



Raspberry Pi Pokemon Finder

Created by Ruiz Brothers



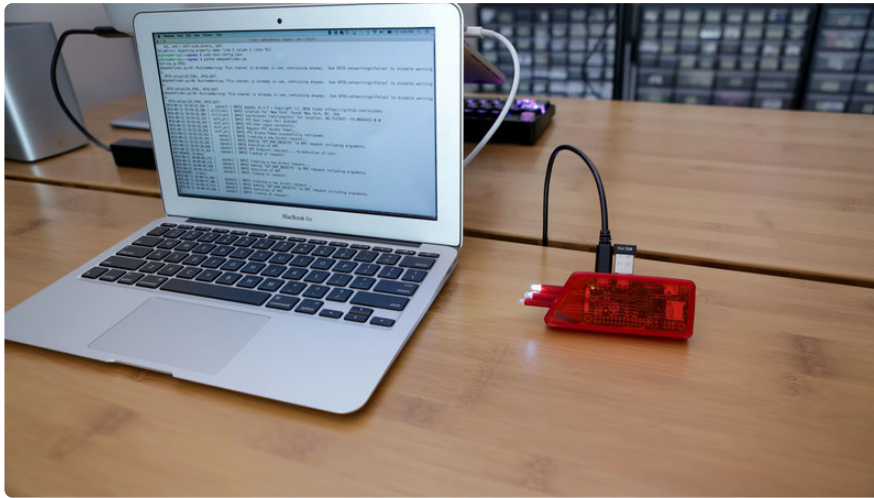
<https://learn.adafruit.com/pi-zero-pokemon-lure-module>

Last updated on 2024-06-03 01:58:21 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Reverse Lure Module!• How Does It Work?• Warning, Warning, Warning!!• Parts• Tools and Supplies	
Software	5
<ul style="list-style-type: none">• Prepare Pi Zero• Install packages• Install Pokemon Go API• Time to configure!• Test it out	
Circuit Diagram	9
<ul style="list-style-type: none">• Circuit Diagram• Wired Connections• 5V Power	
3D Printing	10
<ul style="list-style-type: none">• Slice Settings• Tolerances• CNC Milling Acrylic	
Wiring LEDs	12
<ul style="list-style-type: none">• Prepping LEDs• Ground Wires• Tinning Wires• LED Ground Wire• Repeat LED Ground Wires• Prep Signal Wires• Wire Signal• Wired LEDs• Installing LEDs• Installed LEDs• Connecting Ground Wires• Wiring Resistor• Connecting Resistor• Heatshrink Connections• Assembled LEDs• Tinning GPIO• Connect LEDs to GPIO	
Assemble	22
<ul style="list-style-type: none">• Install Antenna into Case• Install Pi• Installing Cover• Installed Cover• Raspberry Pi Lure Module Case	

Overview



Reverse Lure Module!

In this project we'll show you how to find Pokemon with the Raspberry Pi. It looks like a lure module, but instead of attracting Pokemon, it lets you know when they're close.

It like getting a notification when Pokemon are nearby. This can be really useful, especially when you're at work or stuck in class. So now you don't have to keep your mobile device open or keep checking every ten minutes.



How Does It Work?

When a common Pokemon appears, this first LED turns on. The second LED lights up when rare Pokemon are nearby, and when legendary Pokemon show up, the third LED will turn on.

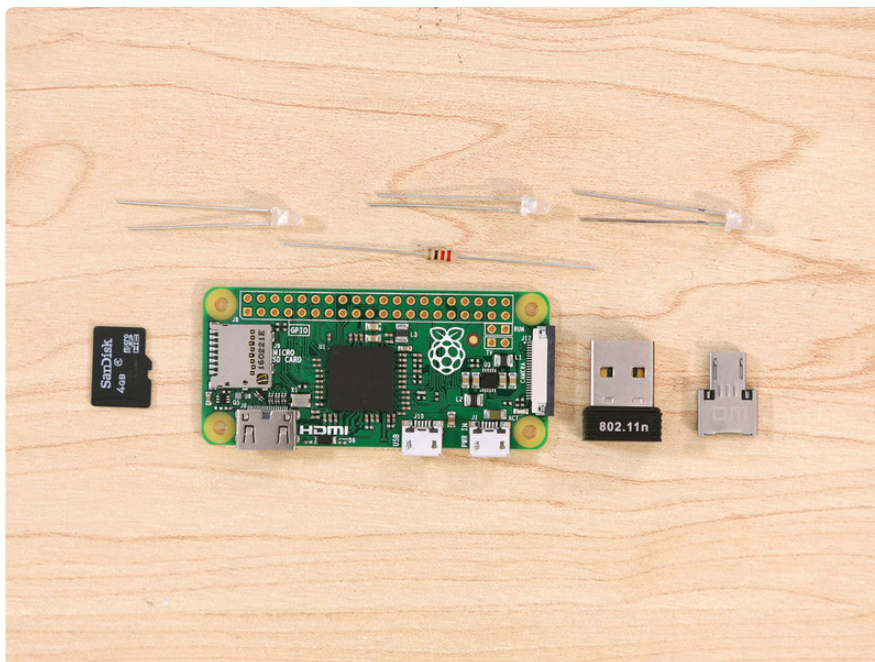
By loading some python code onto the Raspberry Pi, you can connect to the unofficial Niantic Pokemon API and get request when a Pokemon spawns in your area. In the JSON configuration file, you can specify any location.

Warning, Warning, Warning!!

This is really risky hacker business and could potentially get you banned! This is completely against Niantic's terms of service.

It's a proof of concept for educational hacking only! Niantic doesn't like it when folks use the API and they'll probably shut down access to it - So if you're watching this, it may or may not work.

So fair warning, Don't use this on your main account! And there's no guarantee it'll work at all - but the case is pretty cool!



Parts

- [Raspberry Pi Zero \(http://adafru.it/2885\)](http://adafru.it/2885)
- [Tiny OTG Adapter USB Micro to USB \(http://adafru.it/2910\)](http://adafru.it/2910)
- [Mini WiFi Module \(http://adafru.it/814\)](http://adafru.it/814)
- [3mm LED \(http://adafru.it/778\)](http://adafru.it/778)
- [220ohm Resistor \(http://adafru.it/2780\)](http://adafru.it/2780)
- [Perma-Proto Flex-PCB \(http://adafru.it/1518\)](http://adafru.it/1518)

Tools and Supplies

- [Soldering Iron](https://adafru.it/Cgp) (https://adafru.it/Cgp) + [Solder](https://adafru.it/Cgq) (https://adafru.it/Cgq)
- [30AWG Silicone-coated wire](http://adafru.it/2051) (http://adafru.it/2051)
- [Wire Strippers](http://adafru.it/527) (http://adafru.it/527)
- [3D Printer](https://adafru.it/diH) (https://adafru.it/diH) + [Filament](http://adafru.it/2080) (http://adafru.it/2080)



Software

Not only can you make this lovely Lure-like case, you can make the Pi Zero act as a reverse-Lure, a Pokemon Finder! By loading some python software into the Pi, you can connect to the unofficial Niantic Pokemon API and determine when a Pokemon spawns in your area.

