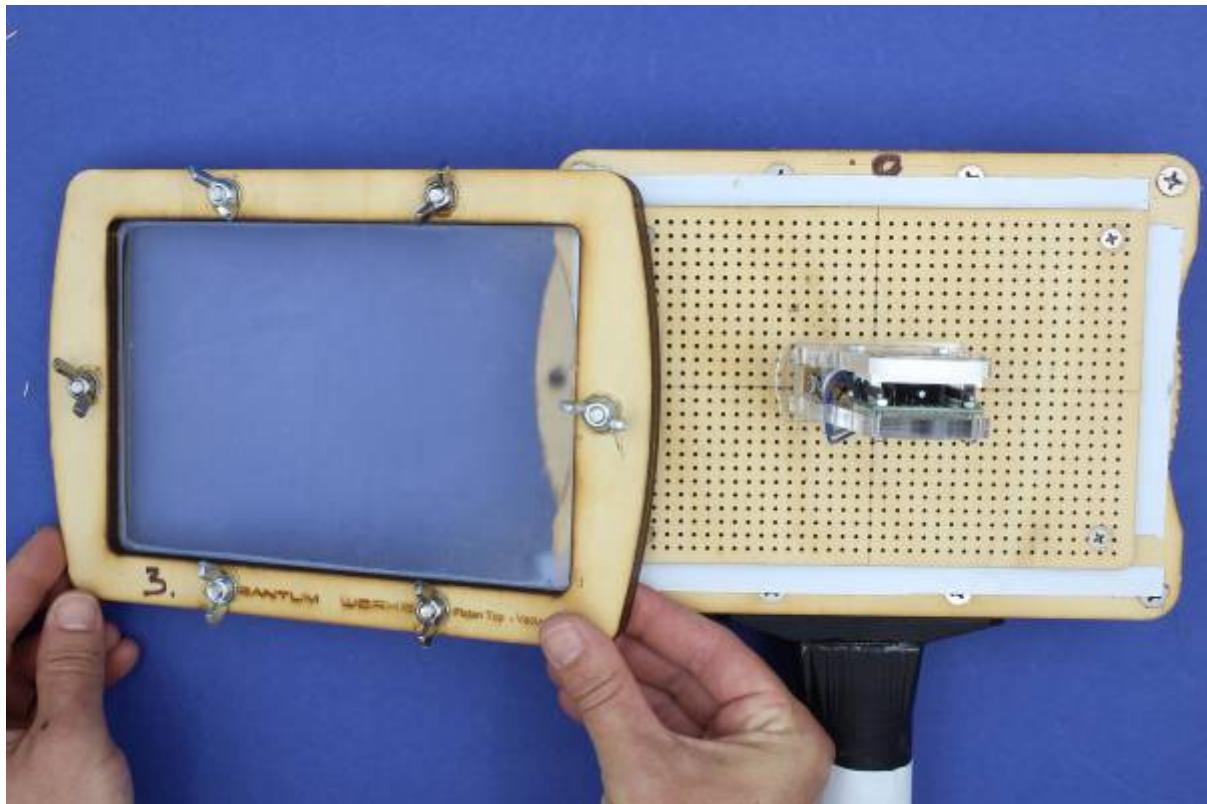
It will slowly start to go clear



Keep heating



When the entire sheet is clear - grab it

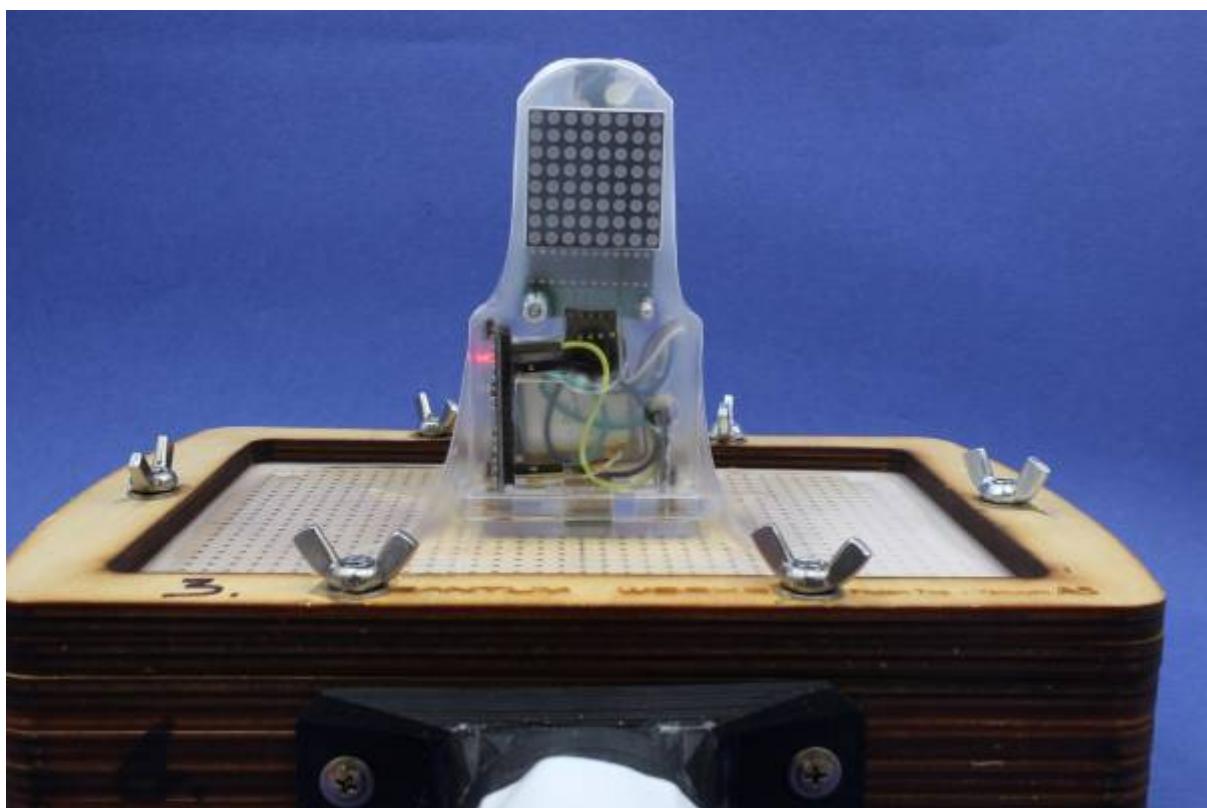


And lower it over your badge

