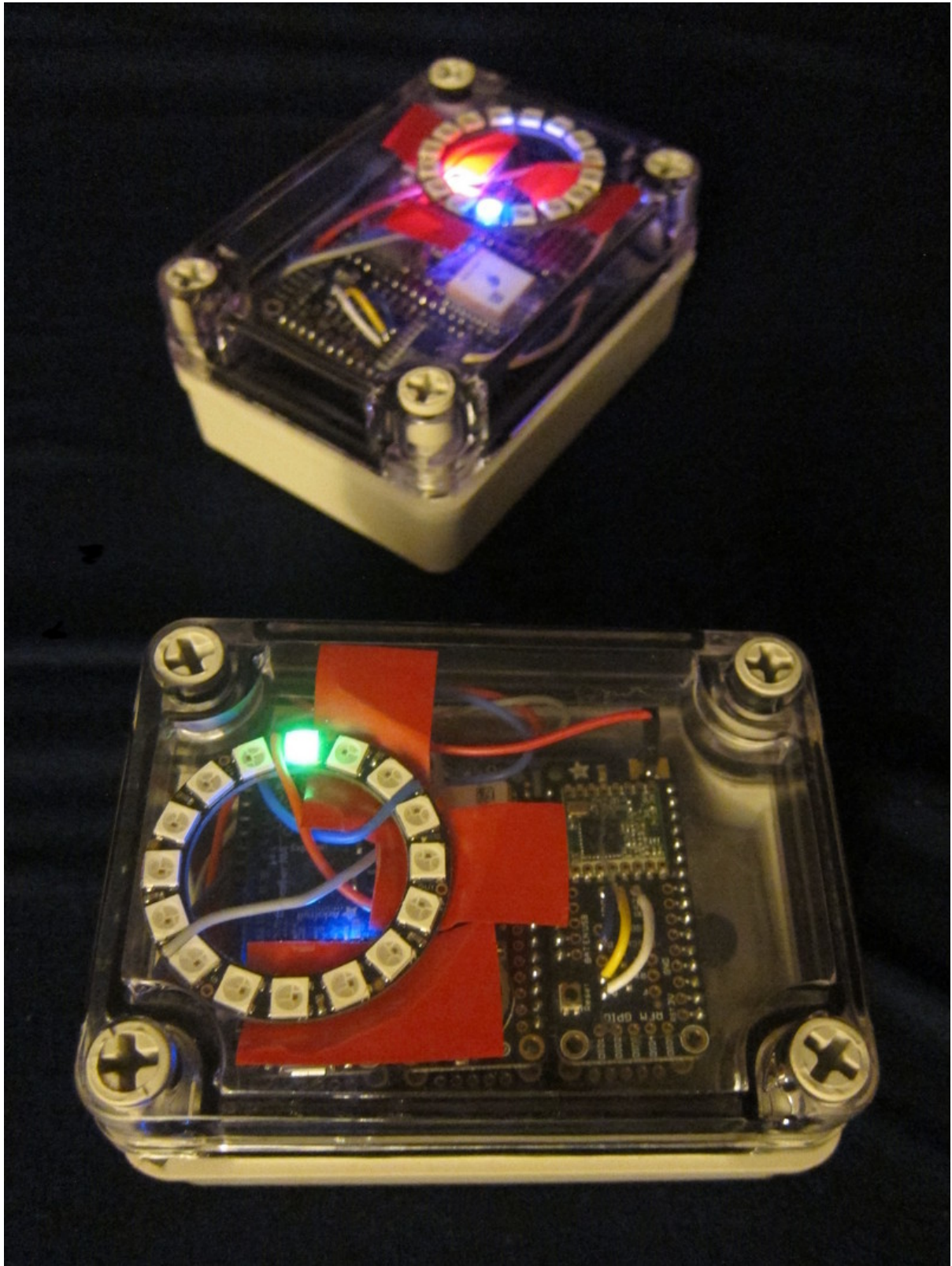




# Glitter Positioning System

Created by Brennen Bearnes



<https://learn.adafruit.com/glitter-positioning-system>

Last updated on 2024-06-03 02:28:18 PM EDT

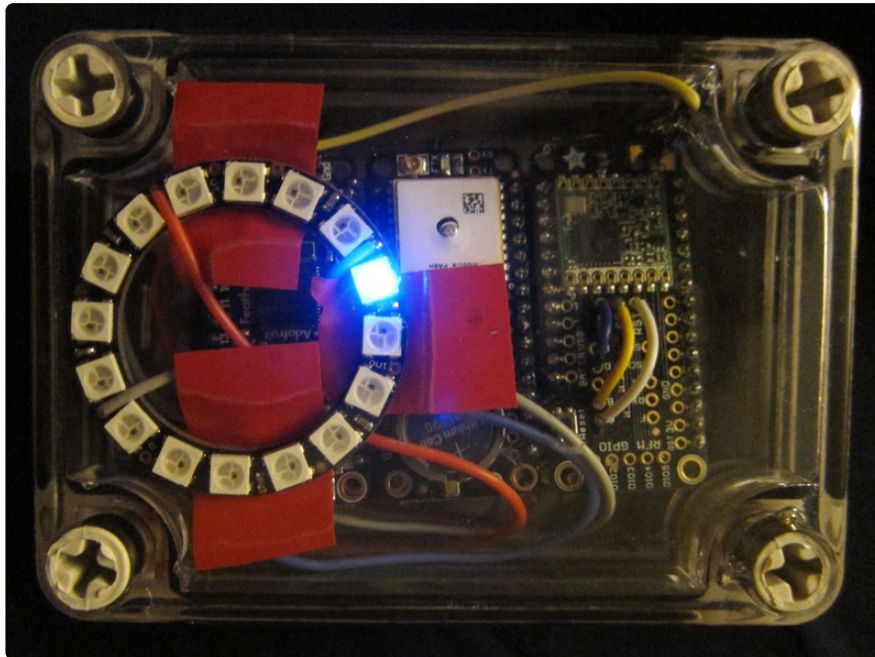
# Table of Contents

<b>Overview</b>	<b>5</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Parts List</a></li><li>• <a href="#">Wiring &amp; Enclosure</a></li></ul>	
<b>Assembly</b>	<b>8</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">FeatherWing Tripler</a></li><li>• <a href="#">Feather M4 Express</a></li><li>• <a href="#">Ultimate GPS FeatherWing</a></li><li>• <a href="#">LoRa Radio FeatherWing</a></li><li>• <a href="#">NeoPixel Ring &amp; LSM9DS1 9-DOF</a></li><li>• <a href="#">LSM9DS1</a></li><li>• <a href="#">NeoPixel Ring</a></li><li>• <a href="#">Final Assembly &amp; Enclosure</a></li></ul>	
<b>CircuitPython Code</b>	<b>19</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Get the Code</a></li><li>• <a href="#">Configuration</a></li><li>• <a href="#">Compass Calibration</a></li><li>• <a href="#">Magnetic North</a></li></ul>	
<b>Usage, Customization, and Next Steps</b>	<b>24</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Display Customization</a></li><li>• <a href="#">Potential Hardware Improvements</a></li><li>• <a href="#">Acknowledgments &amp; Further Reading</a></li></ul>	



---

# Overview



I'm a regular attendee at festivals that involve setting up a camp and then wandering around a large area with friends. I also have friends who are prone to wandering off from the group in the middle of the night and getting confused about their own whereabouts.

It seemed natural to address this somewhat ridiculous problem with a somewhat ridiculous technological solution. Enter the Glitter Positioning System: A collection of boxes which uses NeoPixels to point in the rough direction of the other boxes.

Each box contains a Feather M4 Express, GPS, a LoRa radio for transmitting and receiving coordinates, a magnetometer, and a 16-NeoPixel ring for display purposes. This is a simple proof of concept, but with the Feather M4's processing power and ample room for CircuitPython code, it makes for a capable and flexible platform for experimenting with navigation and simple radio communication.

## Parts List

**For each person** (at least 2 required!) you'll need:

peripherals for support, this will for sure be your favorite new chipset.

#### **1 x [Adafruit Feather M4 Express](https://www.adafruit.com/product/3857)**

<https://www.adafruit.com/product/3857>

This feather is powered by our new favorite chip, the ATSAM51J19 - with its 120MHz Cortex M4 with floating point support and 512KB Flash and 192KB RAM. Your code will zig and zag and zoom, and with a bunch of extra peripherals for support, this will for sure be your favorite new chipset.

---

#### **1 x [Adafruit LoRa Radio FeatherWing - RFM95W 900 MHz](https://www.adafruit.com/product/3231)**

<https://www.adafruit.com/product/3231>

Add short-hop wireless to your Feather with these RadioFruit Featherwings.

