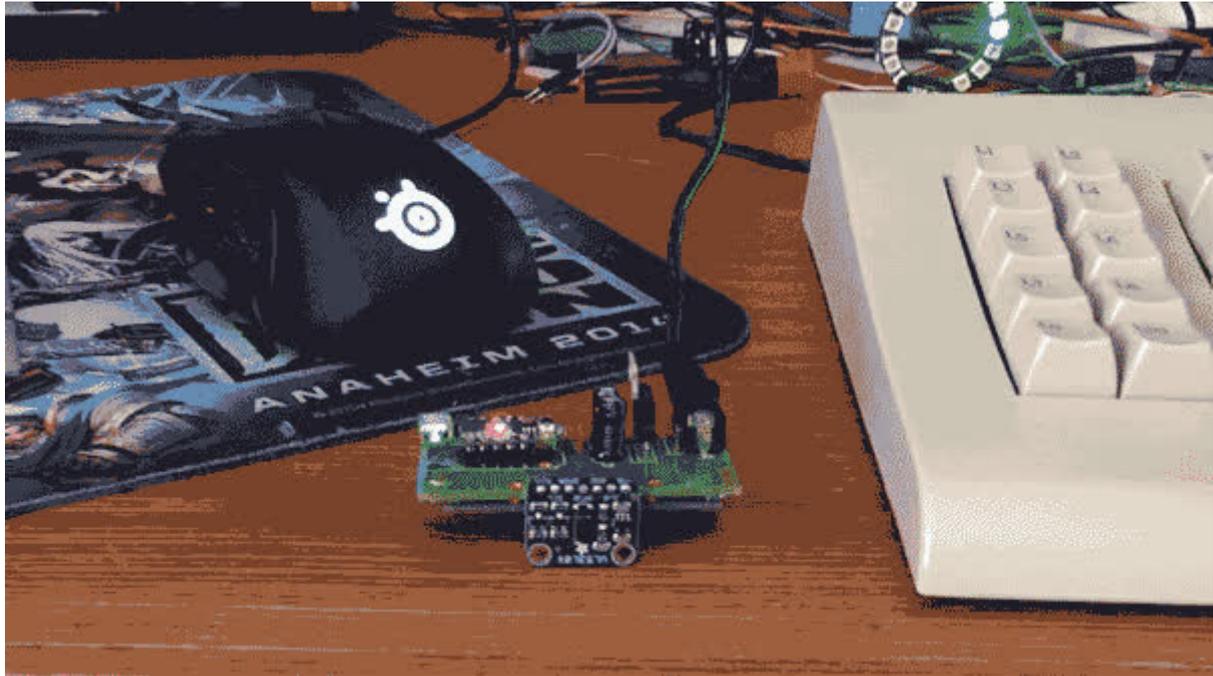




Proximity Based Lighting

Created by Dave Astels



<https://learn.adafruit.com/proximity-based-lighting>

Last updated on 2024-06-03 02:16:12 PM EDT

Table of Contents

Overview	3
• Parts List	
Hardware	4
Prototype Assembly	5
Code	6
PCB Assembly	8
• Getting started	
• Step 1: The proximity breakout header	
• Step 2: The Neopixel strip	
• Step 3: Power	
• Step 4: The Trinket	
• Step 5: The proximity sensor	

String of 8 5050 RGB LED with Integrated Drivers

1 x [NeoPixel Stick](#)

<https://www.adafruit.com/product/1426>

String of 8 5050 RGB LED with Integrated Drivers

1 x [7805 Linear Voltage Regulator](#)

<https://www.adafruit.com/product/2164>

5V 1.5A, Package TO-220

1 x [100uF 16V Electrolytic Capacitor](#)

<https://www.adafruit.com/product/2193>

Pack of 10

1 x [10uF 50V Electrolytic Capacitor](#)

<https://www.adafruit.com/product/2195>

Pack of 10

1 x [2.1mm DC barrel jack](#)

<https://www.adafruit.com/product/373>

Breadboard-friendly

1 x [0.1uF ceramic capacitor](#)

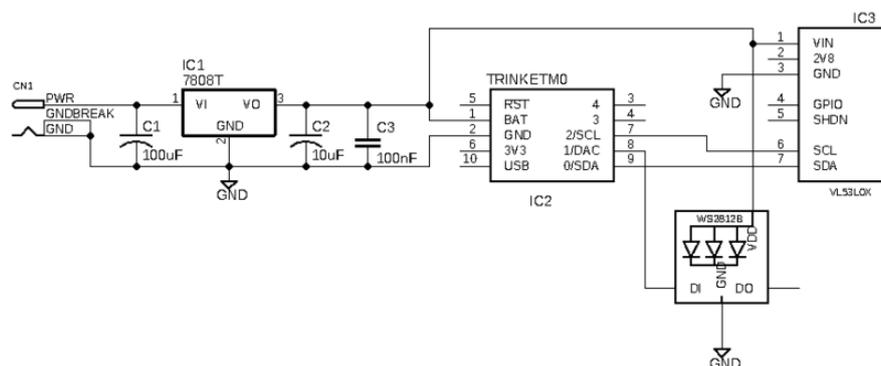
<https://www.adafruit.com/product/753>

10 pack

Hardware

I used a Trinket M0 because it's small, cheap, and yet extremely capable. My current nodes use an ATMEGA328P and I could have done the same here and used an Arduino ProMini but it wouldn't have been much cheaper, would have been larger, and the Trinket lets me write the code in CircuitPython. The Trinket and CircuitPython are perfect for small, simple projects like this.

The circuit is simple (so simple I didn't bother with a schematic or breadboarding before soldering it up): an output to control a small NeoPixel strip, and an I2C proximity sensor. It's an application perfectly suited to a Trinket. Since I plan to run this off a 9v battery or a wall wart like you'd use with an Arduino UNO, I added a power port and 5v regulator. The TO-220 7805 regulator is bulky, but the NeoPixels do draw quite a bit of power.