Thanks to James for the writeup!

WARNING! WARNING! WARNING!

Connecting to Niantic's servers and using the API is completely against the terms of service. It's a fun hack but is not authorized by Niantic and can cause the account used to be soft-banned or perhaps even perma-banned!

Do not use your main account!

Also, Niantic doesn't particularly like it when people use the API and they are very likely to shut access to it (again) soon!

This code may not work past when this was written on August 11, 2016. There is no guarantee or support or expectation that this will work at all!

WARNING! WARNING! WARNING!

This is not a beginner Raspberry Pi tutorial, we assume familiarity with Linux! Check out our other Pi tutorials for getting started

Code on Github

<https://adafru.it/pPE>

Prepare Pi Zero

[Start by loading up Jessie Lite on your Raspberry Pi \(https://adafru.it/pHe\)](https://adafru.it/pHe)

You'll also need to log into a shell on your Pi Zero, [we prefer using a USB console cable \(https://adafru.it/pHf\)](https://adafru.it/pHf) but you can also connect a keyboard and use HDMI or connect an ethernet adapter and ssh to `raspberrypi.local`

Either way, get your Pi Zero up and running and you are logged in. **Expand Filesystem** from within `sudo raspi-config`. Reboot with the WiFi adapter connected to the USB OTG port

[Then set up the WiFi adapter, we prefer to do it this way \(https://adafru.it/dDN\)](https://adafru.it/dDN) but there's other techniques, just do whatever you like to get the Pi Zero on the network.

Reboot and make sure you can `ping raspberrypi.org` successfully

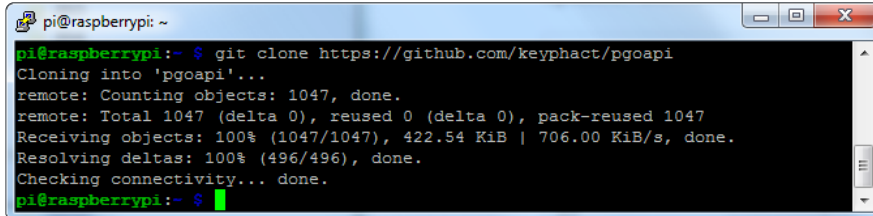
Install packages

Once networking is working, run the following at the shell to install pkgs:

- `sudo apt-get update`
- `sudo apt-get install git python python-pip python-dev`
- `sudo pip install --upgrade pip`

Install Pokemon Go API

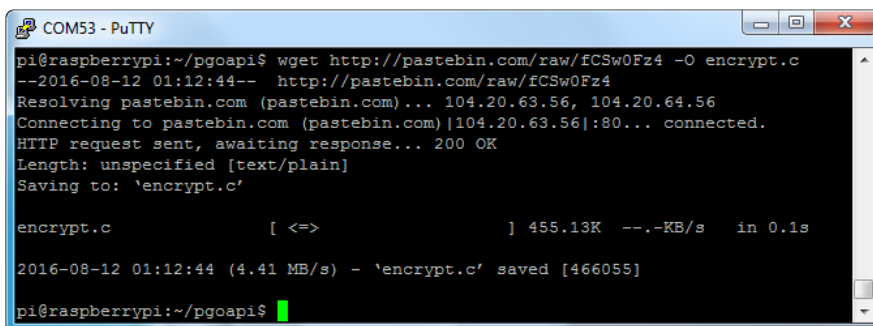
From the shell, get the latest PGoAPI with `git clone https://github.com/keyphact/pgoapi`



```
pi@raspberrypi:~$ git clone https://github.com/keyphact/pgoapi
Cloning into 'pgoapi'...
remote: Counting objects: 1047, done.
remote: Total 1047 (delta 0), reused 0 (delta 0), pack-reused 1047
Receiving objects: 100% (1047/1047), 422.54 KiB | 706.00 KiB/s, done.
Resolving deltas: 100% (496/496), done.
Checking connectivity... done.
pi@raspberrypi:~$
```

cd in the new directory (`cd pgoapi`) and then find and download the latest encryption file. As of this writing it was available at <http://pastebin.com/raw/fCSw0Fz4>

Run `wget http://encryptionfilelocationhere (https://adafru.it/pHA) -O encrypt.c` to save it as `encrypt.c`



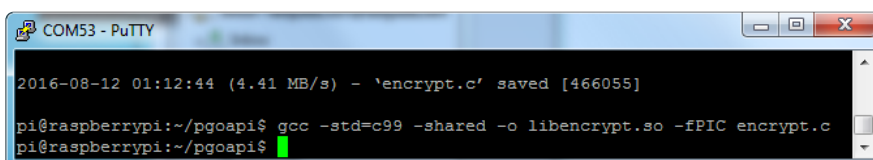
```
COM53 - PuTTY
pi@raspberrypi:~/pgoapi$ wget http://pastebin.com/raw/fCSw0Fz4 -O encrypt.c
--2016-08-12 01:12:44-- http://pastebin.com/raw/fCSw0Fz4
Resolving pastebin.com (pastebin.com)... 104.20.63.56, 104.20.64.56
Connecting to pastebin.com (pastebin.com)|104.20.63.56|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: 'encrypt.c'

encrypt.c      [ <=>          ] 455.13K  --.-KB/s  in 0.1s

2016-08-12 01:12:44 (4.41 MB/s) - 'encrypt.c' saved [466055]

pi@raspberrypi:~/pgoapi$
```

Compile it into a linked library (libencrypt.so) with `gcc -std=c99 -shared -o libencrypt.so -fPIC encrypt.c`



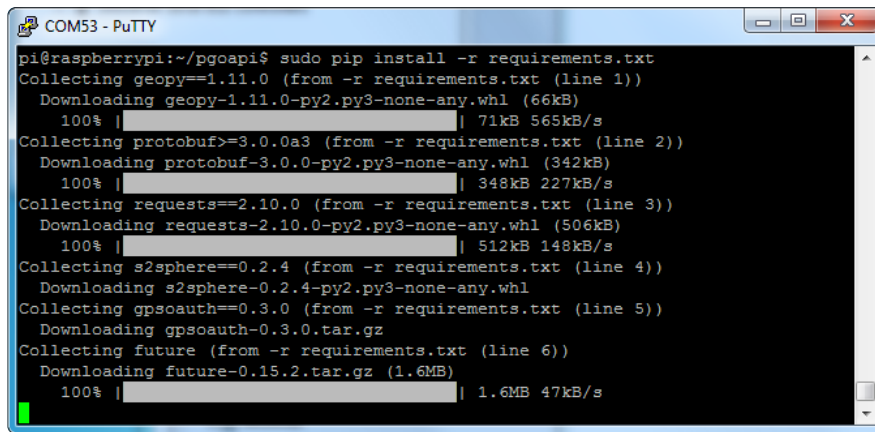
```
COM53 - PuTTY
2016-08-12 01:12:44 (4.41 MB/s) - 'encrypt.c' saved [466055]

pi@raspberrypi:~/pgoapi$ gcc -std=c99 -shared -o libencrypt.so -fPIC encrypt.c
pi@raspberrypi:~/pgoapi$
```