- do it from the top - looking down
- do it quickly! You have about 5 seconds before the plastic starts to cool



Turn on your vacuum**fast**



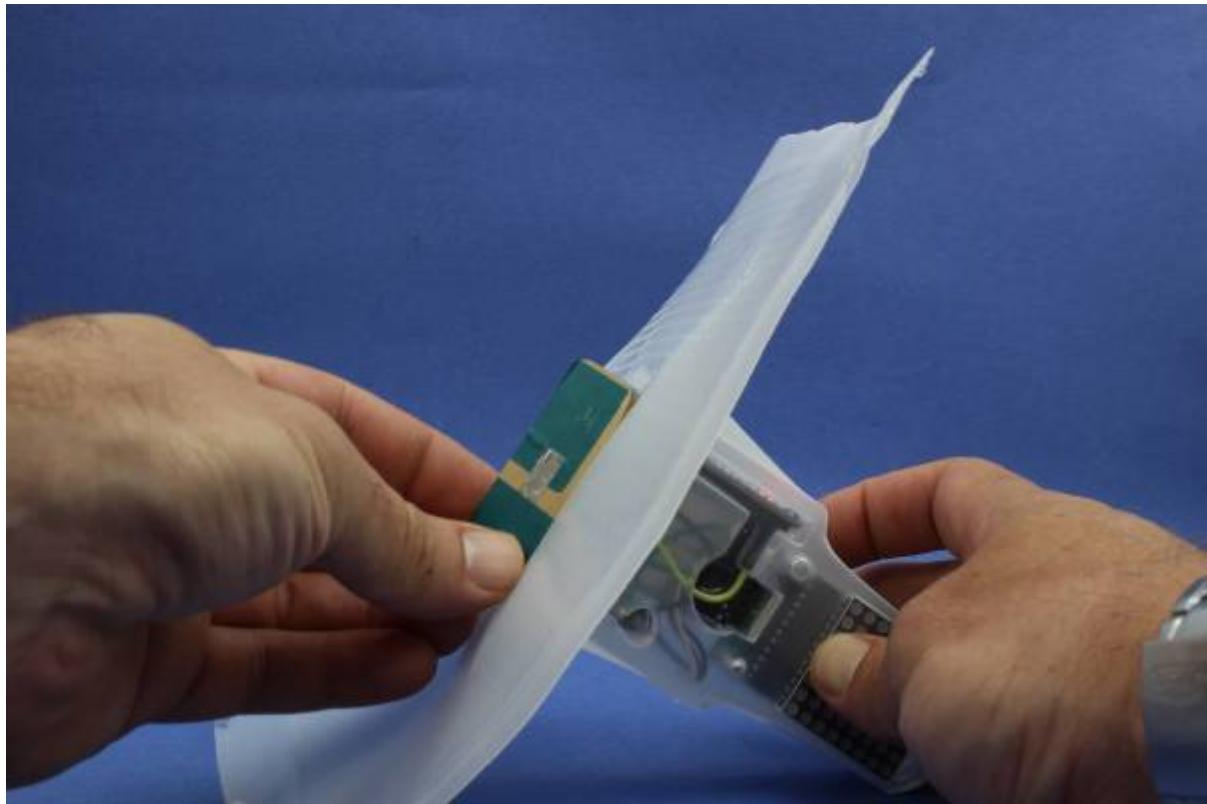
Turn it off when the plastic has gone milky again