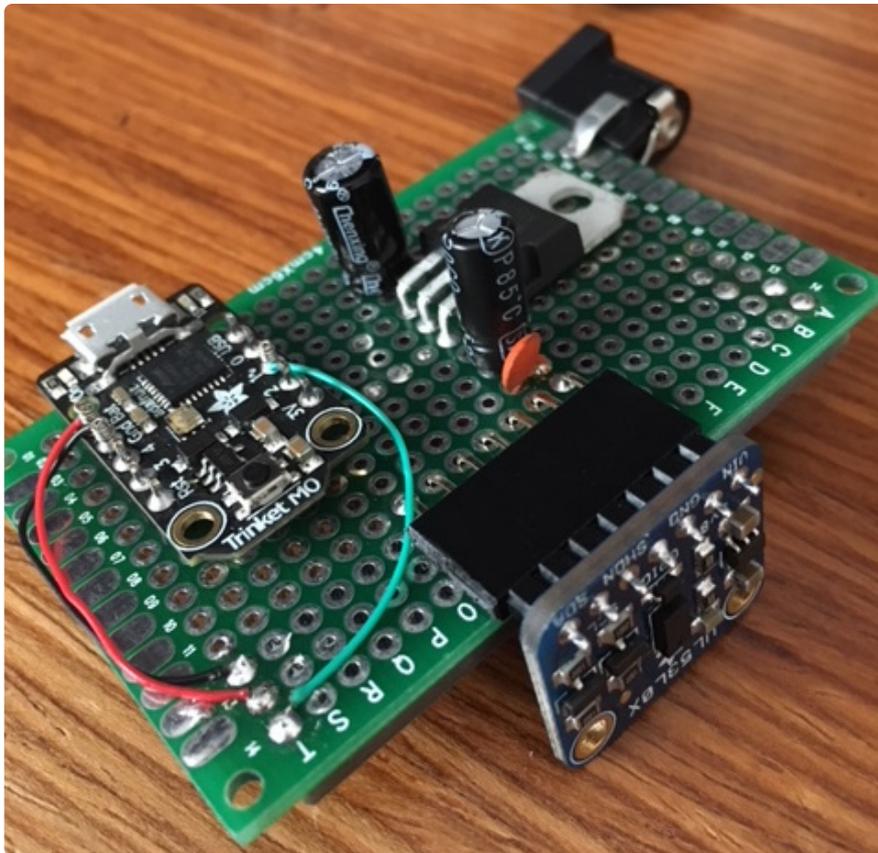


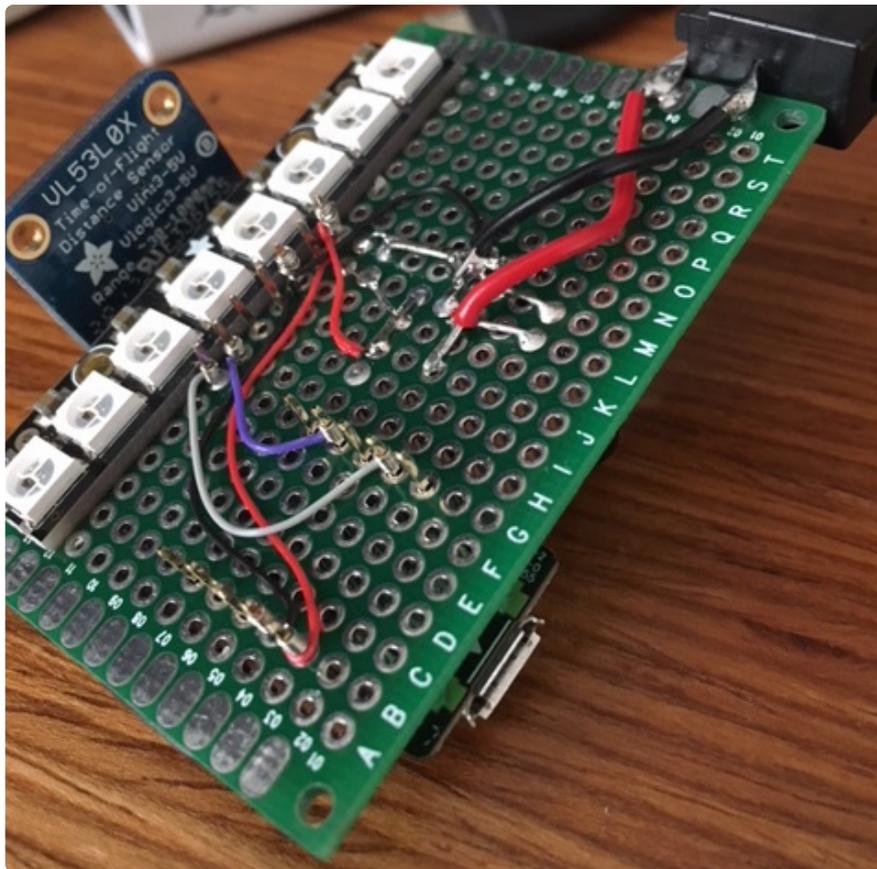
I show a single Neopixel in the schematic, but any chain of Neopixels will work, as long as you don't exceed the available power.

Prototype Assembly

The circuit is simple enough that I hacked it up on a piece of perfboard in about half an hour. I love prototyping on this stuff. It's just holes, but there is a tinned pad on each side of the board and the holes are plated through. It provides lots of options for mounting components and is available in various sizes. I buy it via [Amazon \(https://adafruit.it/BxB\)](https://adafruit.it/BxB).

Since the major components were the Trinket and the distance ranging breakout, which had square mounting pins, I wirewrapped them, but point to point soldered wiring would have been almost as easy.





The NeoPixel strip has its contacts sweat soldered onto tinned holes in the perfboard: the front 3 holes the front of the A and T rows. This not only physically mounted the strip, but also allowed me to connect to it from the top of the board (via the pads and plated holes). The VX53L0X breakout is mounted by using the usual male header strip on the breakout, and plugging it into a matching strip of stacking headers, the legs of which have been bent 90 degrees so that the body lays against the perfboard. This lets the breakout hang down at 90 degrees to the main board, pointing out and away. The whole thing can then be mounted with the NeoPixels facing down and the sensor pointing out.

Code

The code is simple, like the hardware. That's one reason to use CircuitPython: the resulting code can be far simpler than the equivalent code in C++.

Are you new to using CircuitPython? No worries, [there is a full getting started guide here \(https://adafru.it/cpy-welcome\)](https://adafru.it/cpy-welcome).

Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. [You can learn about Mu and installation in this tutorial \(https://adafru.it/ANO\)](https://adafru.it/ANO).

The code needs three libraries to be in the CIRCUITPY drive's `lib` directory (which you may have to create):

- `adafruit_bus_device` (a directory)
- `adafruit_vl53l0x.mpy`
- `neopixel.mpy`

If you haven't installed CircuitPython libraries before, the Welcome to CircuitPython guide has [a page that covers it in detail \(https://adafru.it/ABU\)](https://adafru.it/ABU).

First, the distance sensor and Neopixel strip are set up. The loop continuously checks the distance reading, and if it's close enough (half a meter in my code) the Neopixels are turned on, otherwise they are turned off. I put in a 100 milisecond delay so it wasn't running the distance sensor non-stop.

As I said, the code is simple, and there are plenty of ways to extend it. If used for presence aware lighting effects, you can make use of the color capabilities of the Neopixels. You could set color based on how far away someone is, or change an animation, its speed, or any manner of things.