---

#### **1 x [Adafruit Ultimate GPS FeatherWing](https://www.adafruit.com/product/3133)**

<https://www.adafruit.com/product/3133>

Give your Feather a sense of place, with an Ultimate GPS FeatherWing.

---

#### **1 x [FeatherWing Tripler Mini Kit - Prototyping Add-on for Feathers](https://www.adafruit.com/product/3417)**

<https://www.adafruit.com/product/3417>

This is the FeatherWing Tripler - a prototyping add-on and more for all Feather boards. Connect your Feather to two other Feather Wings without needing any stacking headers!

---

#### **1 x [Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS1](https://www.adafruit.com/product/3387)**

<https://www.adafruit.com/product/3387>

Add motion, direction and orientation sensing to your project with this all-in-one 9-DOF sensor.

---

#### **1 x [Neopixel Ringer - 16 x 5050 RGB LED with Integrated Drivers](https://www.adafruit.com/product/1463)**

<https://www.adafruit.com/product/1463>

Round and round and round they go! 16 ultra bright smart LED NeoPixels are arranged in a circle with 1.75" (44.5mm) outer diameter.

---

#### **1 x [Lithium Ion Battery - 3.7v 2000mAh](https://www.adafruit.com/product/2011)**

<https://www.adafruit.com/product/2011>

Lithium ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light and powerful. The output ranges from 4.2V when completely charged to 3.7V. This battery has a capacity of 2000mAh.

---

## Wiring & Enclosure

You'll also need hookup wire, and an enclosure for each unit. I used silicone-cover stranded-core wire, because it's flexible, tough, and easy to fit into tight enclosure spaces. A waterproof plastic enclosure with a clear top keeps the electronics free of moisture and dust, and allows the NeoPixel ring display to be seen easily.

**1 x Silicone Cover Stranded-Core Wire - 2m 26AWG** <https://www.adafruit.com/product/1877>  
**Red**

Silicone-sheathing wire is super-flexible and soft, and its also strong!

---

**1 x Silicone Cover Stranded-Core Wire - 2m 26AWG** <https://www.adafruit.com/product/1881>  
**Black**

Silicone-sheathing wire is super-flexible and soft, and its also strong!

---

**1 x Silicone Cover Stranded-Core Wire - 2m 26AWG** <https://www.adafruit.com/product/1878>  
**Blue**

Silicone-sheathing wire is super-flexible and soft, and its also strong!

---

**1 x Silicone Cover Stranded-Core Wire - 2m 26AWG** <https://www.adafruit.com/product/1880>  
**Green**

Silicone-sheathing wire is super-flexible and soft, and its also strong!

---

**2 x Small Plastic Project Enclosure - Weatherproof with Clear Top** <https://www.adafruit.com/product/903>

Store your project safe and sound in this nice weatherproof box with a clear top.

---

Within the enclosure, I also used:

- A small piece of acrylic to shield the battery
- Self-adhesive Velcro strips to hold components in place
- Electrical tape

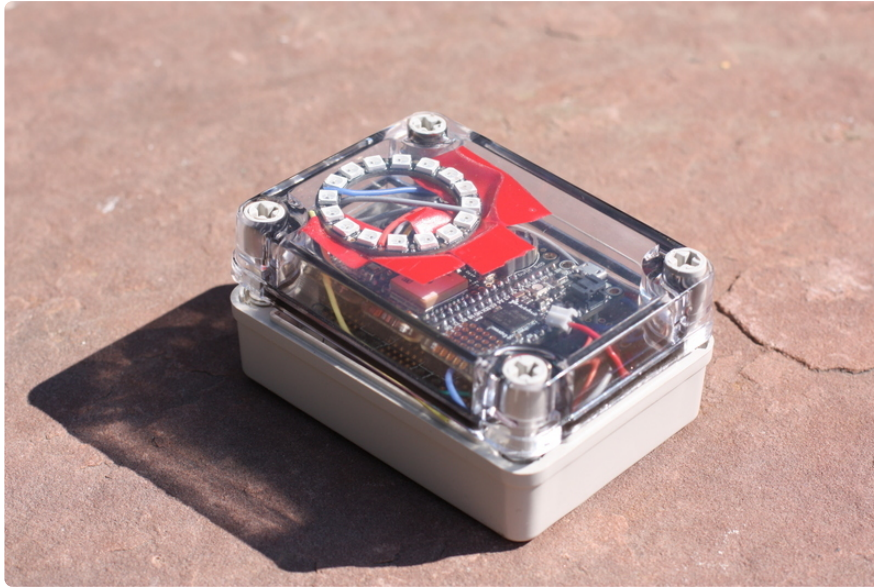
You'll also want a Phillips-head screwdriver on hand for opening and closing the enclosure.

Note that this part list contains the components to build one GlitterPOS box, but **two is the minimum** you'll need to do anything with! Of course, the more the merrier!



---

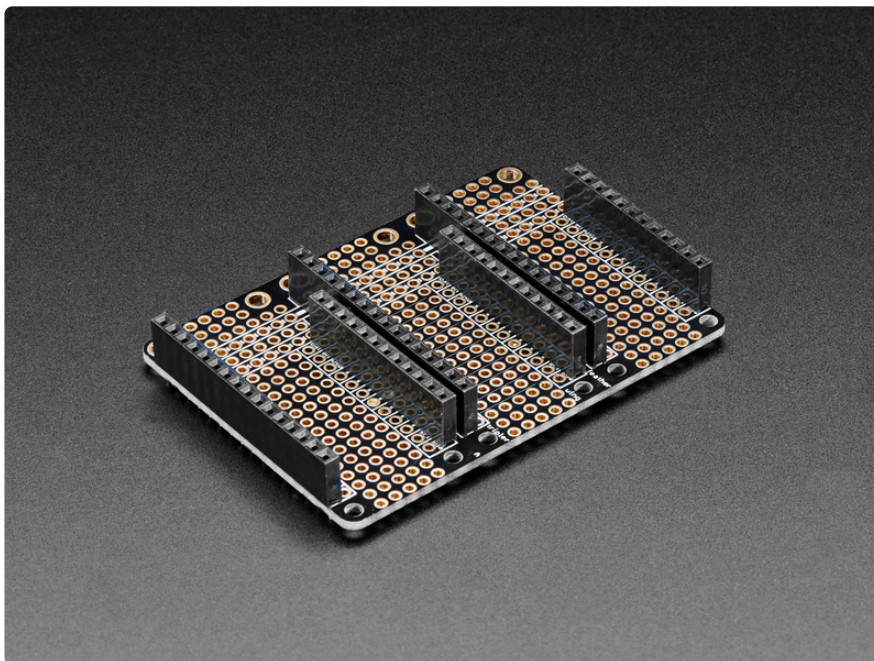
# Assembly



You'll need two (or more) boxes for the Glitter Positioning System to work. This section of the guide describes the build for a single box.

## FeatherWing Tripler

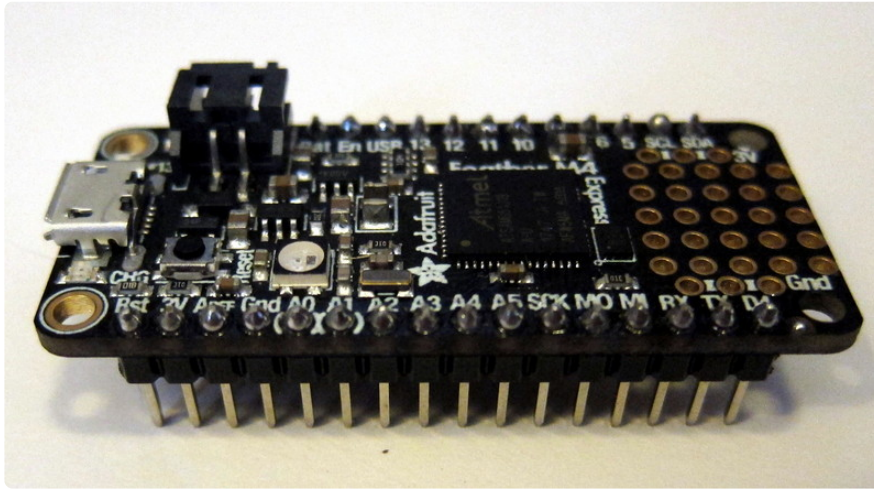
Since the FeatherWing Tripler forms the base of the entire project, it makes sense to start there. Solder female headers onto the top of the Tripler, as seen here:





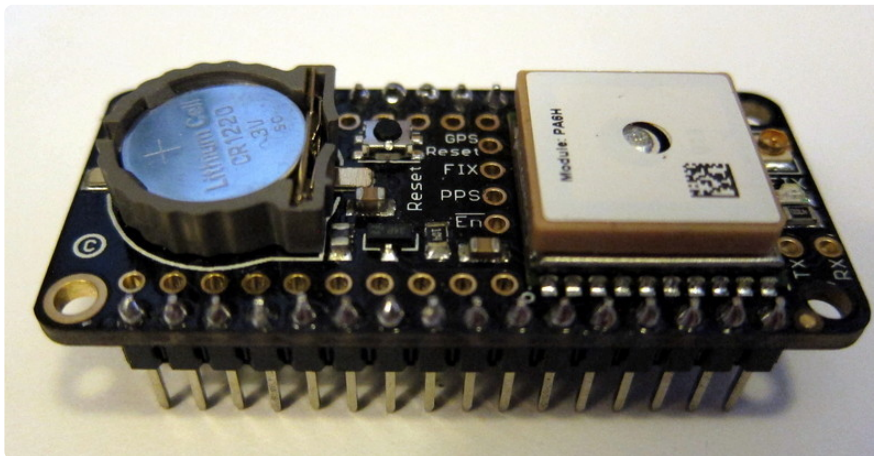
# Feather M4 Express

Solder plain male headers onto the Feather M4 Express:



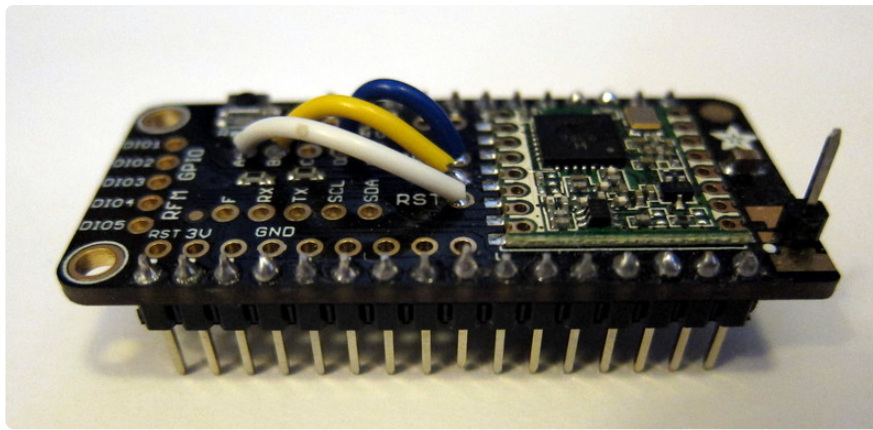
# Ultimate GPS FeatherWing

Solder plain male headers onto the Ultimate GPS FeatherWing, and install a CR1220 battery:

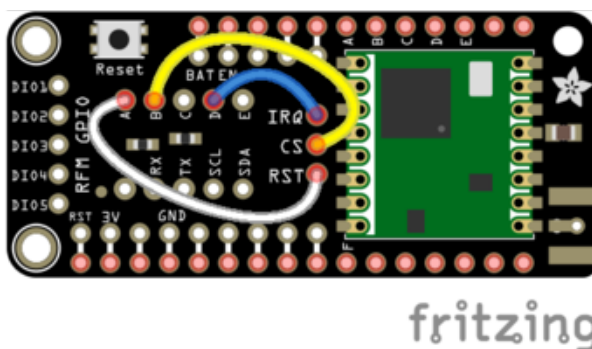


# LoRa Radio FeatherWing

Solder plain male headers onto the LoRa Radio Featherwing. Because each Feather uses a different processor, there's also some light wiring that needs to be done on the FeatherWing to configure the radio pins. Additionally, you'll need to attach an antenna.



You will want to solder three short wires for **CS**, **RST** and **IRQ**, as follows:



A to RST  
B to CS  
D to IRQ

When plugged into the FeatherWing Tripler, **A** maps to pin 11 on the Feather M4, **B** to pin 10, and **D** to pin 6.

Next, attach an antenna. There are [detailed instructions here \(https://adafru.it/CFZ\)](https://adafru.it/CFZ). For this project, a 3-inch length of wire works well - remember you'll need to adjust the length if you're using a module other than the 900 MHz RFM95W LoRa Radio FeatherWing. You can also use an external antenna, but you'll need to consider mounting it on your enclosure.

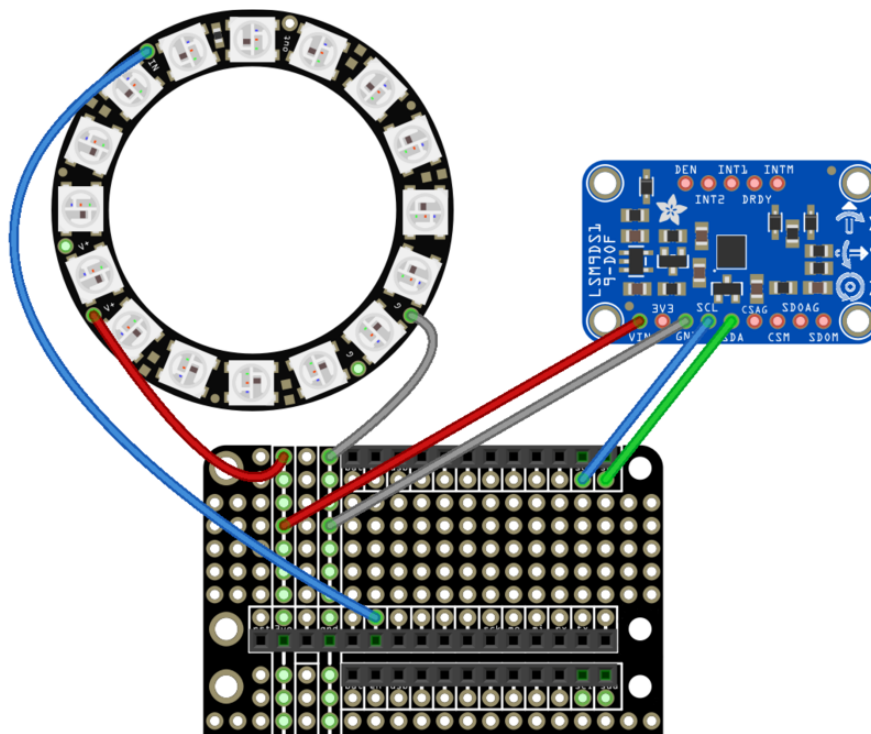
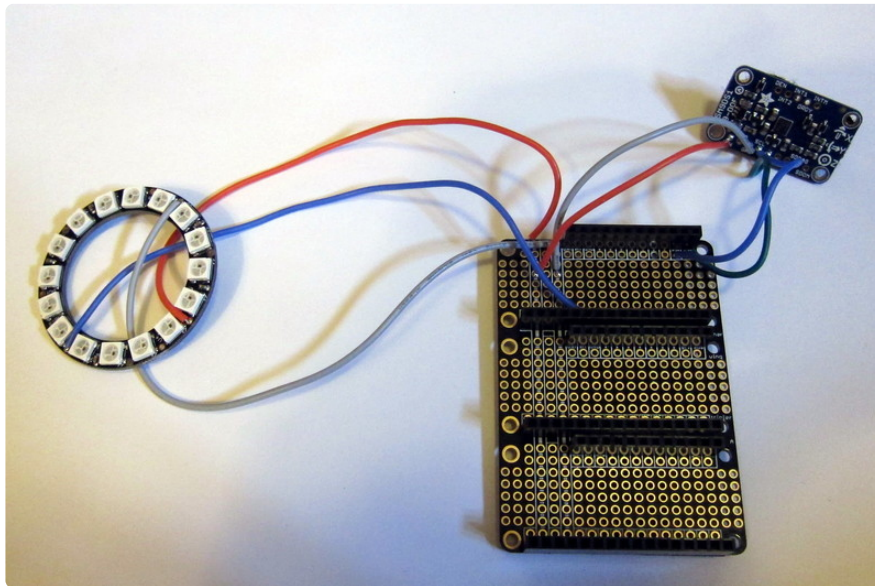
As you can see in the photos, for my build, I soldered a single pin in place and used it to connect a length of jumper wire. This worked well for testing, but you're probably better off just soldering the wire directly to the board.

## NeoPixel Ring & LSM9DS1 9-DOF

The NeoPixel ring and the LSM9DS1 breakout need to be connected to the FeatherWing Tripler underneath the Feather M4 Express. It's helpful to have the

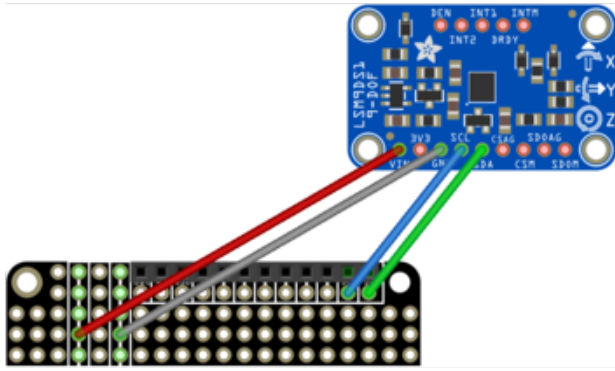
Feather ready but not yet seated on the Tripler, so you can double-check which pins you're soldering to.