You'll need to install a bunch of software, luckily it's in a file so you can just run

```
sudo pip install -r requirements.txt
```

To do them all



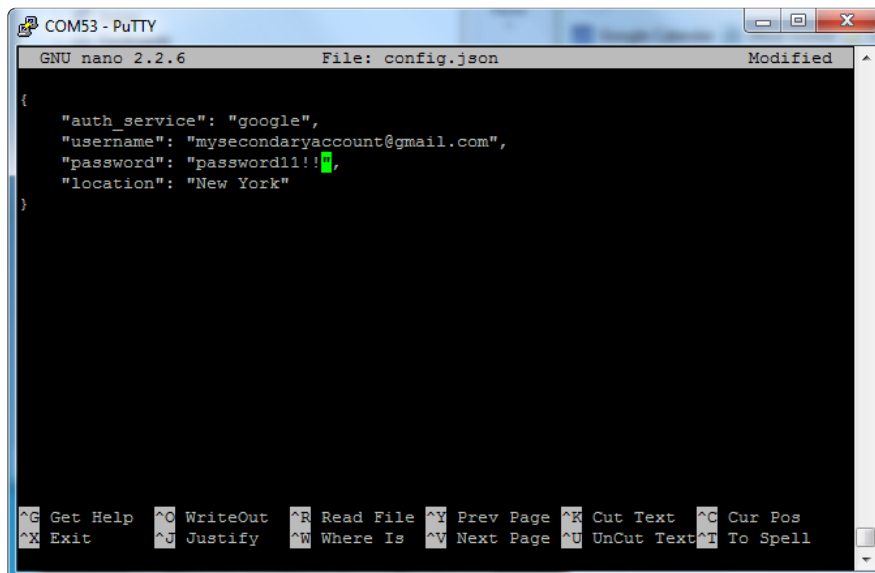
```
COM53 - PuTTY
pi@raspberrypi:~/pgoapi$ sudo pip install -r requirements.txt
Collecting geopy==1.11.0 (from -r requirements.txt (line 1))
  Downloading geopy-1.11.0-py2.py3-none-any.whl (66kB)
    100% |#####| 71kB 565kB/s
Collecting protobuf>=3.0.0a3 (from -r requirements.txt (line 2))
  Downloading protobuf-3.0.0-py2.py3-none-any.whl (342kB)
    100% |#####| 348kB 227kB/s
Collecting requests==2.10.0 (from -r requirements.txt (line 3))
  Downloading requests-2.10.0-py2.py3-none-any.whl (506kB)
    100% |#####| 512kB 148kB/s
Collecting s2sphere==0.2.4 (from -r requirements.txt (line 4))
  Downloading s2sphere-0.2.4-py2.py3-none-any.whl
Collecting gpsoauth==0.3.0 (from -r requirements.txt (line 5))
  Downloading gpsoauth-0.3.0.tar.gz
Collecting future (from -r requirements.txt (line 6))
  Downloading future-0.15.2.tar.gz (1.6MB)
    100% |#####| 1.6MB 47kB/s
```

Grab the `adapokefinder.py` file with `wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Pokemon-Detector/master/adapokefinder.py -O adapokefinder.py`

Time to configure!

Create your config with `mv config.json.example config.json` and then edit it with `nano config.json` - put in your account and the location you want to monitor

Remember, do not use your main account!



```
COM53 - PuTTY
GNU nano 2.2.6      File: config.json      Modified
{
  "auth_service": "google",
  "username": "mysecondaryaccount@gmail.com",
  "password": "password111!",
  "location": "New York"
}

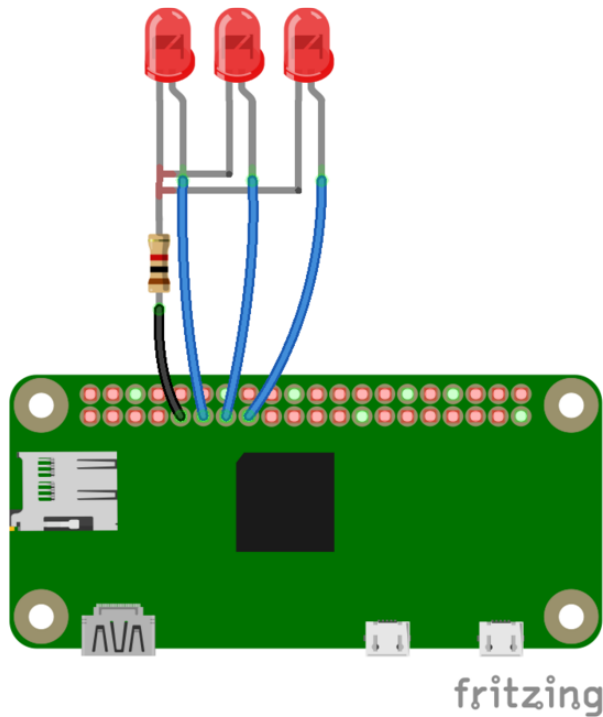
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^X Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Test it out

That's it! time to test. Run `python adapokefinder.py` and make sure you authenticate and see pokemon 'appear'