That's what makes this a fun project: simple hardware, simple code, but plenty of possibilities.

```
# SPDX-FileCopyrightText: 2018 Dave Astels for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import busio
import neopixel
import adafruit_vl53l0x

TRIGGER_DISTANCE = 500          # number of mm where the light toggles
NEOPIXEL_PIN = board.D1
NUMBER_OF_PIXELS = 8

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_vl53l0x.VL53L0X(i2c)

strip = neopixel.NeoPixel(NEOPIXEL_PIN, NUMBER_OF_PIXELS, brightness=1.0,
auto_write=False)

strip.fill((0, 0, 0))
for i in range(NUMBER_OF_PIXELS):
    strip[i] = (64, 64, 64)
    strip.show()
    time.sleep(0.1)
    strip[i] = (0, 0, 0)
    strip.show()

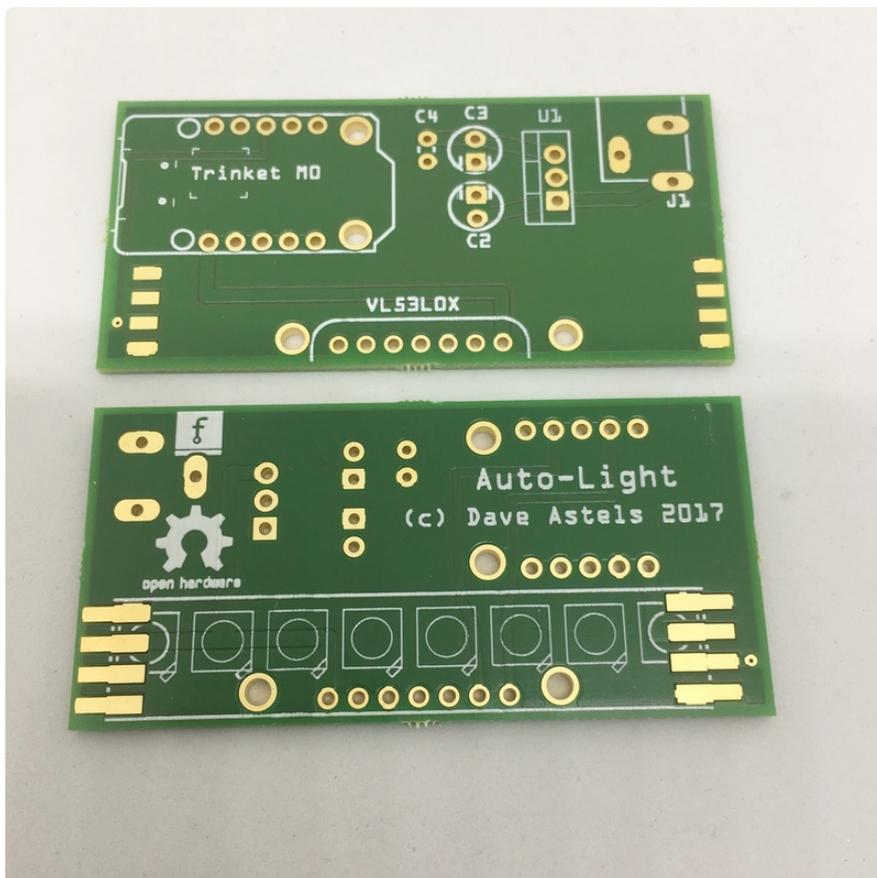
while True:
    if sensor.range < TRIGGER_DISTANCE:
```

```
strip.fill((255, 255, 255))
strip.show()
else:
    strip.fill((0, 0, 0))
    strip.show()
time.sleep(0.1)
```

PCB Assembly

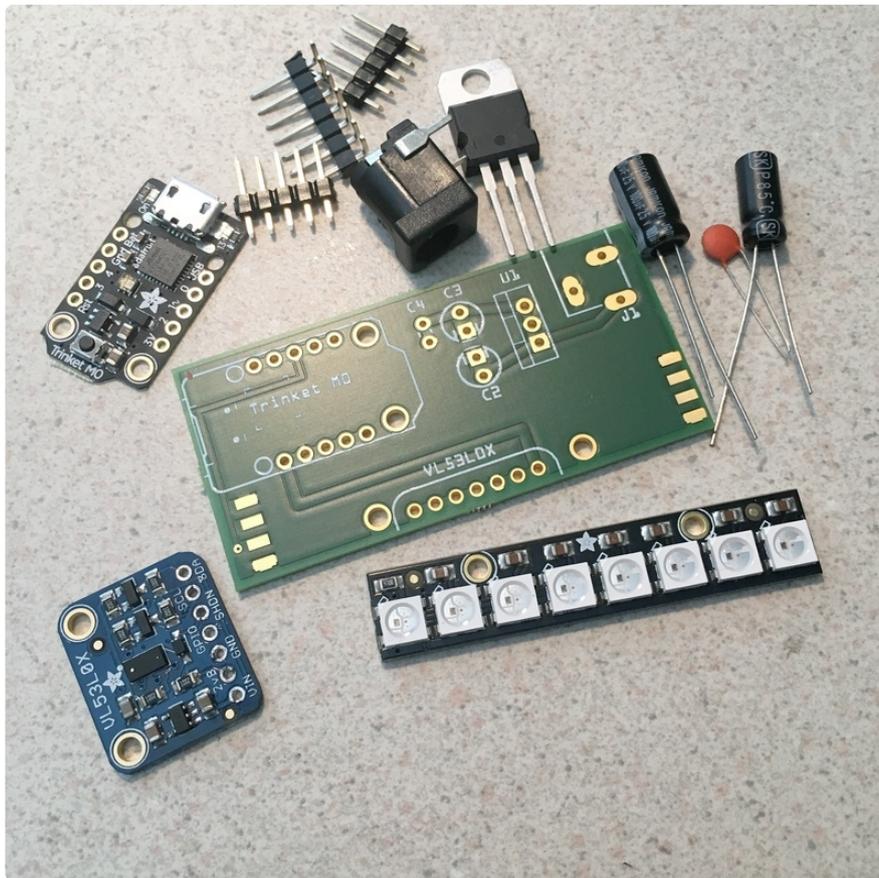
I've designed a PCB for this project and had some made by Aisler (who runs the PCB fabrication for Fritzing now). They make good boards and are a great group of people to deal with.

You can order [boards or a full kit \(https://adafru.it/BJ-\)](https://adafru.it/BJ-) directly from Aisler.



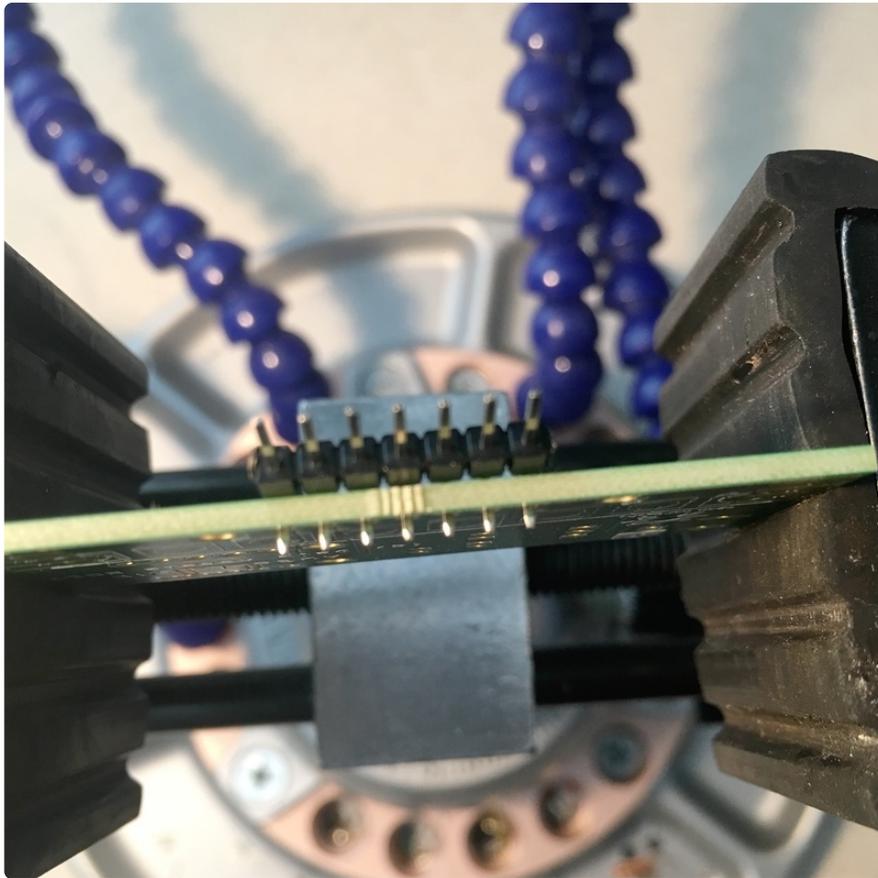
Getting started

Here are all the needed pieces, including cut-to-size pieces of header.