When finished with this section, your build should look like this:

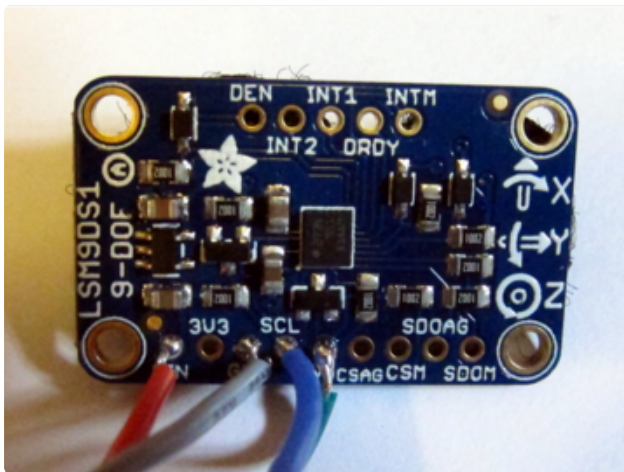


## LSM9DS1

Start by cutting four pieces of hookup wire for the LSM9DS1. You'll want enough length to tuck the breakout board into a corner of the enclosure, next to the battery. I used roughly 8cm pieces. Solder these as follows:



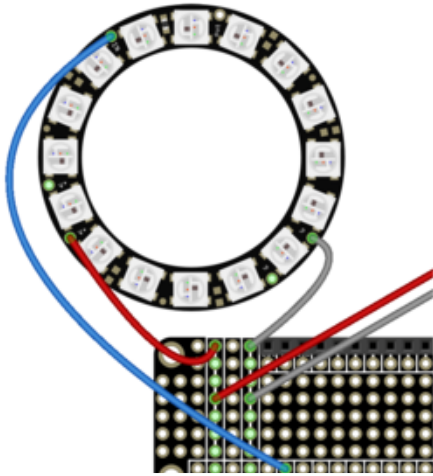
Tripler power rail to LSM9DS1 VIN  
 Tripler ground rail to LSM9DS1 GND  
 Tripler / Feather SCL to LSM9DS1 SCL  
 Tripler / Feather SDA to LSM9DS1 SDA



## NeoPixel Ring

Next, cut three pieces of hookup wire for the NeoPixels. You'll want enough length to position the NeoPixel within the enclosure without creating too much of a tangle. I used roughly 15cm pieces of the silicone-cover wire. Solder these as follows:

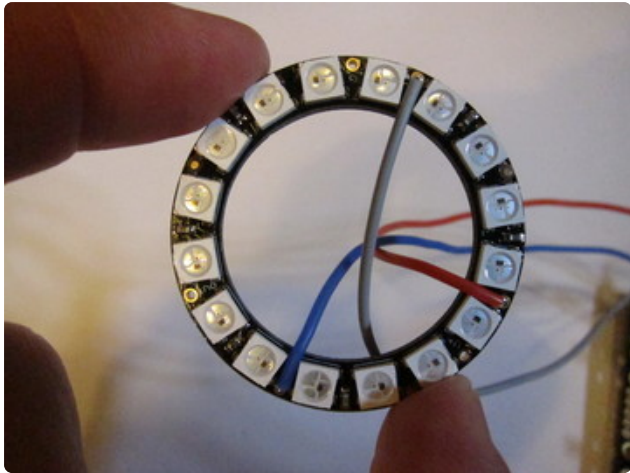




Tripler ground rail to NeoPixel Ground

Tripler power rail to NeoPixel Power 5V DC

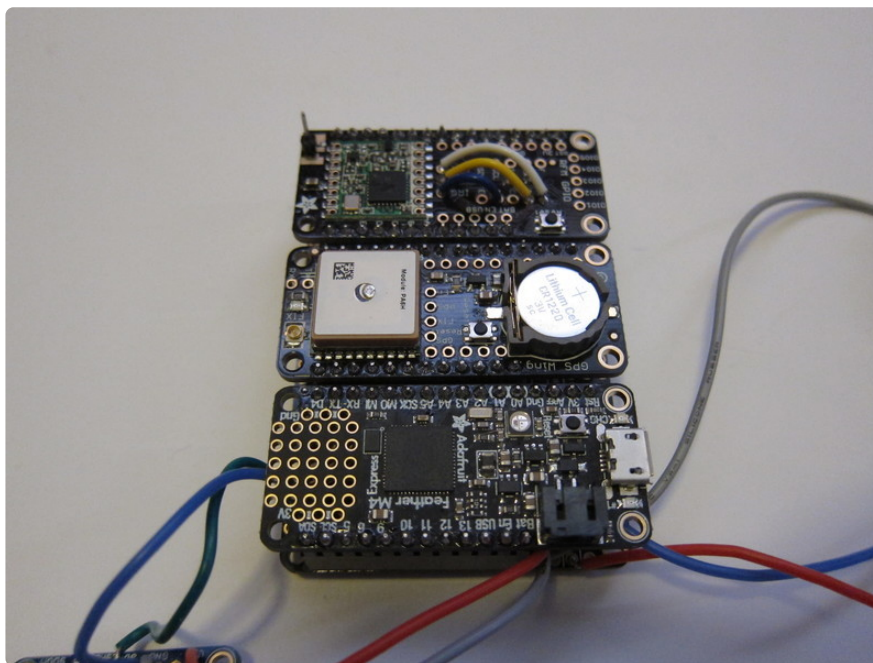
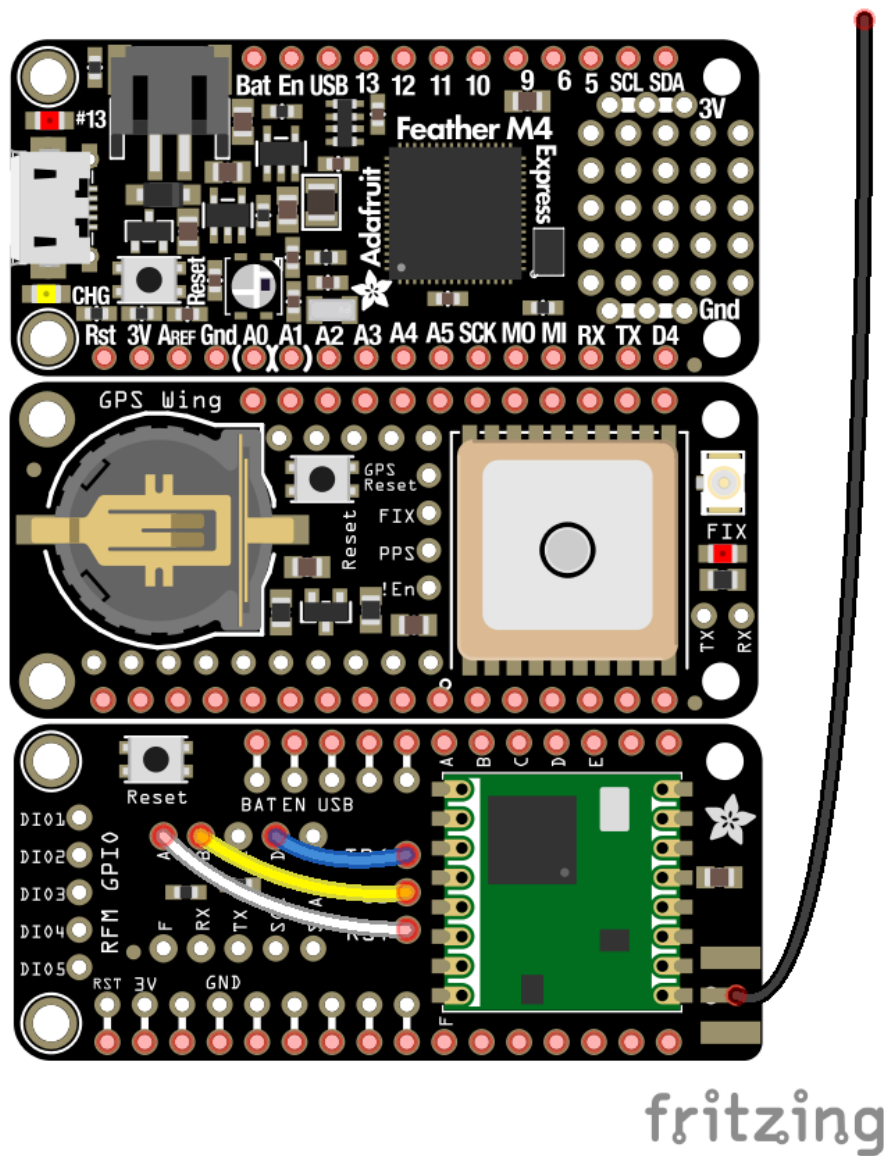
Tripler / Feather A1 to NeoPixel Data Input