```
COM53 - PuTTY
pi@raspberrypi:~/pgoapi$ python adapokefinder.py
Setting up GPIO
2016-08-12 01:39:57,023 [ pgoapi] [ INFO] pgoapi v1.1.7 - Copyright (c) 2016
tjado <https://github.com/tejado>
2016-08-12 01:39:58,109 [ utilities] [ INFO] Location for '150 Varick st, New Yo
rk' found: 150 Varick St, New York, NY 10013, USA
2016-08-12 01:39:58,111 [ utilities] [ INFO] Coordinates (lat/long/alt) for loca
tion: 40.7260191 -74.00536 0.0
2016-08-12 01:39:58,113 [auth_google] [ INFO] Google User Login for: adafruitnex
us7@gmail.com
2016-08-12 01:39:58,955 [auth_google] [ INFO] Google User Login successful.
2016-08-12 01:39:58,957 [auth_google] [ INFO] Request Google Access Token...
2016-08-12 01:39:59,415 [auth_google] [ INFO] Google Access Token successfully r
eceived.
2016-08-12 01:39:59,784 [ pgoapi] [ INFO] Creating a new direct request...
2016-08-12 01:39:59,785 [ pgoapi] [ INFO] Adding 'GET_MAP_OBJECTS' to RPC req
uest including arguments
2016-08-12 01:39:59,794 [ pgoapi] [ INFO] Execution of RPC
2016-08-12 01:40:01,344 [ pgoapi] [ INFO] API Endpoint redirect... re-executi
on of call
2016-08-12 01:40:04,802 [ pgoapi] [ INFO] Cleanup of request!
#13 NEARBY, Weedle!
#21 NEARBY, Spearow!
RARE POKEMON FOUND!
#116 NEARBY, Horsea!
RARE POKEMON FOUND!
```

Circuit Diagram



Circuit Diagram

Use the circuit diagram to reference for connecting the components together. The diagram does not depict exact wire lengths or size of components.

Wired Connections

- LED 1 Cathode to Resistor
- LED 2 Cathode to Resistor
- LED 3 Cathode to Resistor
- LED 1 Anode to GPIO 17
- LED 2 Anode to GPIO 22
- LED 3 Anode to GPIO 27

5V Power

To power the Raspberry Pi, you'll need a [5V power supply \(http://adafru.it/1994\)](http://adafru.it/1994) / wall adapter. The Pi needs 5V power via a microUSB cable. This project is not meant to be portable.