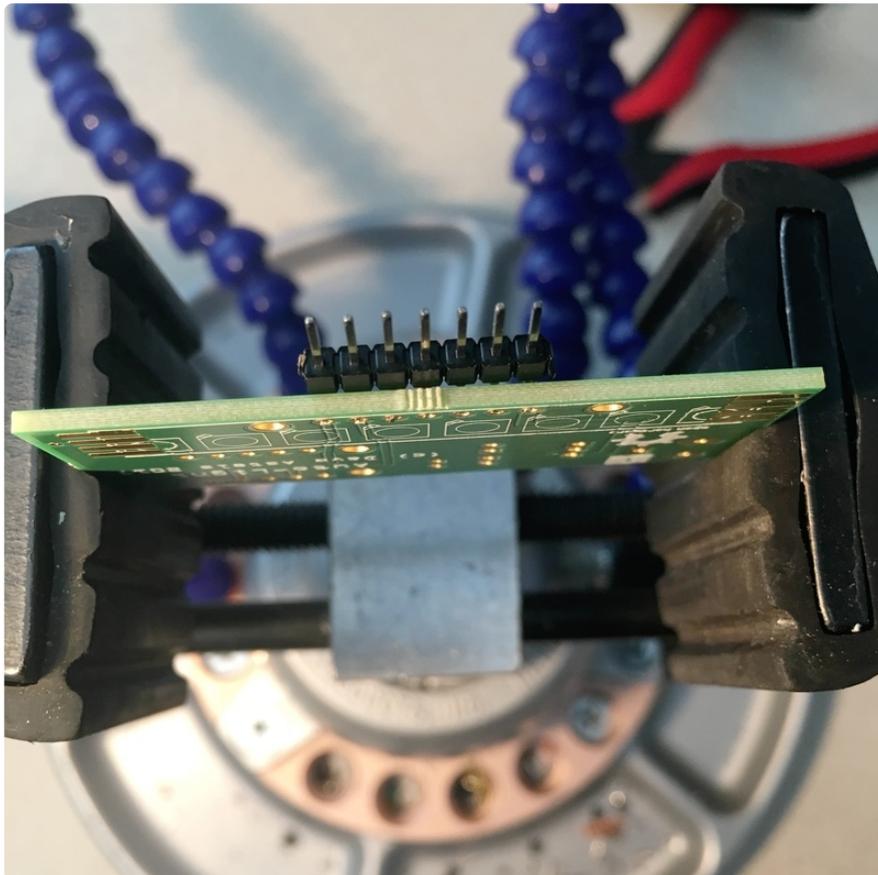
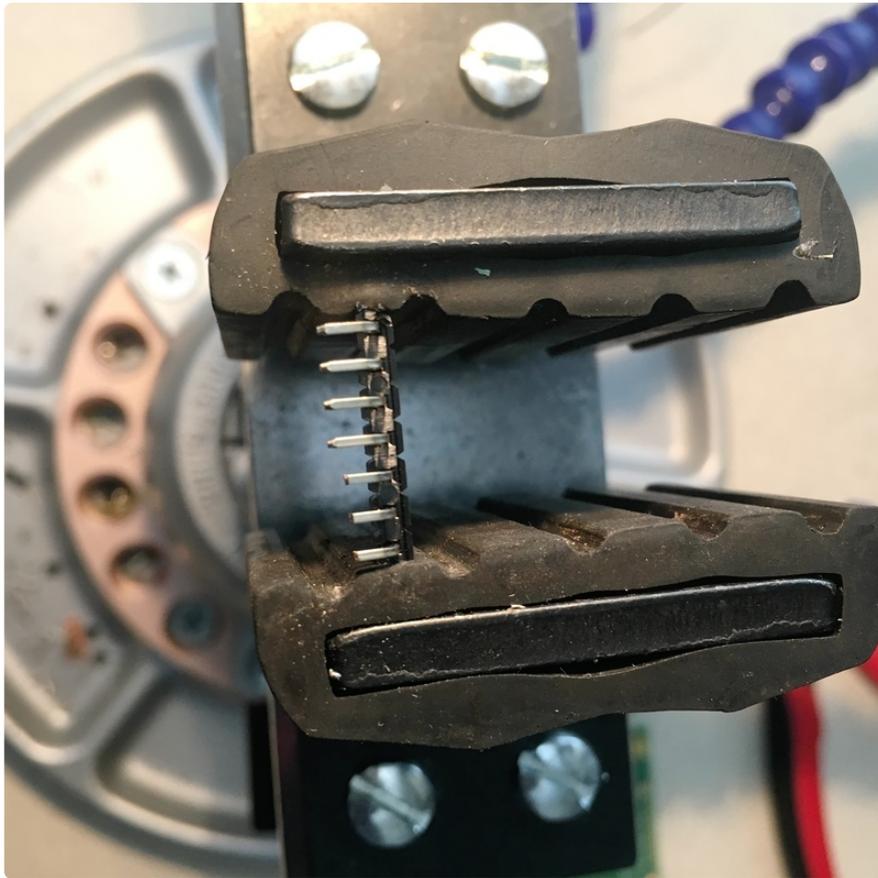


Step 1: The proximity breakout header

This will probably be a bit different than the way you've connected (or will connect) most breakouts. As shown in the prototype page, the proximity sensor is at a right angle to the board. In the final PCB assembly I've used a piece of right angle header to accomplish this.

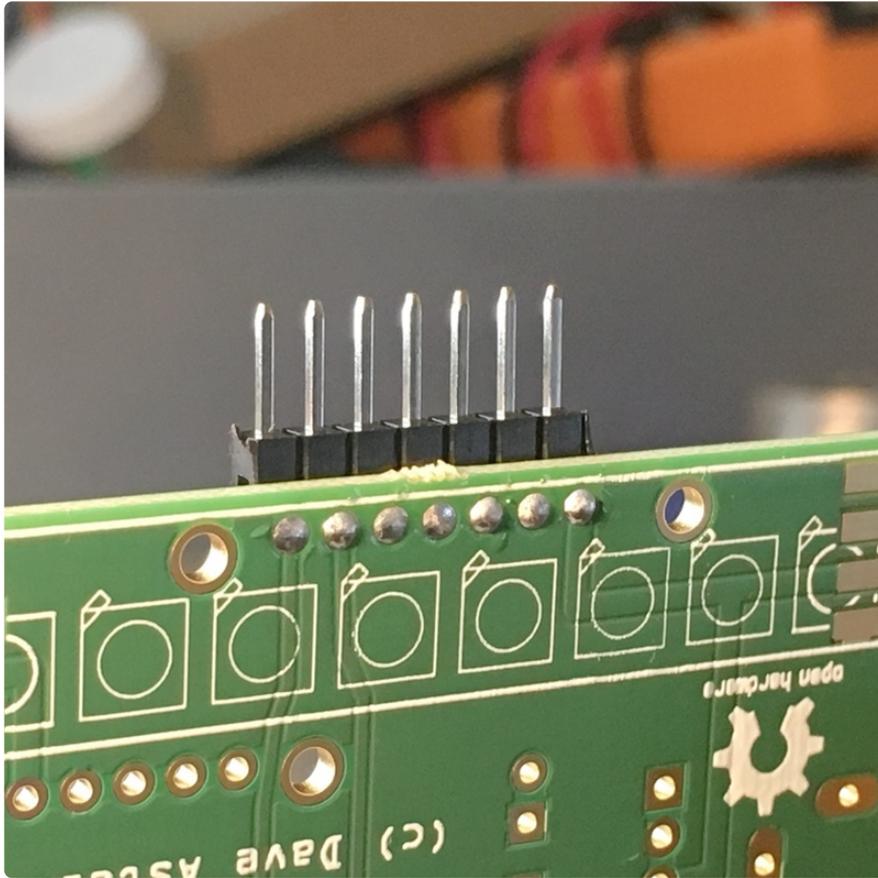


The bent pins go through the board. There's a problem, though. If you look at the board, the Neopixel strip covers the spot where the header pins come through the board. My approach was to trim those pins to be more or less flush with the board.