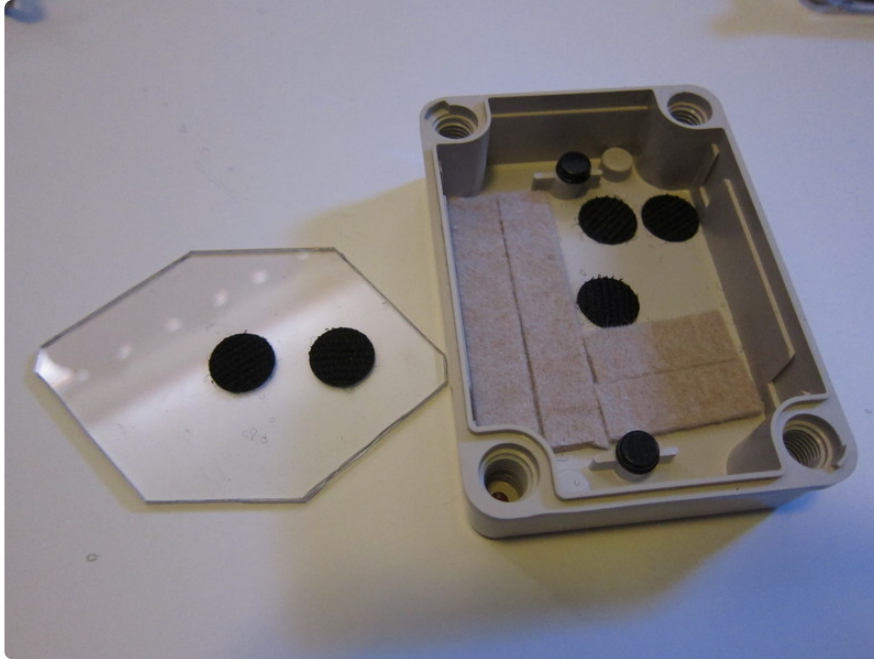
## Final Assembly & Enclosure

Plug the Feather M4 Express, Ultimate GPS FeatherWing, and Radio FeatherWing into the FeatherWing Tripler. The Feather M4 Express should be seated on the section of the FeatherWing Tripler where you connected the NeoPixel ring and LSM9DS1.



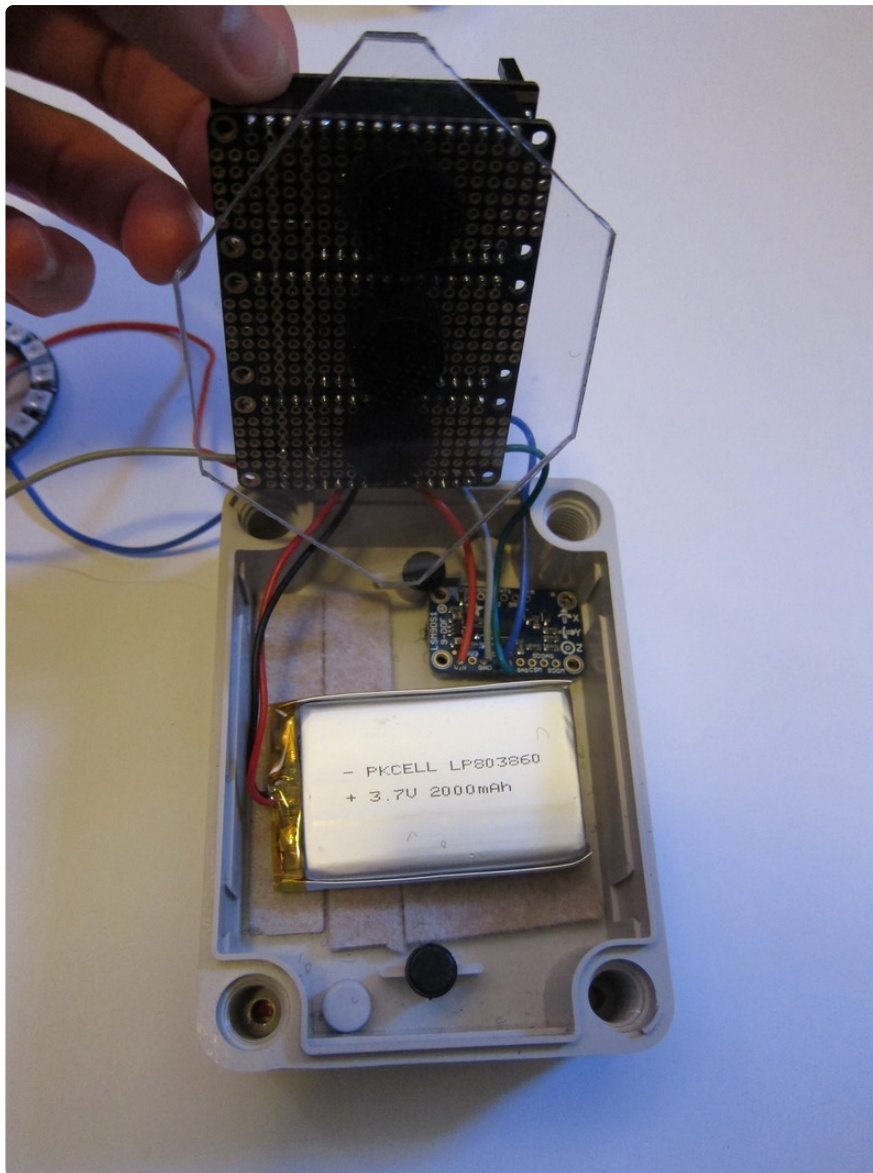


Next, prepare the enclosure. I used a small sheet of acrylic to shield the battery and the compass, self-adhesive Velcro fasteners to hold things in place, and some strips of felt to line the bottom of the enclosure. The felt turned out to be more trouble than it was worth, but the Velcro is a good way to keep things in place without permanently fastening them.



The acrylic can be replaced with a piece of cardboard or other material - just make sure the pins on the bottom of the FeatherWing Tripler won't be able to puncture the battery!

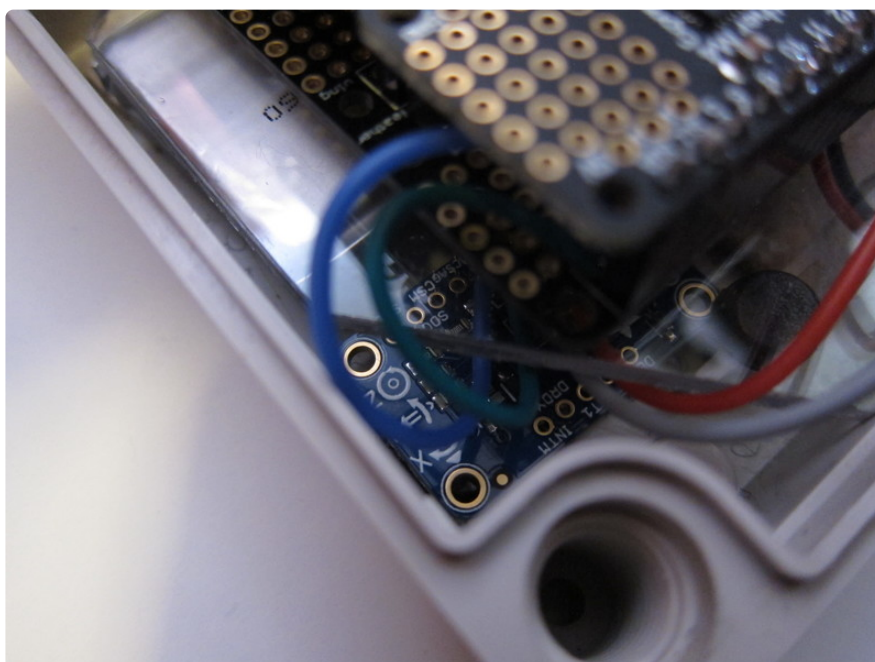
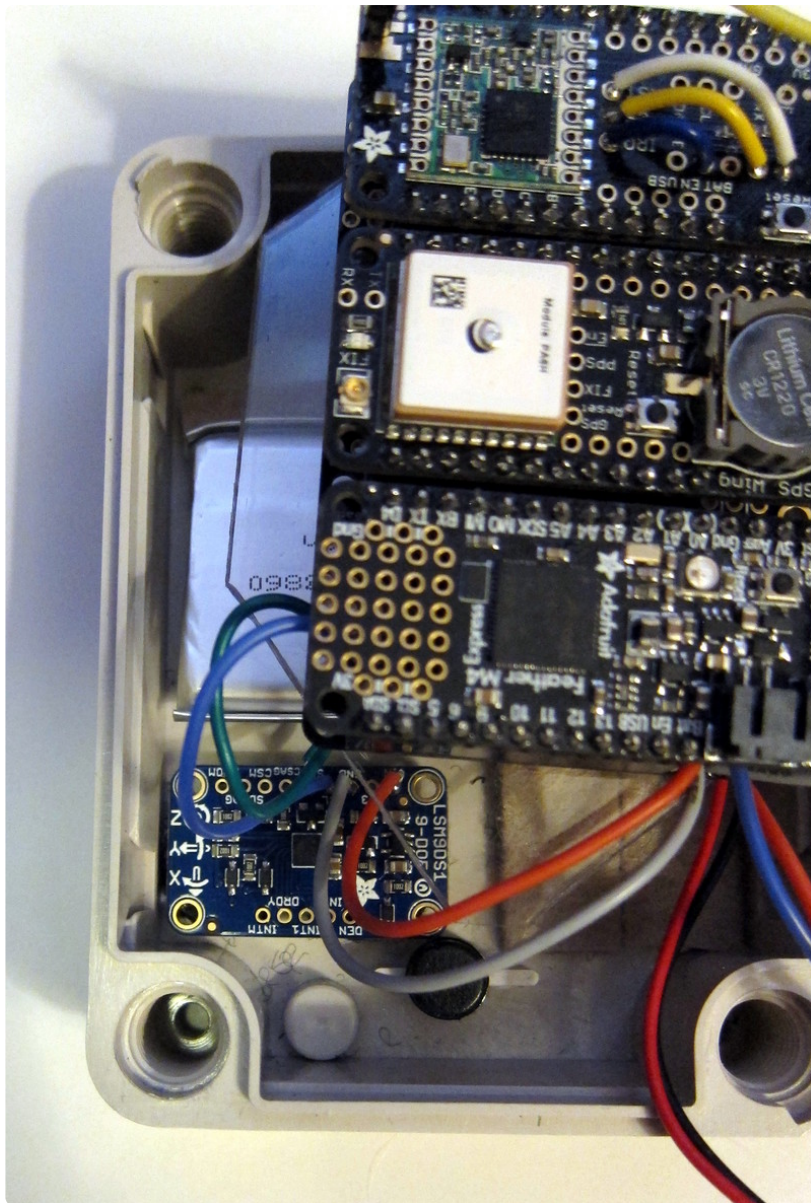
Install the compass and battery:



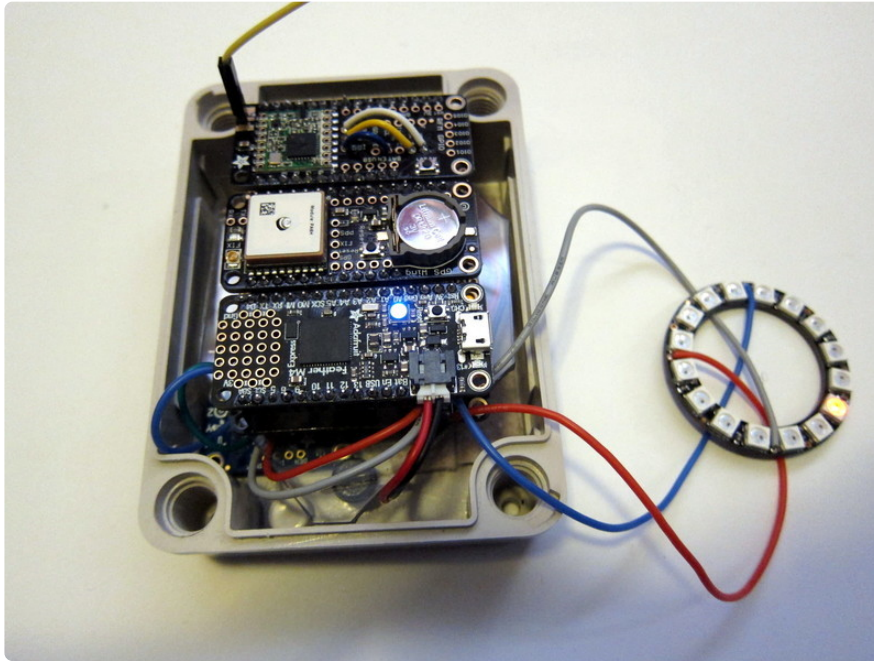
Orientation matters!

The LSM9DS1 should be installed in the corner of the enclosure with the X and Y axis labels pointed outwards, at the same end as the Feather M4 Express, like so:

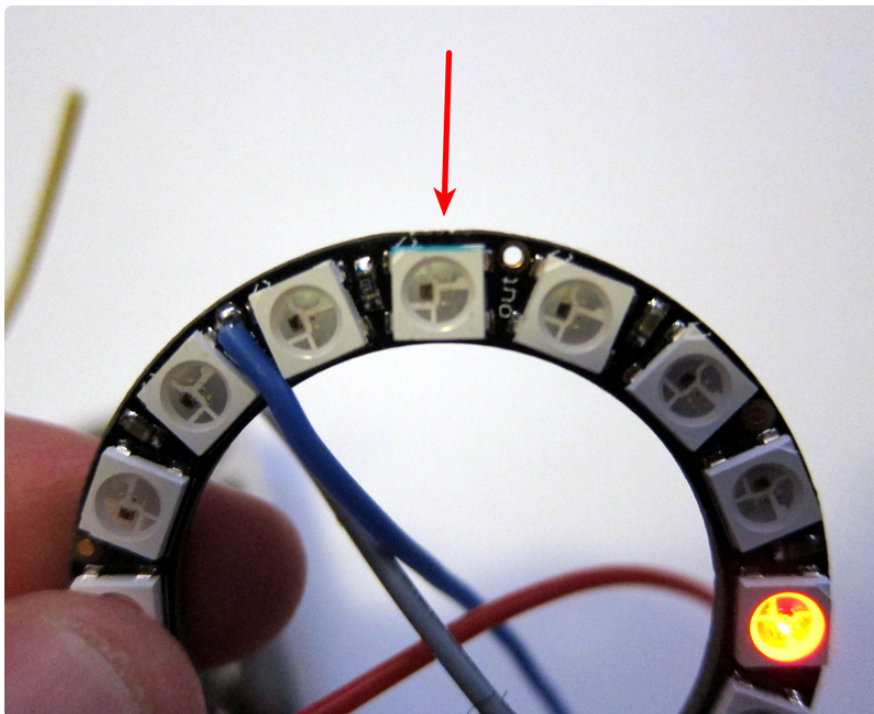




Now fit the rest of the electronics on top of the battery, LSM9DS1, and acrylic divider:

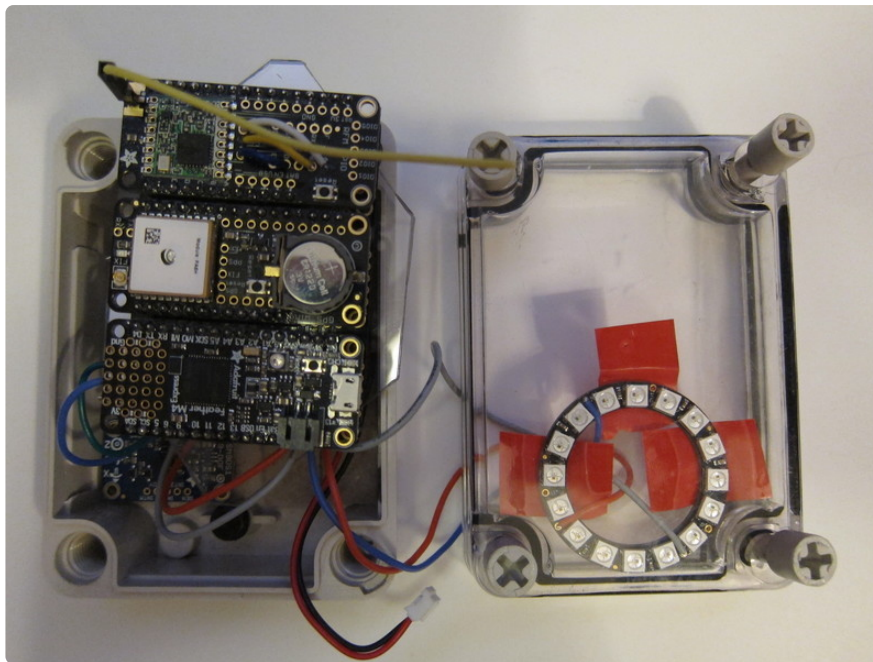


With everything else in place, mount the NeoPixel ring on the lid of the enclosure. I used a handful of pieces of electrical tape to make later adjustments easy. Again, orientation matters. Locate the first LED on the ring, just counter-clockwise from the **Data Out / out** pin:



It can be helpful to mark the edge of this LED with a permanent marker for quick identification. Once you've got it located, orient it directly opposite the solid arrow on the Y axis marker of the LSM9DS1:





Finally, connect the battery to the JST connector on the Feather M4 Express.