3D Printing



Im-antenna.stl	Antenna for LEDs	
Im-case.stl	Case for the Raspberry Pi Zero	
Im-top.stl	Cover snap fits on top of case	

Download STLs

<https://adafru.it/pHC>

Download Source

<https://adafru.it/19Yc>

Slice Settings

Below are some recommended slice settings. We used simplify3D to slice these parts and generate the gcode. These parts were printed on a Sigma BCN3D using a .4mm nozzle and 1.75mm filament.

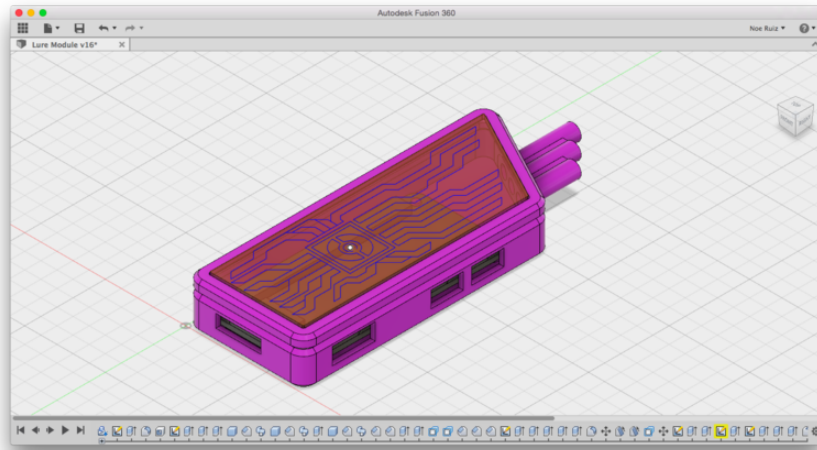
- .2mm layer height
- 220C extruder / 65c bed
- 20% infill
- 4 top/bottom layers
- 2 shells / parameters
- 60mm/s printing speed with 120mm/s travel

Tolerances

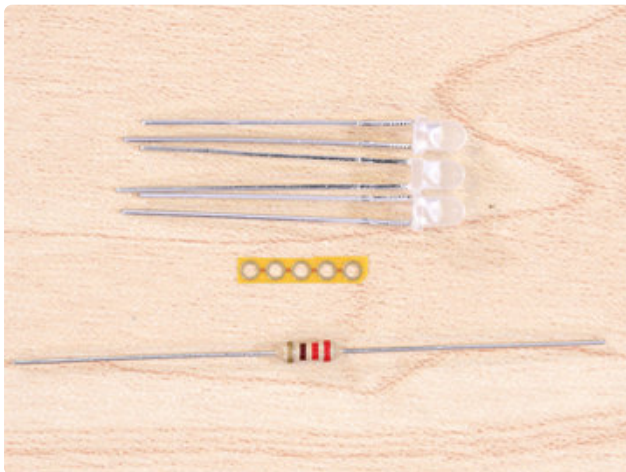
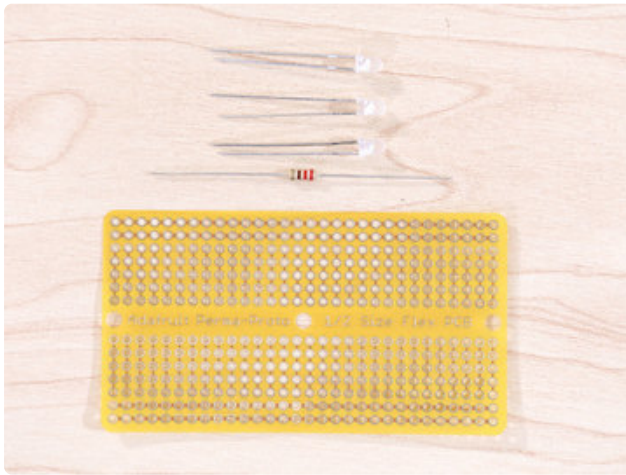
The top cover should have a friction fit to the case, but you might find it difficult to snap together. Tolerances will differ depending on the 3D printer. If you find the pieces are too tight, you may need to use a filing tool on the edges to loosen up the tolerances.

CNC Milling Acrylic

The Im-cover.stl part was designed to be CNC milled or laser cut. Although it can be 3D printed, the details on the top will not appear because they're produced via CAM tools. The STL and source file is available to download so you can adjust the CAM tools to work with a number of milling machines.

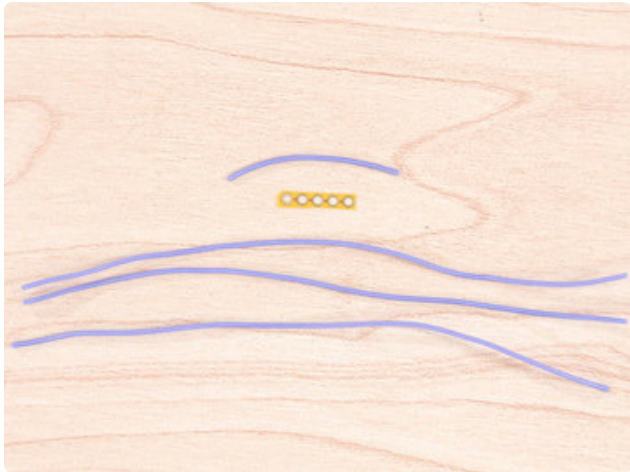


Wiring LEDs



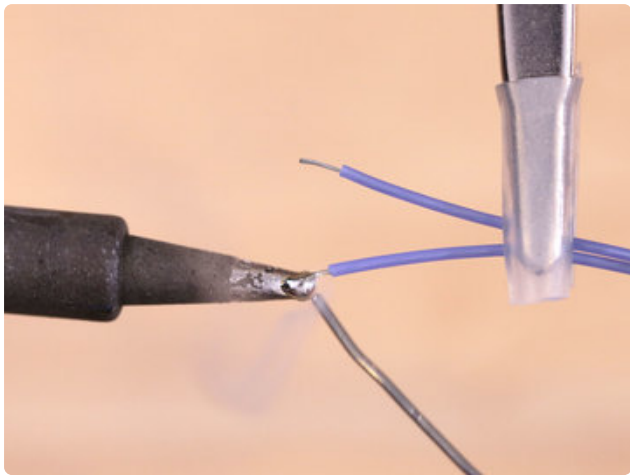
Prepping LEDs

Let's start off by getting our parts together for wiring our LEDs. In this project, I used a piece of flexible PCB, a 220ohm resistor and 3x 3mm LEDs.



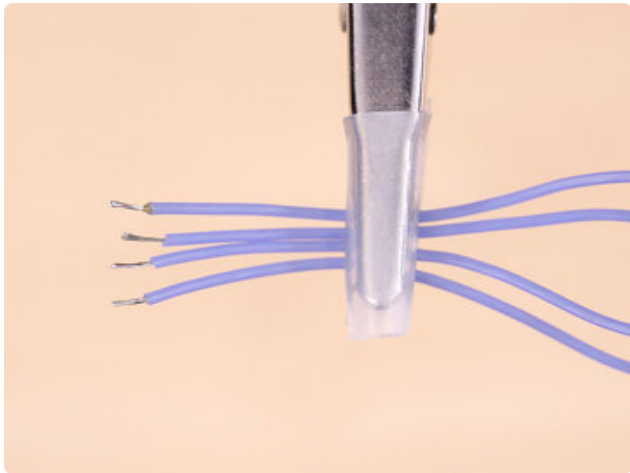
Ground Wires