Now they don't protrude. Soldering is much as usual. Just be sure to heat the pin, a little more attention is required since it isn't sticking through the board. Solder will

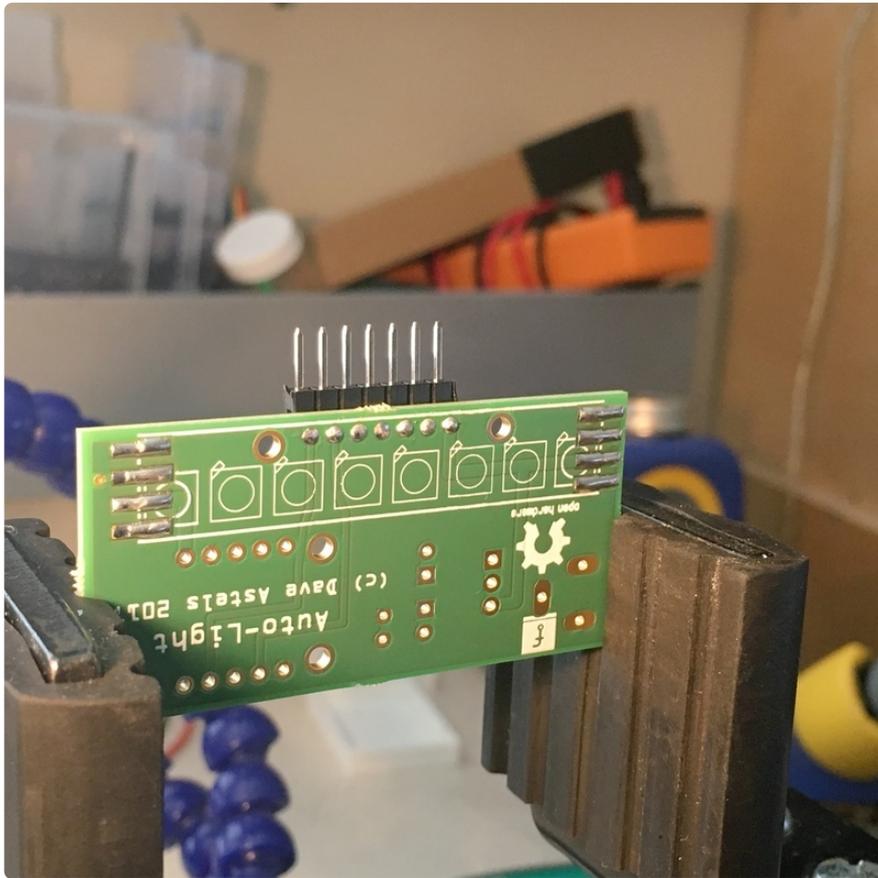
wick into the plated hole and, as long as you don't put on too much solder, will make a nice flush connection. As usual, I solder one pin to tack it in place, then melt that joint and tweak the positions, angle, etc of the header. Once it's how it should be, I solder the rest of the pins.



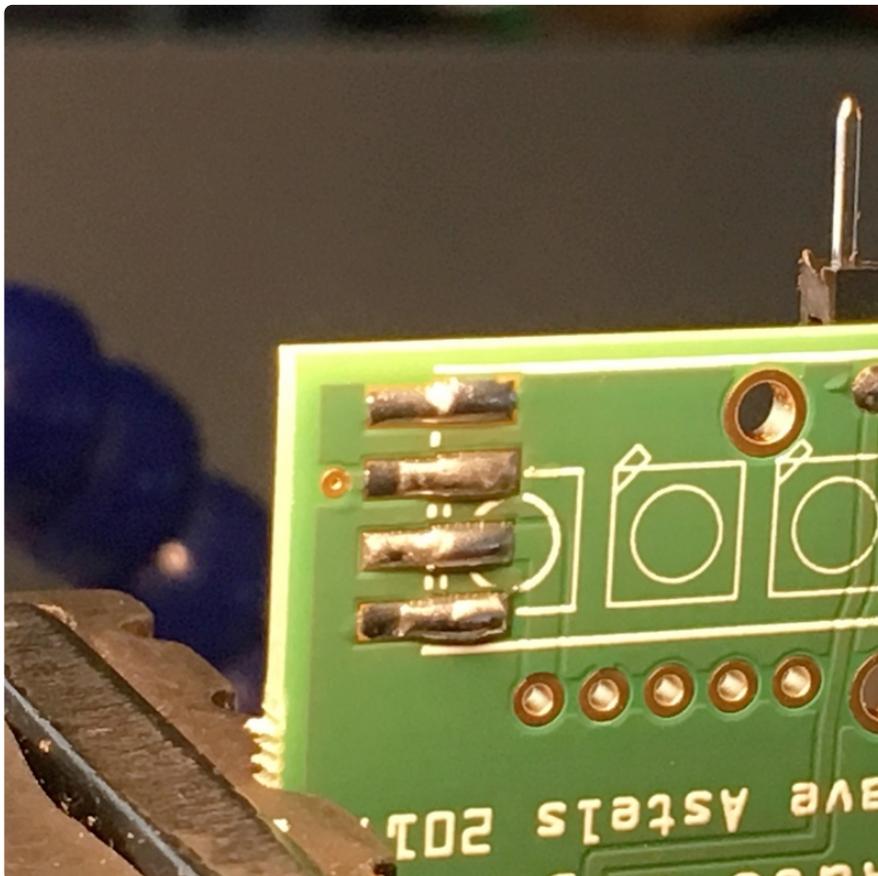
Step 2: The Neopixel strip

This is also a bit different. The Neopixel strips have solder pads on either end. The PCB has matching solder pads. The different thing we're going to do here is not using wires or headers to connect them.