Congratulations ! You've made a vacuum form!

Clean up and fit Lanyard

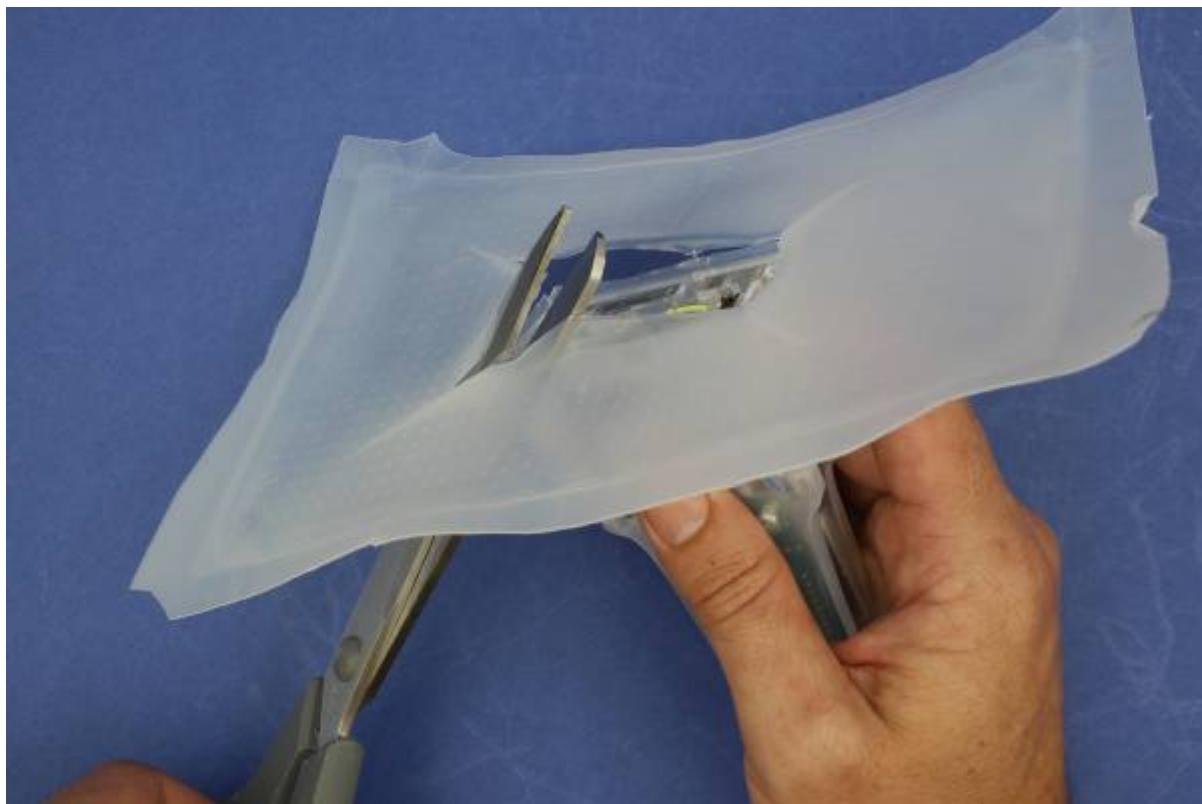
Remove the jig



Carefully cut through the thinned plastic with the craft knife



Cut through the thick fins with your scissors



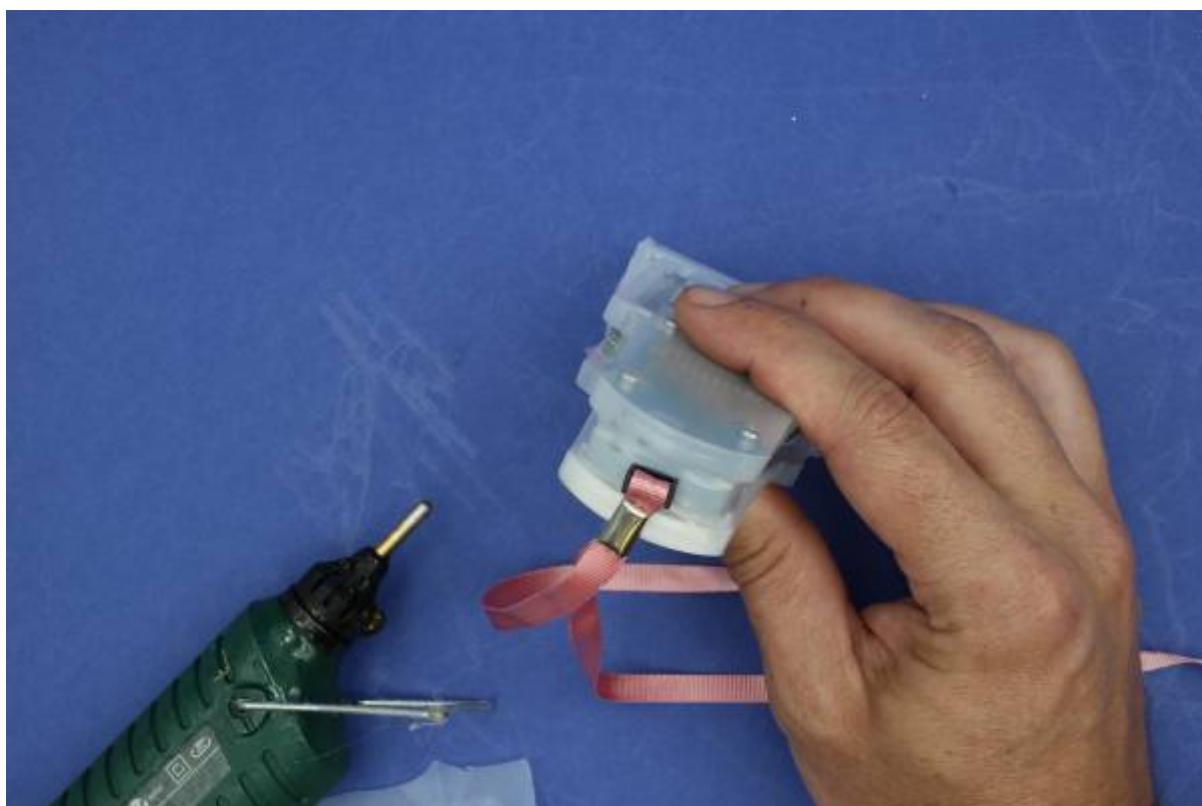
Trim the fins with your knife - always cut **away** from yourself



[Cut free the lanyard socket](#)



Glue the lanyard clip in



Step Seven: Creative PLay

Plug your Badgeduino in to test!



Hacking the Code

While it is beyond the scope of this workshop to teach you how to program your Arduino - here are some tips on:

- changing the displayed text
- displaying a symbol
- changing the speed of the text scrolling

```
void loop()
{
 // displayText(text1, 100); // Send scrolling Text
 displayText(text2, 100); // Send scrolling Text
 //displayCustom(smile01, 1000);
}
```

Look for the `void loop()` section. This contains what your Arduino will do once it turns on. In this case it will:

- ignore `displayText(text1, 100);` as it has been commented out with two backslashes

- `displayText(text2, 100)`
- `ignore displayCustom(smile01, 1000);`

Want to change the displayed text?