We'll need some wires to connect our LEDs together. Here, I'm using 30AWG silicone coated stranded wire. I'll started with making four pieces of wire for our ground connections. Three of these pieces are 10cm in length. The four, is shorter and a bit longer than the piece of flex PCB.

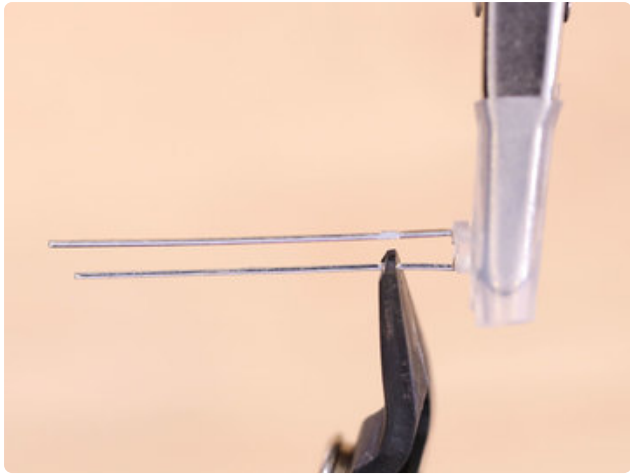


Tinning Wires

Next, remove 2-3mm of insulation from the tips of each wire. Then, apply some solder to the tips to tin them. This will prevent the stranded wire from fraying and makes it easier to solder to LEDs and the GPIO.

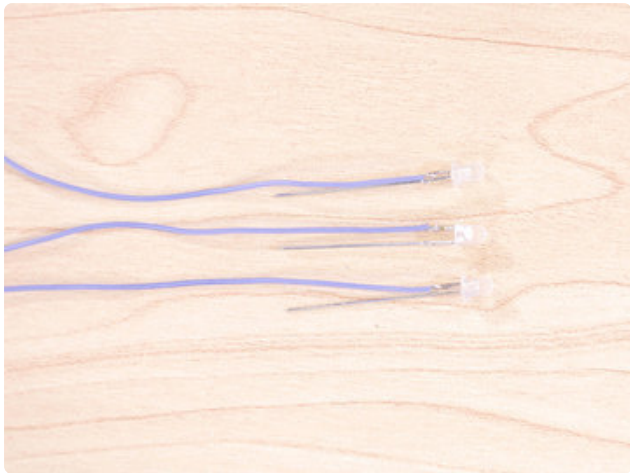


Pro Tip: Use a helping third hand to hold multiple wires to tin more than one wire at a time.



LED Ground Wire

Trim the **short** leg from the LED - This is the ground (cathode). Then, apply some solder to it and connect one of the three long wires by soldering it in place.



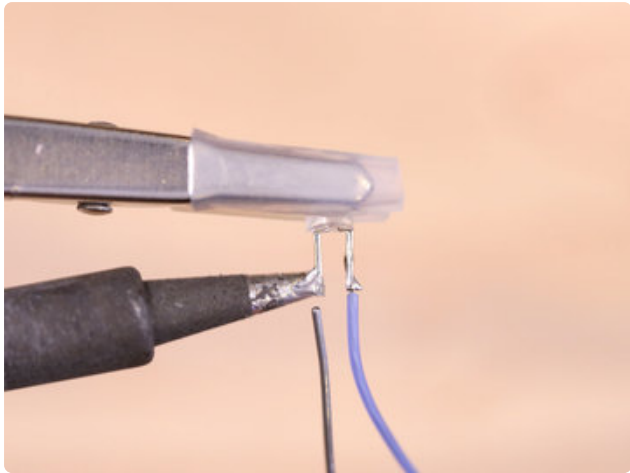
Repeat LED Ground Wires

Repeat this process for the other two LEDs.



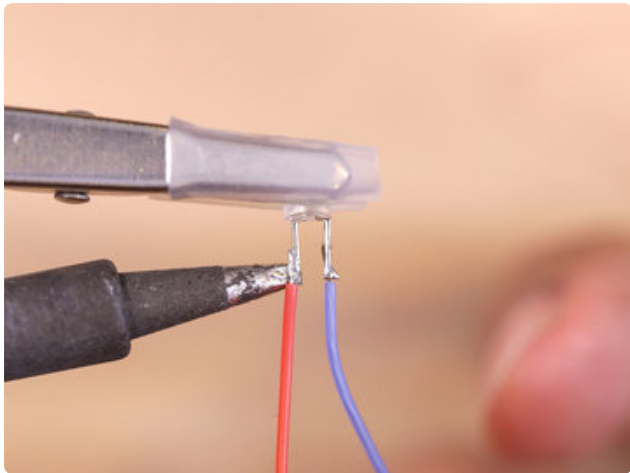
Prep Signal Wires

Next up, let's prep our wires for the signal connections. You only need 3 wires, this can be different colors to help tell them apart. Same length (10cm) just like the ground wires.



Wire Signal

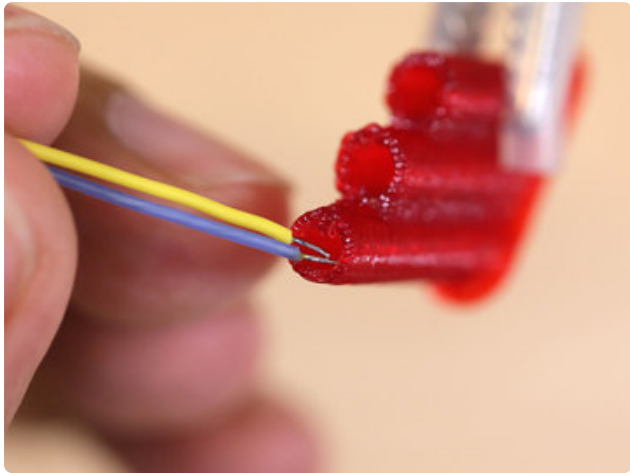
Just like we did for the ground (cathode), trim the power (anode) legs from the LED and apply solder to tin. Then, connect one of the signal wires to it.





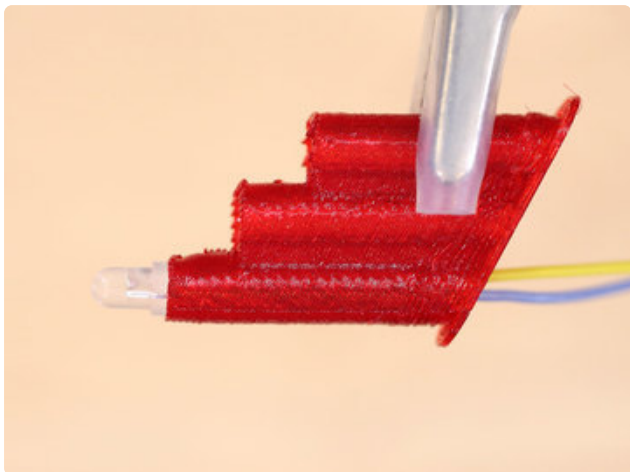
Wired LEDs

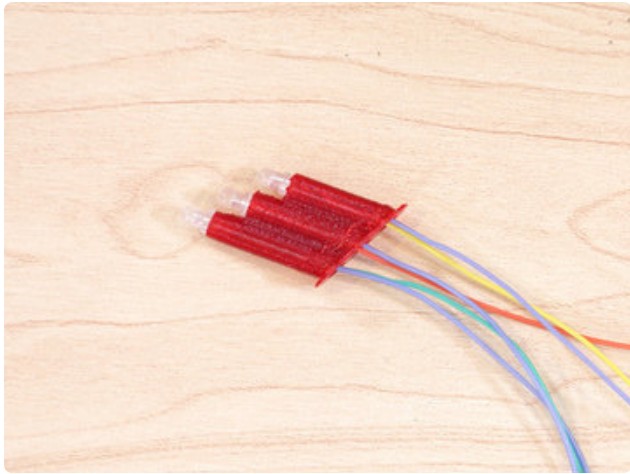
Now we have our three LEDs wired up.



Installing LEDs

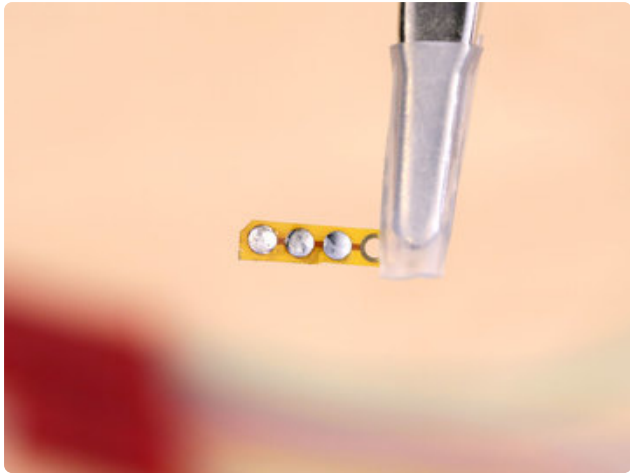
We can now install the LEDs into the 3D printed antenna part. Insert the ground and signal wires from one of the LEDs into one of the holes in the part. Thread it through and pull the wire until the head of LED is touching the antenna part. It should have a snug fit.