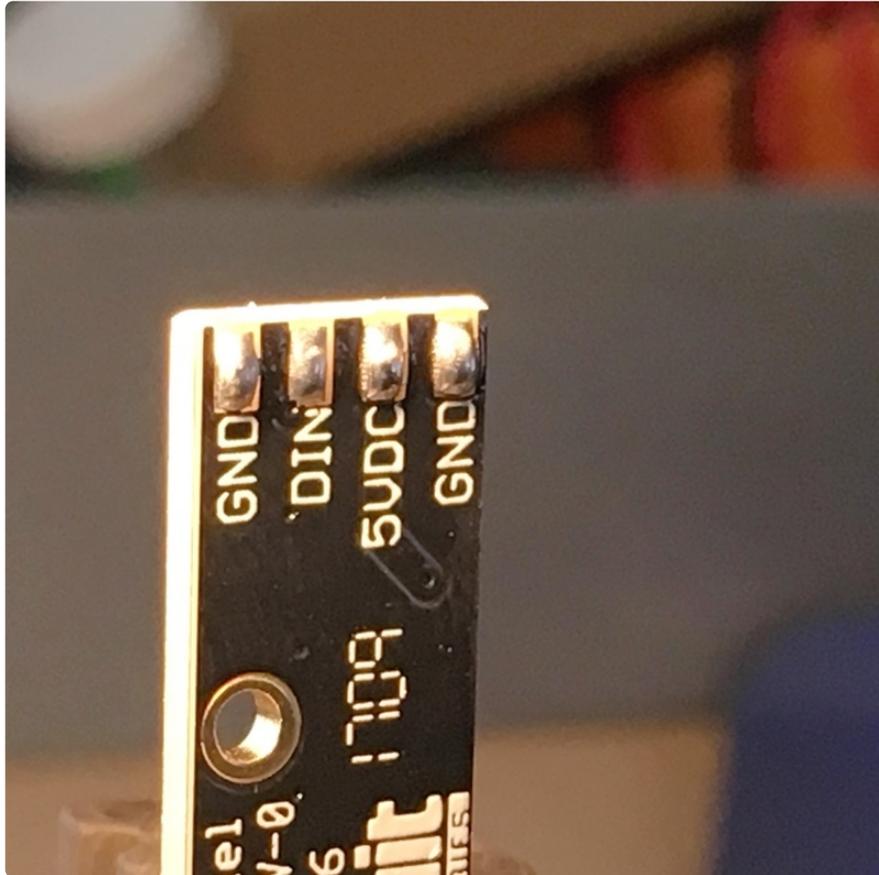
Start by tinning the pads on the PCB with a generous amount of solder.



No, I mean it. Be quite generous.



Next, do the same thing on the pads of the Neopixel strip.

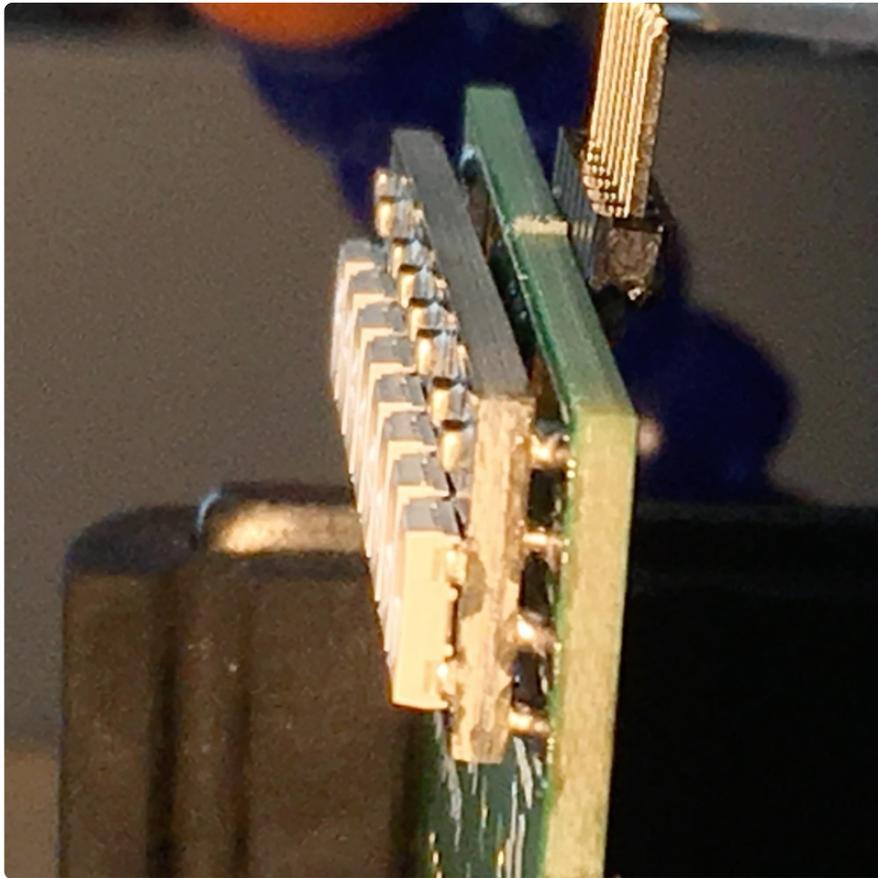


Now the tricky bit. While holding the strip in place against the PCB, solder pad to matching solder pad. Heat one with your iron. Be careful as it will have a tendency to slip such that the pads are not lined up; they'll end up staggered. So hold them together gently.

The goal is to melt the solder blobs on the matching pair of pads simultaneously so that they flow together. Let them cool and you will have a nice connection.

My advice is to do a pair of pads on one end of the strip. Then do a pair of the opposite end. One pad is far easier to redo than several. When you're done, have a close look to verify that the solder on matching pads has, indeed, flowed together. If not, reheat them, adding a bit more solder if necessary. Be careful not to overdo it, though..

Below, you can see the four pairs of pads at one end of the strip nicely connected.



The strip won't be sitting flush on the PCB, but that's fine. All that's required is a solid connection on those eight pairs of pads.

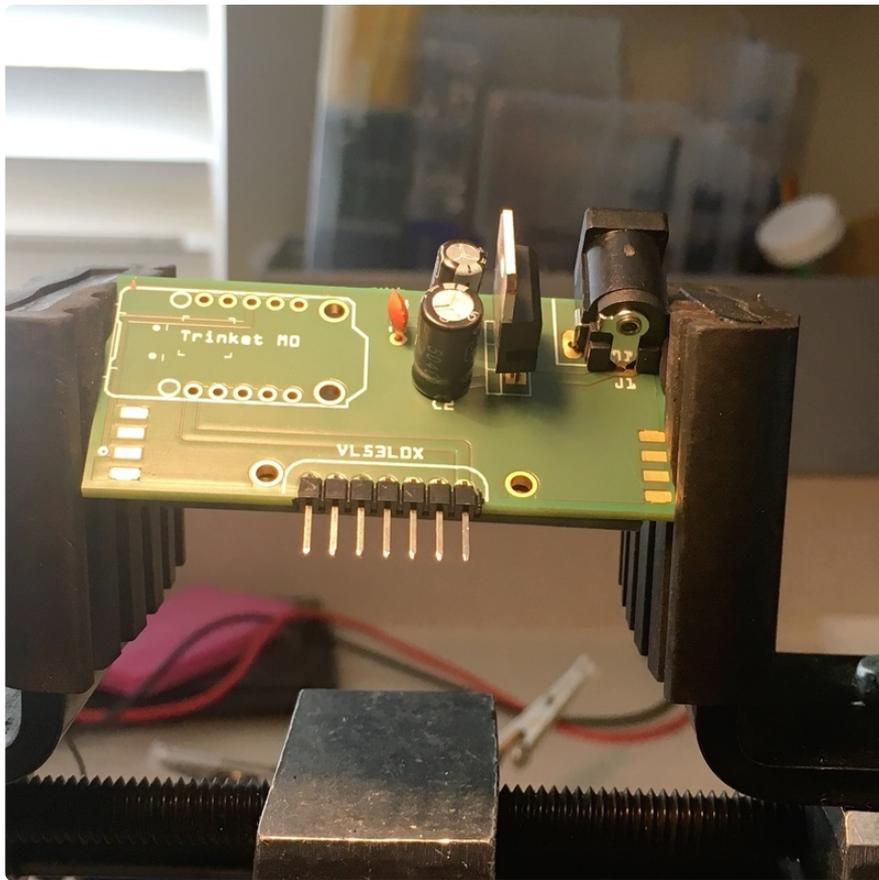
This is why we trimmed the header pins: it lets the strip sit as close as it can to the board. This is also why we started with that header.