- look for `text2`

```
char text2[] = " FREE BEAR HUGS!!!      "; // Scrolling Text
```

- and change it.

```
char text2[] = " I Have Changed!!  "; // Scrolling Text
```

References

- [MaxMatrix Github](#)

Downloads

Arduino

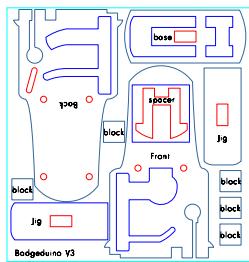
- [Project Files](#)
- [MaxMatrix Library](#)

Laser Cutter Layouts

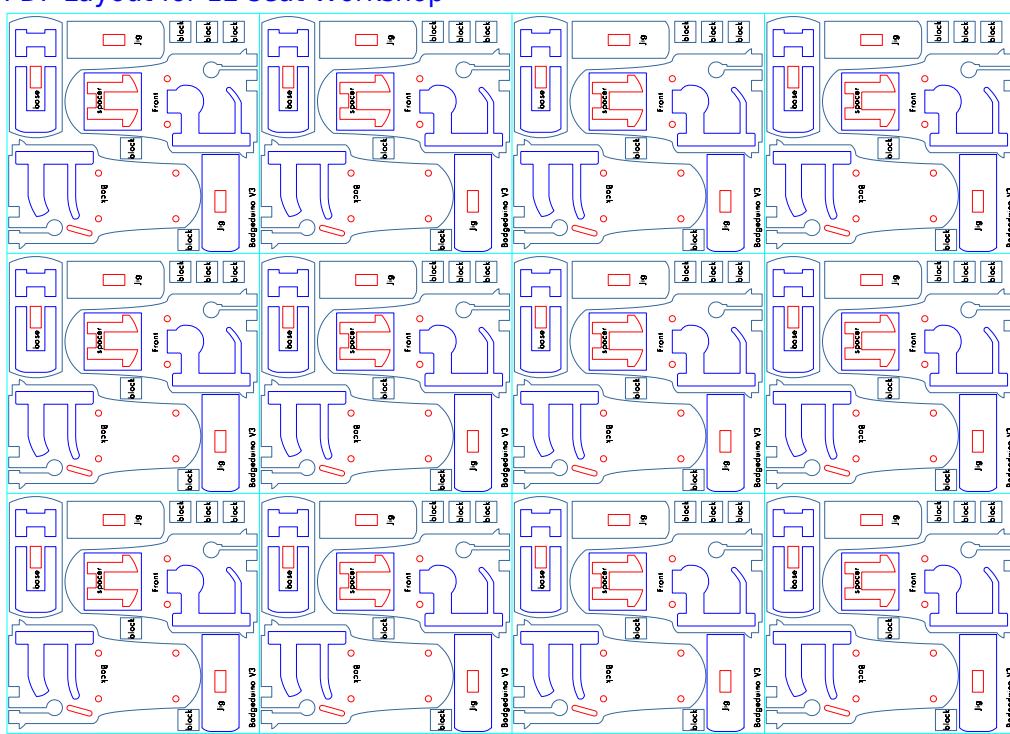
These files are for cutting on The Edge Rayjet. Settings are:

Colour	Power	Speed	Description
Black	100	10	text markings
Red	100	1	smallest shapes
Blue	100	1	inner shapes
Desert Blue	100	1	outer shapes
Cyan	100	1	frame outline

- [badgeduino_v3.cdr](#)
- [badgeduino_v3.dxf](#)
- [badgeduino_v3.pdf](#)



- Corel Layout for 12 seat Workshop
- PDF Layout for 12 seat Workshop



- DXF Layout for 12 seat Workshop

Laser cutter single shapes

These are the single shapes as DXFs

- [back.dxf](#)
- [front.dxf](#)
- [base.dxf](#)
- [spacer.dxf](#)
- [jig.dxf](#)
- [mold_blocks_x5.dxf](#)

Old Versions

- [V2_housing](#)
- [workshop_20180120.zip](#)

1)

https://en.wikipedia.org/wiki/Wearable_technology

2)

<https://en.wikipedia.org/wiki/Arduino>