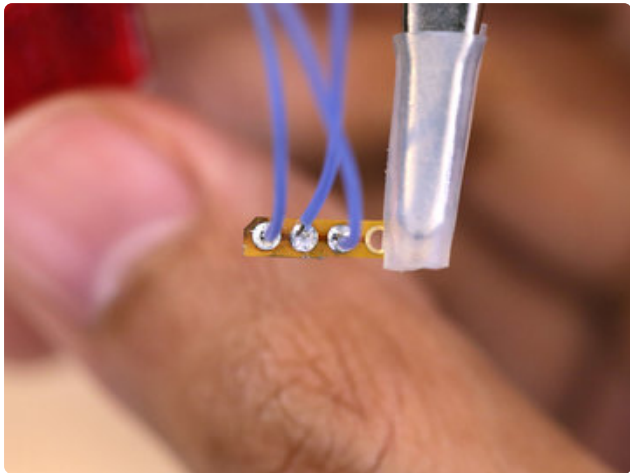
Installed LEDs

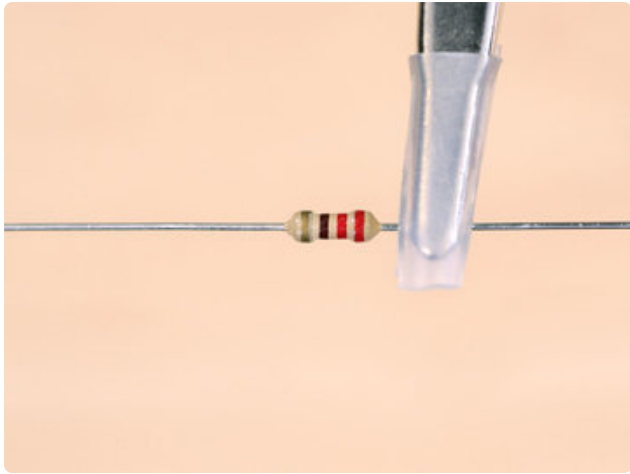
Repeat for the other two LEDs.



Connecting Ground Wires

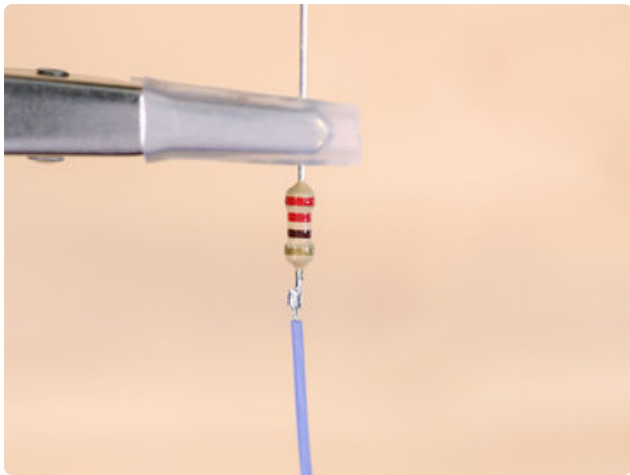
Now it's time to connect our ground wires together. Start by tinning the holes on the flexible PCB. This will help connect the ground wires to them easier. Then, solder each wire into a pad on the flexible PCB.

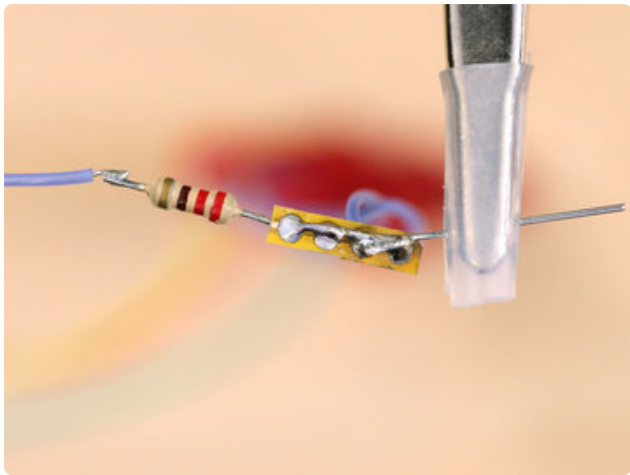
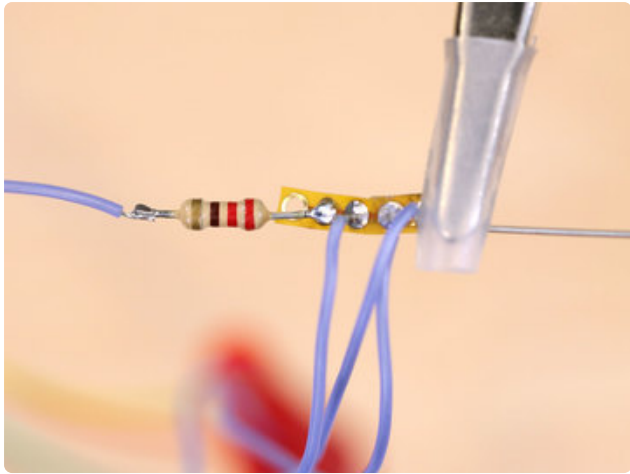




Wiring Resistor

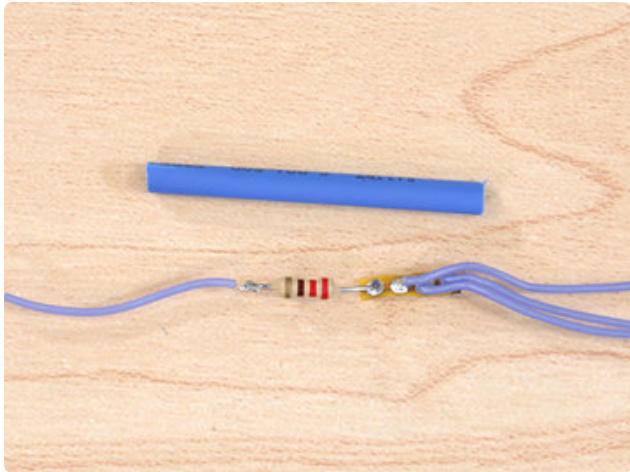
We'll need to connect a resistor to avoid burning out the LEDs. Cut one of the legs from the resistor short and solder on the short ground wire by tinning.





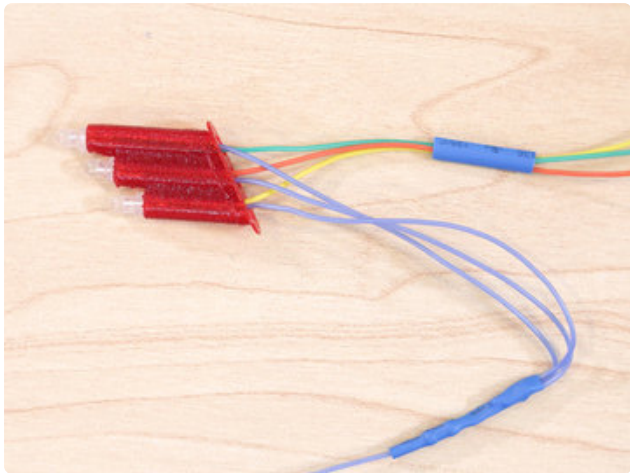
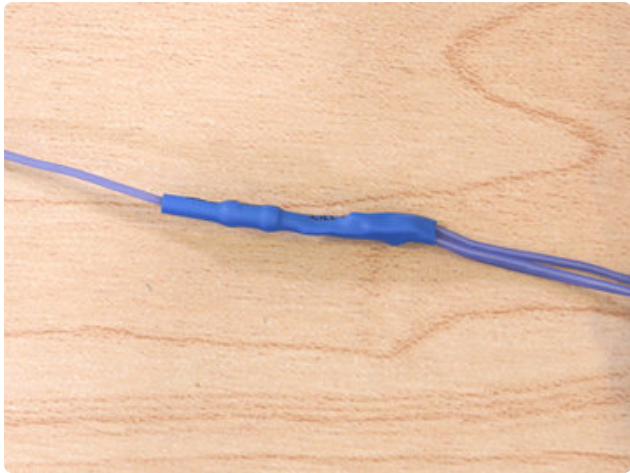
Connecting Resistor

Insert the remaining leg from the resistor into the last hole on the flexible PCB. Then, solder the leg in place. Flip the flexible PCB over and solder the other ground wires to the leg - this will make the flexible PCB strong and less likely to break. After that, you can trim the excess leg from the resistor.



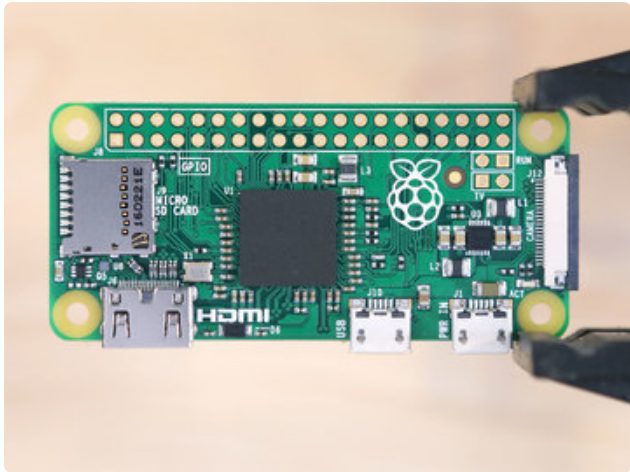
Heatshrink Connections

We'll need to insulate to expose resistor and flex PCB so it doesn't short anything out when we install it on the Pi. We can do this by using a piece of heat shrink tubing (or electrical tape). Slip the piece over the short ground wire and heat it up to tighten the tubing.



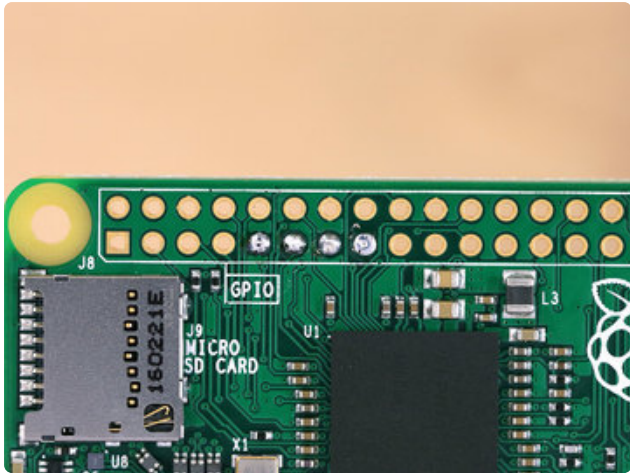
Assembled LEDs

Alright, now we have our LEDs assembled and ready to connect to the Raspberry Pi!

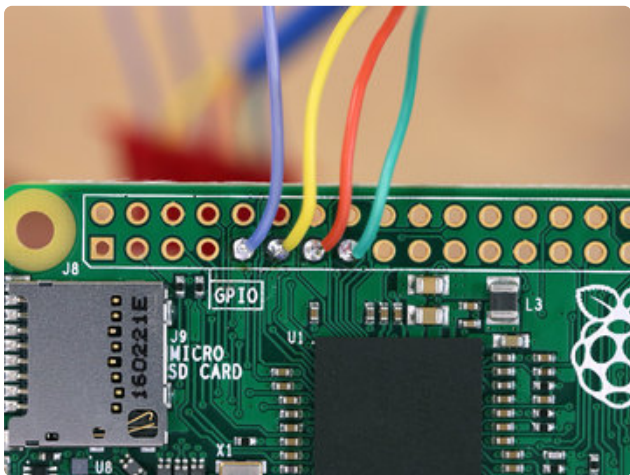


Tinning GPIO

Let's add some solder to the GPIO listed here:



Ground
GPIO 17
GPIO 27
GPIO 22



Connect LEDs to GPIO