Step 3: Power

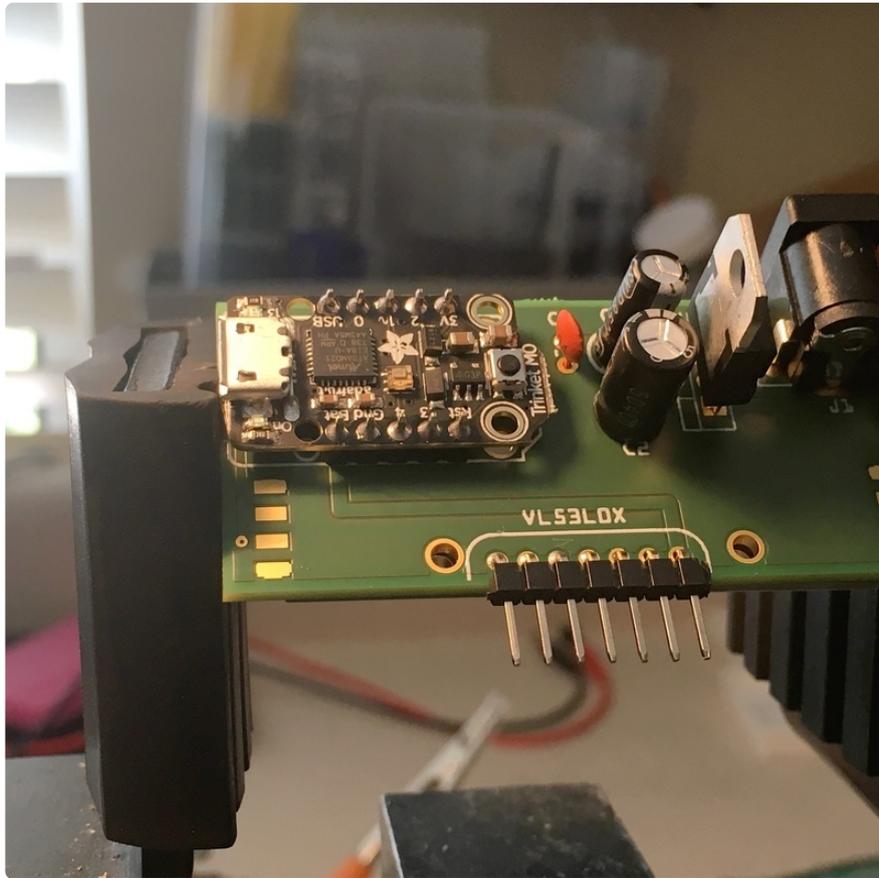
Next, add the components to the power supply section of the board. This includes the power jack, the 7805 5v regulator, and 10uF (C3), 100uF (C2), & 100nF (C4) capacitors. What happened to C1? I have no idea... bitten by Fritzing auto-numbering.

Note that the two larger capacitors are polarized and need to be mounting with the negative leads facing each other. These are well marked, often with a stripe. Similarly the regulator has to go the right way round. The metal heatsink is shown on the board. See the photo below.

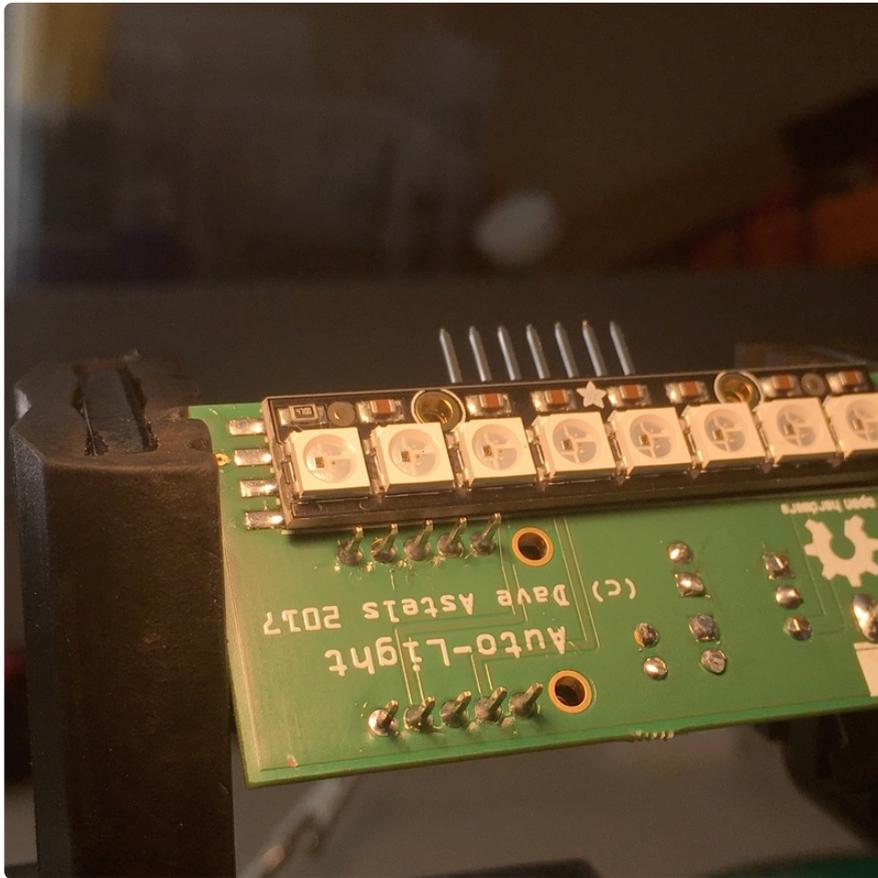


Step 4: The Trinket

Attach the bits of straight header to the Trinket and place it on the board. If you haven't done this sort of thing before, have a look in the [Feather M0 Express tutorial for advice/instruction \(https://adafru.it/B4b\)](https://adafru.it/B4b). It's the same for the Trinket, the only different is the exact position and size of the header strips.