You should now be ready to move on to programming the unit and calibrating the compass.

## CircuitPython Code

```

Reading: 42.4279
Reading: 41.339
Reading: 41.1153
Reading: 40.8191
Reading: 41.0075
Reading: 40.799
Reading: 42.3219
Reading: 41.7641
Reading: 42.4685
Reading: 41.7648
Reading: 41.4194
Reading: 40.6131
Reading: 42.255
Reading: 40.6158
Reading: 42.2195
Reading: 41.341
Reading: 41.7327
.: 5536.52
2048/1/1 01:13:44
quality: 2
56736 bytes free
sending: e:1:0:40.2222:-105.266:0.11:41.7327
Reading: 41.9738
Reading: 40.5894
Reading: 41.8075
Reading: 41.8469
Reading: 41.3047
Reading: 40.2814
.: 5540.55
2048/1/1 01:13:48
quality: 2
17408 bytes free
sending: e:1:0:40.2222:-105.266:0.11:40.2814
Reading: 41.8496
Reading: 40.5619
Reading: 41.6347
Reading: 42.6052
Reading: 41.7471
Reading: 40.6951
Reading: 41.971
Reading: 40.8718
Reading: 42.012

File Recent Files Edit Tools Buffers Window Syntax DrChip Plugin Vimwiki Cheatsheet
Press ? for help
glitter_positioning system**
import time
import gc
import math
import adafruit_lsm9ds1
import adafruit_gps
import adafruit_rfm9x
import board
import busio
import digitalio
import neopixel
import rtc
from glitterpos.util import timestamp, compass_
bearing, bearing_to_pixel, map_range
glitterpos.py 6% 1/276 ln: 2
adafruit/cfg.py
adafruit/cfg.py 66% 8/12 ln: 1
NV_ID = 1
# Compass calibration values. From the CircuitPython REPL, use 'import
# calibrate' to find values for MAG_NIN and MAG_MAX:
brennen@externality 17:57:38 /home/brennen/code/adafruit/glitterpos (master 2/0)
brennen@externality 18:13:38 /home/brennen/code/adafruit/
glitterpos (master 2/0) -
(qvin:9486) Gtk-CRITICAL **: gtk_distribute_natural_allo
cation: assertion 'extra_space >= 0' failed
(qvin:9486) Gtk-CRITICAL **: gtk_distribute_natural_allo
brennen@externality 19:13:41 /home/brennen/code/adafruit/glitter
pos (master 2/0) -
(qvin:9486) Gtk-CRITICAL **: gtk_distribute_natural_allocation:
brennen@externality 19:13:42 /home/brennen/code/adafruit/glitterpos (master 2/0)
(qvin:9486) Gtk-CRITICAL **: gtk_distribute_natural_allocation: assertion 'extr
a_space >= 0' failed
brennen@externality 19:13:44 /home/brennen/code/adafruit/glitterpos (master 4/0)

```

If you're new to using CircuitPython, there's a [full getting started guide here \(https://adafru.it/cpy-welcome\)](https://adafru.it/cpy-welcome).

If you don't already have a preferred editor, Adafruit suggests using the Mu editor to edit your code and use the interactive REPL in CircuitPython. [You can learn about Mu and installation in this tutorial \(https://adafru.it/ANO\)](https://adafru.it/ANO).

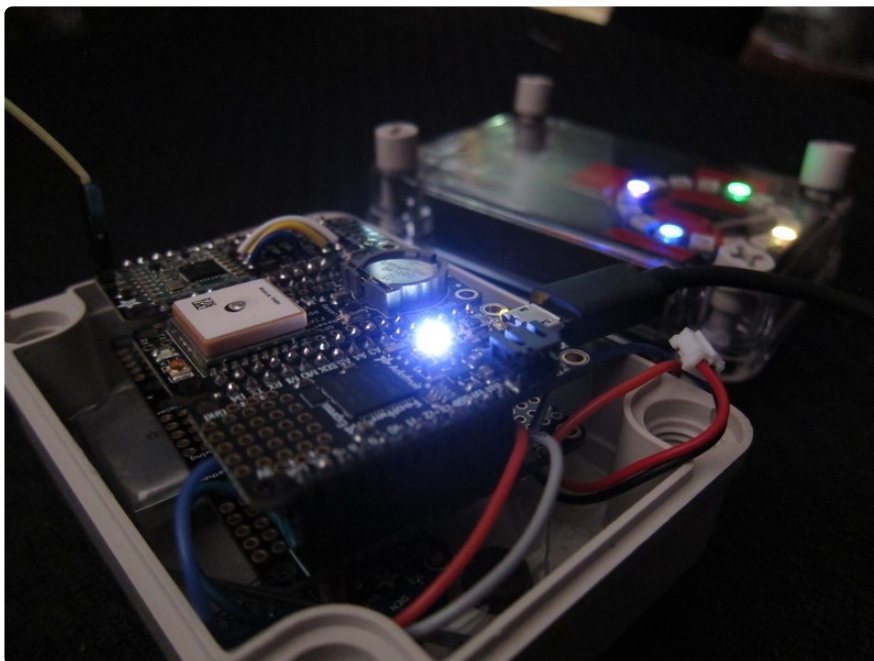
# Get the Code

Download the CircuitPython code from [the GitHub repository \(https://adafru.it/CF-\)](https://adafru.it/CF-). You can use [this download link \(https://adafru.it/CG0\)](https://adafru.it/CG0) to get a zip file, or, if you're familiar with Git, you can clone the repository with the following command:

```
git clone https://github.com/adafruit/glitterpos.git
```

Connect a USB cable to the Feather M4 Express, and use your computer to copy the contents of the **glitterpos** file folder to the **CIRCUITPY** drive that appears when you plug the Feather in. If you don't have a **CIRCUITPY** drive but do have a **FEATHERBOOT** drive, you will need to [first load CircuitPython onto the Feather \(https://adafru.it/CG1\)](https://adafru.it/CG1) then [load the libraries for talking to the sensors and radio \(https://adafru.it/CG1\)](https://adafru.it/CG1).

Once the code is copied and the Feather has had a chance to reboot, you should see a test pattern appear on the NeoPixel ring:



At this point, the GlitterPOS box will wait until it has a GPS fix and then begin broadcasting and listening for coordinates. There are a few more steps, however, before it's ready to use.

## Configuration

On each box, you'll need to edit a configuration file called `glitterpos_cfg.py`:

```

"""Configuration values for the Glitter Positioning System."""

# id should be a unique integer value for each box:
MY_ID = 0

# Compass calibration values. From the CircuitPython REPL, use `import
# calibrate` to find values for MAG_MIN and MAG_MAX:
MAG_MIN = [-0.1883, -0.16002, -0.53634]
MAG_MAX = [0.5887, 0.72618, 0.19474]

# Magnetic North - should be customized for your rough location:
DECLINATION_RAD = 0.1451 # Declination for Boulder, CO

```

**MY\_ID** should be a unique integer. For the first box, you should leave it set to **0**. On the second box, use **1**, on the third use **2**, etc. This number will be transmitted when the box sends coordinates over LoRa, and used by the other boxes to keep track of its position and choose a display color.

## Compass Calibration

**MAG\_MIN** and **MAG\_MAX** are compass calibration values. They specify the minimum and maximum magnetometer values from the LSM9DS1. In order to get a reliable compass heading from your device, you'll need to set these for each box.

This process can be a bit finicky, so we've written code to simplify the task. Begin by [connecting to the serial console \(https://adafru.it/Bec\)](https://adafru.it/Bec) on the Feather M4 Express.