Now it's time to solder the signal wires from the LEDs to the GPIO pins on the Raspberry Pi. You'll want to take note of which wire is connected to which LED on the antenna part. Below, I've listed out the LEDs, wire color, GPIO and pin number.

Ground - Blue Wire - Pin #9
Tallest LED - Green Wire - GPIO 22 (Pin #11)
Middle LED - Red Wire - GPIO 27 (Pin #13)
Short LED - Yellow Wire - GPIO 17 (Pin #15)

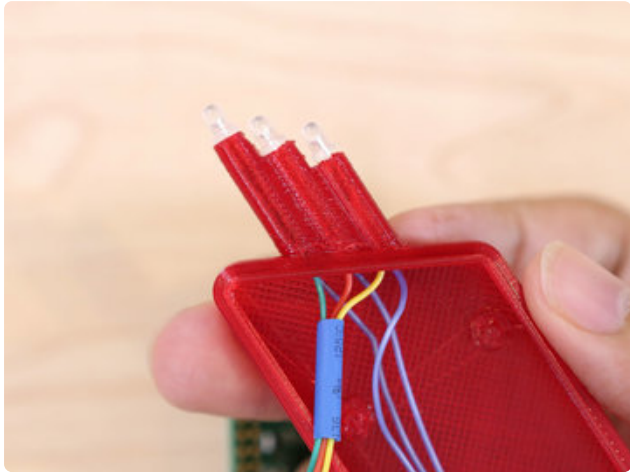
Assemble

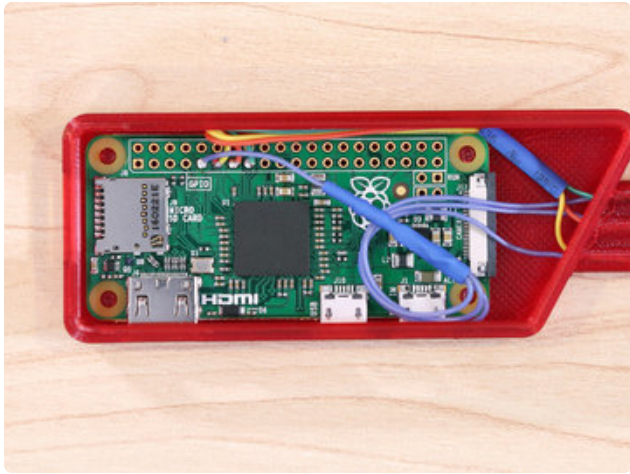


Install Antenna into Case

Carefully insert the 3D printed antenna piece into the large cutout on the case.

Press the part all the through until it snaps into place - it should have a snug fit and stay in place.





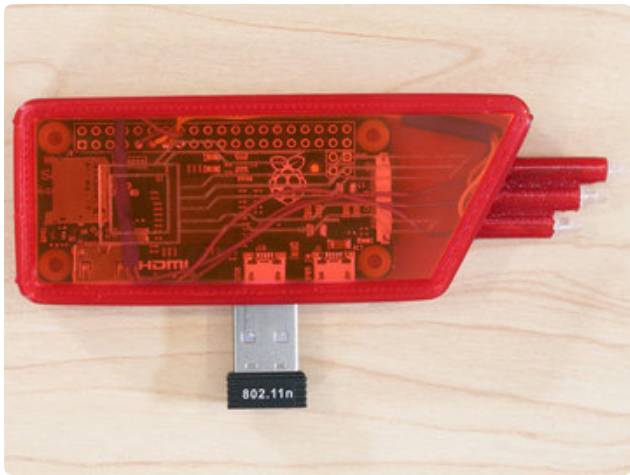
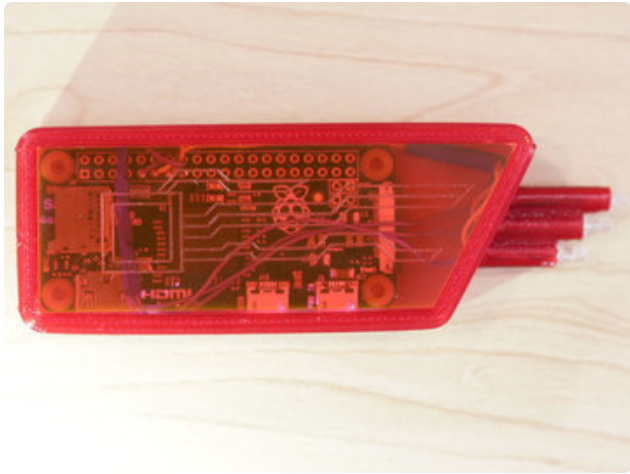
Install Pi

Orient the Raspberry Pi board so the microSD, USB and HDMI ports are lined up with the cutouts in the case. Then, press the board down until the nubs from the standoffs are inserted into the mounting holes on the Raspberry Pi.



Installing Cover

The acrylic top can be press fitted into the cutout on the case cover. It'll have a snug fit and hold in place when its press fitted onto the case.



Installed Cover

The cover has a pretty tight fit. Start by claming it on the corners of the slanted edge and press each corner of the cover into the case. Pro Tip - Use the surface of a table to help while applying force to the two pieces until they snap together.

Raspberry Pi Lure Module Case

And that's it! Now you have a pretty sweet looking case for the Raspberry Pi. If Niantic stops the python scripts from ever working again, at least you have a dope looking case! You can also change up the python code to make the LED blink, since they are already connected ;-)