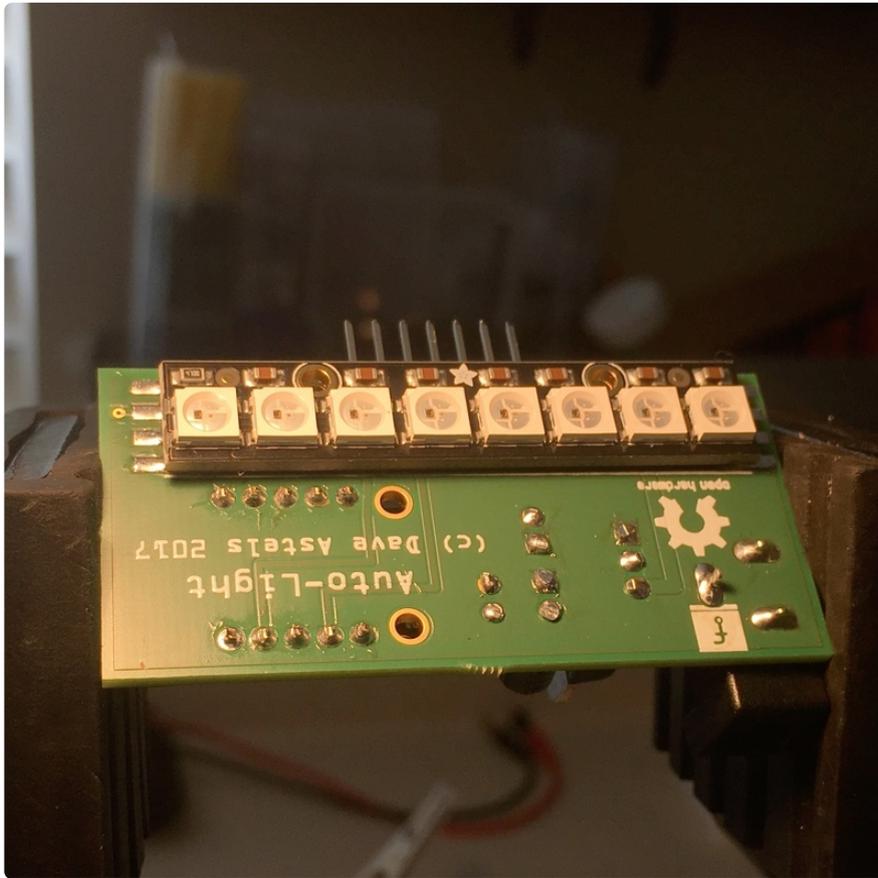


Flip the board over and solder the Trinket's pins. Again, I always start with one and make sure everything is how it should be. Then I do the opposite corner, making sure the entire thing is right. Then I do the rest of the pins.



Whether you trim the pins is up to you. My thought is to design a printable case for this at some point, so I trimmed the pins so they won't be in the way.

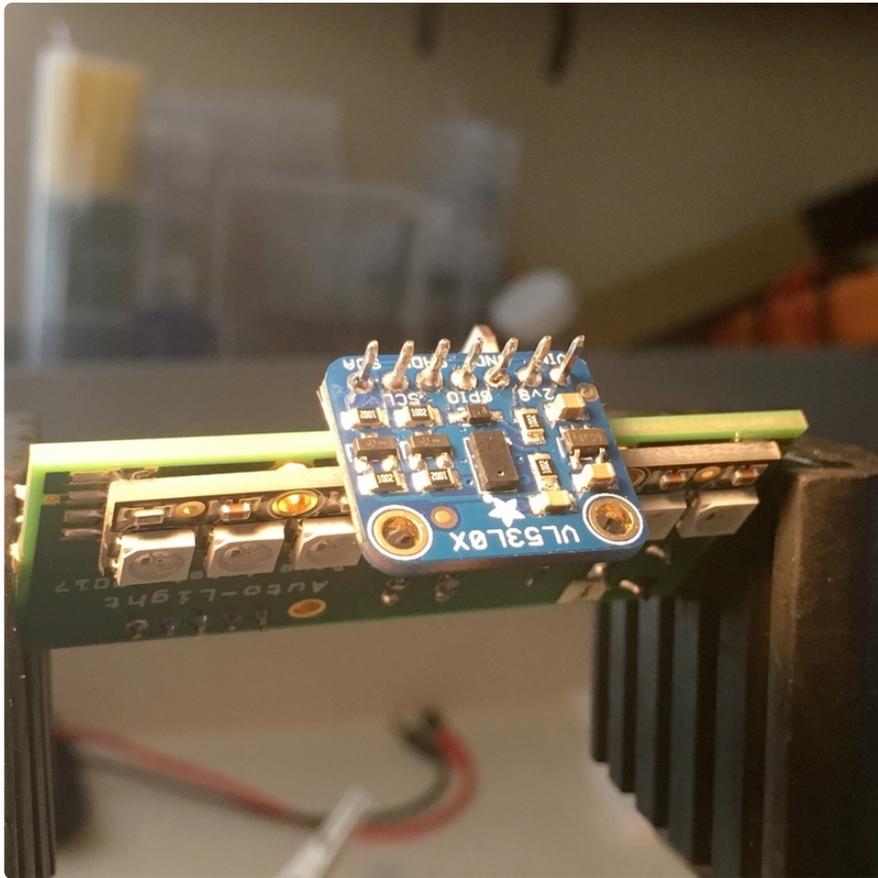
Just a note: I keep an older pair of side cutters for this. The blade alignment is a bit off and I replaced them a while ago with a more expensive pair. I keep my good pair for cutting fine wire (30 or 24 gauge usually) and use the older pair for rougher work. Trimming header pins classifies as rougher work. So as you replace and/or upgrade your tools, hold onto the old ones unless they are broken.



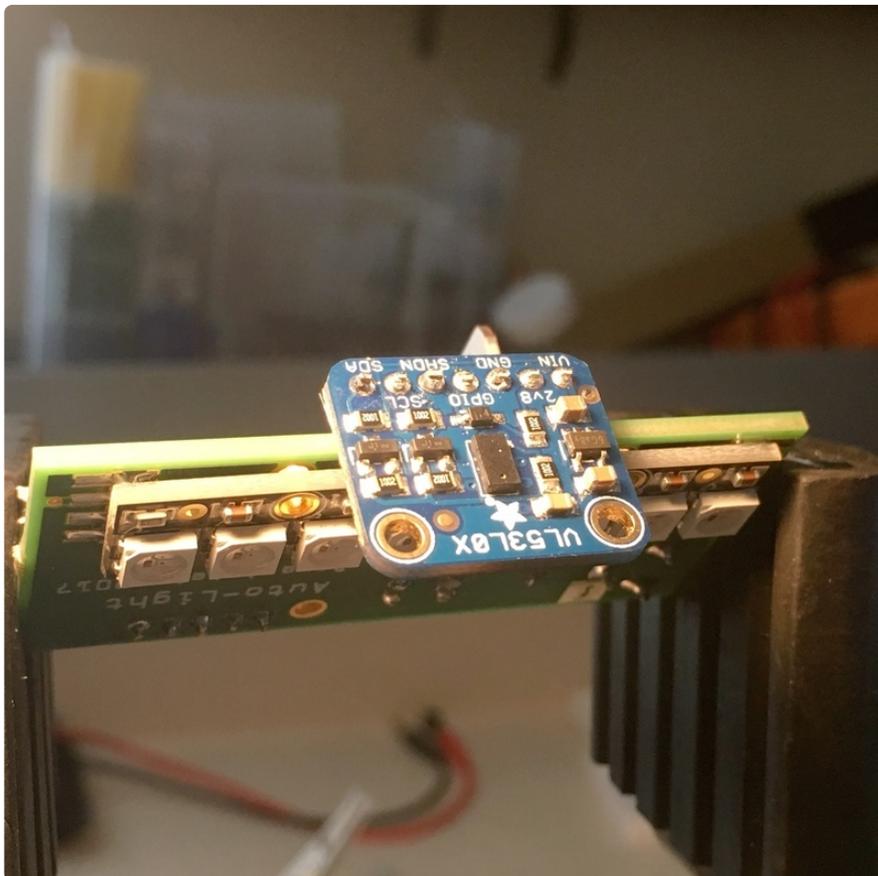
Step 5: The proximity sensor

The final step is mounting the VL53L0X breakout on the header strip we started out with.

Slide the breakout on the header pins, solder one pin, adjust the breakout (you want a nice 90 degree angle with the PCB) and solder up the rest.



Once again, trim the excess pin length if you wish.



That's it. Time to load the CircuitPython runtime, libraries, and code.py.