You should now see a stream of debugging information, including GPS fix quality, headings, and data packets sent:

```

startup time: 2000/1/1 00:00:00
op mode: 0b10001000
  sending: e:d:0:hello world
heading: 162.521
heading: 165.844
heading: 161.111
.: 1937.99
2047/12/31 21:01:46
  quality: 1
  83584 bytes free
  sending: e:l:0:40.2222:-105.266:None:161.111
heading: 162.875
heading: 154.866
heading: 167.992
heading: 161.328
heading: 157.305
heading: 151.068
heading: 159.424

```

Press **Ctrl-C** to interrupt the running code, and hit any other key to enter the CircuitPython REPL:

```

heading: 157.179
heading: 152.52
heading: 151.521
heading: 144.878
heading: 158.518
heading: 155.398
heading: 156.005
heading: 158.139
heading: 163.32
heading: 160.683
heading: 151.958
Traceback (most recent call last):
  File "code.py", line 4, in <module>
    File "glitterpos.py", line 135, in advance_frame
    File "glitterpos.py", line 207, in radio_rx
    File "/lib/adafruit_rfm9x.py", line 590, in receive
    File "/lib/adafruit_rfm9x.py", line 443, in listen
    File "/lib/adafruit_rfm9x.py", line 308, in __set__
    File "/lib/adafruit_rfm9x.py", line 400, in _read_u8
KeyboardInterrupt:

Press any key to enter the REPL. Use CTRL-D to reload.

Adafruit CircuitPython 3.0.0 on 2018-07-09; Adafruit Feather M4 Express with samd51j19
>>> 

```

You'll be presented with a prompt. Now, type `import calibrate`:

```

Press any key to enter the REPL. Use CTRL-D to reload.

Adafruit CircuitPython 3.0.0 on 2018-07-09; Adafruit Feather M4 Express with samd51j19
>>> import calibrate
Magnetometer Calibration
Uncalibrated: 0.17038 0.31262 -0.40824
Calibrated: 0.083334 0.344254 0.0559454
MAG_MIN = [0.15946, 0.30114, -0.42938]
MAG_MAX = [0.17962, 0.31822, -0.38934]
Heading: 76.3921
----
Uncalibrated: 0.16534 0.31738 -0.39102
Calibrated: -0.259257 0.681148 0.141667
MAG_MIN = [0.15694, 0.30114, -0.42938]
MAG_MAX = [0.17962, 0.32046, -0.36218]
Heading: 110.838
----
Uncalibrated: 0.16758 0.3115 -0.4109
Calibrated: -0.126437 0.0724568 -0.450001
MAG_MIN = [0.15694, 0.30114, -0.42938]
MAG_MAX = [0.1813, 0.32046, -0.36218]
Heading: 150.184
----

```

This will load and run a function that records minimum and maximum values for the magnetometer. Now, move the box in a figure eight and rotate it around the x, y, and z axes multiple times until the `MAG_MIN` and `MAG_MAX` values don't appear to be changing. Copy those values into your `glitterpos_cfg.py`.

Mike Tuupola has a [good writeup on magnetometer calibration](https://adafru.it/CG2) (<https://adafru.it/CG2>) which goes into more detail.

## Magnetic North

Because magnetic north varies from true north by different amounts depending on where you're at on the surface of the earth, you'll need to find the correct [magnetic declination](https://adafru.it/CG3) (<https://adafru.it/CG3>) and set `DECLINATION_RAD` for your general location.

Start by finding your local declination. You can get this value from a few different places:

- [magnetic-declination.com](https://adafru.it/CG4) (<https://adafru.it/CG4>) - offers a clickable map.
- [The NCEI Magnetic Field Calculators](https://adafru.it/CG5) (<https://adafru.it/CG5>) - requires that you know either your coordinates, a local United States zip code, or a country and city.

Here's an example for Boulder, CO using the NCEI site:

The screenshot shows the NOAA NCEI Magnetic Field Calculators interface. The 'Declination' tab is active, displaying a map of Boulder, CO. A red compass rose is overlaid on the map, indicating the magnetic declination. The 'Calculate Declination' section on the left shows the input fields: Latitude: 40° 2' 23", Longitude: 105° 16' 46", Model: WMM (2014-2019), Date: Year 2018, and Result format: HTML. The 'Calculate' button is visible. The 'Declination' pop-up window shows the results: Model Used: WMM2015, Latitude: 40° 2' 23" N, Longitude: 105° 16' 46" W, Date: 2018-09-27, and Declination: 8° 19' E ± 0° 21' changing by 0° 6' W per year.

And here's one using magnetic-north.com:

The screenshot shows the magnetic-north.com interface. The 'Find the magnetic declination at your location' section is active. A map of Boulder, CO is displayed, with a red pin indicating the location. A pop-up window shows the results: You clicked here: Latitude: 40° 1' 30.6" N, Longitude: 105° 16' 50.2" W, **BOULDER**, Magnetic Declination: +8° 19', Declination is **POSITIVE (EAST)**, Inclination: 66° 25', and Magnetic field strength: 52036.9 nT. The left sidebar shows the search process: 'Find your location' with 'boulder' entered, 'USA' selected, and 'COLORADO' selected. The 'Browse countries' section shows 'BOULDER CO' and 'BOULDER JUNCTION CO' as options.



Both give positive 8 degrees, 19 minutes. In order to use this, you'll need to convert it to radians. In decimal degrees, that's  $8 \times 60 + 19 / 60 = 8.3166^\circ$ . To find radians, multiply by  $\pi/180$ :

$$8.3166 * (\pi/180) = 0.1451$$

So you can set `DECLINATION_RAD = 0.1451` and be reasonably comfortable with the result.

Wolfram Alpha also provides a quick answer here:

The screenshot shows the Wolfram Alpha interface. The input bar contains "8° 19' in radians". Below the input bar, there are options for units and a "more" dropdown. The "Input interpretation:" section shows "convert 8 degrees 19 arc minutes to radians". The "Result:" section shows the fraction  $\frac{499\pi}{10800}$  radians. The "Additional conversions:" section shows "145.2 mrad (milliradians)" and "8.317° (degrees)".

With those values set on each GlitterPOS box, you should be ready to strike out and track your friends' whereabouts very imprecisely.

## Usage, Customization, and Next Steps

The Glitter Positioning System isn't intended for use as serious wayfinding equipment! Have fun, but please don't rely on this project in any situation where life and limb are at risk, or where you might become seriously lost.

It's good to keep some basic constraints in mind before setting out:

- The code doesn't presently handle tilt compensation, so you'll need to keep the box fairly flat to get good compass readings.
- The code doesn't currently special-case the situation where a NeoPixel needs to point at more than one box, so at present a box can be hidden "behind" another box if it's on the same compass bearing.

- Within 2-5 meters, GPS readings are unlikely to be accurate enough to provide a useful bearing to the other units.
- The GPS will work best with a clear view of the sky, and under adverse conditions it may take many minutes to get a fix.
- Radio range will vary depending on terrain and atmospheric conditions. You'll do best in a relatively flat landscape with few buildings or other obstructions, but with a simple wire antenna you should be able to transmit upwards of half a kilometer in many environments.

It is best to test everything at home prior to any event you plan to attend with your boxes, just to make sure everything appears to work as you believe it should.

## Display Customization

LED colors for the first eight box IDs are RGB tuples in a dictionary constant called `COLOR_LOOKUP`, defined in `glitterpos.py`:

```
# Color presets for each glitterpos_id:
COLOR_LOOKUP = {
    0: GREEN,
    1: BLUE,
    2: PURPLE,
    3: YELLOW,
    4: CYAN,
    5: (100, 0, 255),
    6: (0, 100, 200),
    7: (100, 50, 100),
    # BOULDER_ID: RED
}
```

You can extend or modify this dictionary to your liking. (Note that the file also contains some handy pre-defined color constants.)

Lots of interesting variations on the display code are possible. Another easy tweak is to display headings for fixed points of interest, such as your camp or a high-visibility landmark. As an example, `glitterpos.py` contains commented-out coordinates for downtown Boulder, CO. Open that file and uncomment all the lines containing the string `BOULDER_ID`, starting around line 35.

Practically speaking, you can add as many fixed points as you want, but remember that you only have 16 NeoPixels to work with, so the display can get cluttered fast.

# Potential Hardware Improvements

Both the Ultimate GPS FeatherWing and the Radio FeatherWing can be equipped with an external antenna for greatly improved performance.

See the Radio FeatherWing guide for details on [mounting a uFL or SMA antenna \(https://adafru.it/CFZ\)](https://adafru.it/CFZ).

See the [Ultimate GPS FeatherWing guide \(https://adafru.it/CG6\)](https://adafru.it/CG6) for details on mounting an [external active GPS antenna \(http://adafru.it/960\)](http://adafru.it/960). In the case of the Adafruit GPS antenna, keep in mind that the antenna base contains a strong magnet, so you'll need to mount it at some distance from the body of the GlitterPOS box or it'll interfere with the compass.

The current build lacks an on/off switch, which would make for a considerable improvement in usability!

## Acknowledgments & Further Reading

While the end goal is somewhat different, [this Burning Man LoRa + GPS tracker project \(https://adafru.it/CK8\)](https://adafru.it/CK8) by Melissa Loudon and Drew Griscom Roos was a source of inspiration for the Glitter Positioning System, and good evidence that the radios would work well in that environment.

The compass-bearing code in `glitterpos_util.py` is from [this Gist by Jérôme Renard \(https://adafru.it/CG7\)](https://adafru.it/CG7).

Ozzmaker's [Create a Digital Compass with the Raspberry Pi \(https://adafru.it/CG8\)](https://adafru.it/CG8) series is an excellent resource for many of the techniques used here.

Thanks to Ladyada for working out the finicky details of compass calibration!