



## MASLOW: an Open WiFi Detector with Adafruit Pro Trinket and CC3000

Created by Phillip Burgess



Last updated on 2018-08-22 03:44:04 PM UTC

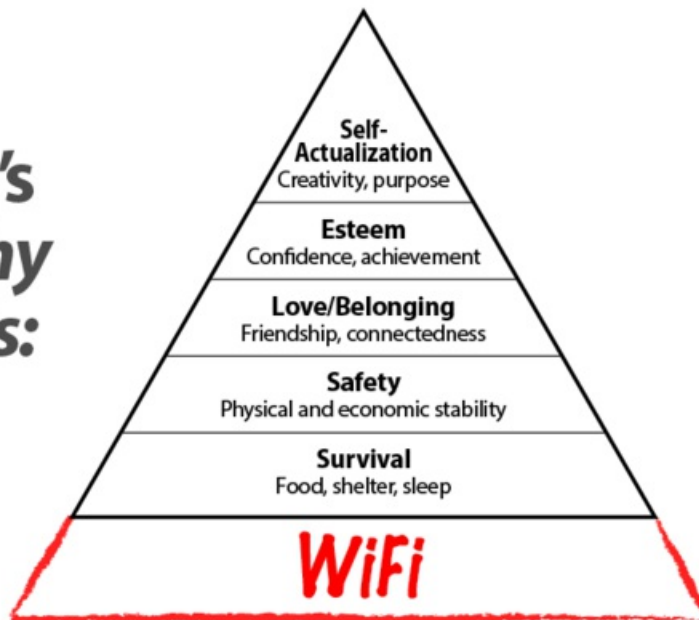
## Guide Contents

Guide Contents	2
Overview	3
How does it work?	3
Prerequisite Guides	4
Parts	4
Tools & Supplies	4
3D Printing	6
FDM 3D Printing	6
Light Diffusion	6
Slicing Software	6
PLA or ABS Material	7
Code	8
Assembly	12
Measure wire	13
LiPoly backpack	14
LiPoly backpack headers	14
Share Power cable with headers	15
Solder CC3000	15
LED and resistor	16
uFL cable	16
Insert Circuit into Enclosure	17
Slide switch	17
Battery	17
Wearable clip	18
Wear it!	18

## Overview

Traveling or on the road, sometimes finding an open network to get online and finish a job is more important than lunch or a nap. MASLOW (*Mini Adafruit System Locates Open Wireless*) is a tiny battery-powered WiFi detector with a 3D-printed enclosure...assemble one yourself and clip it to your laptop bag.

### Maslow's Hierarchy of Needs:



### How does it work?

In this project, we're making a portable WiFi hotspot finder using the Adafruit CC3000 and Pro Trinket micro-controller. The CC3000 is a WiFi breakout board that adds connectivity to your micro-controller projects.

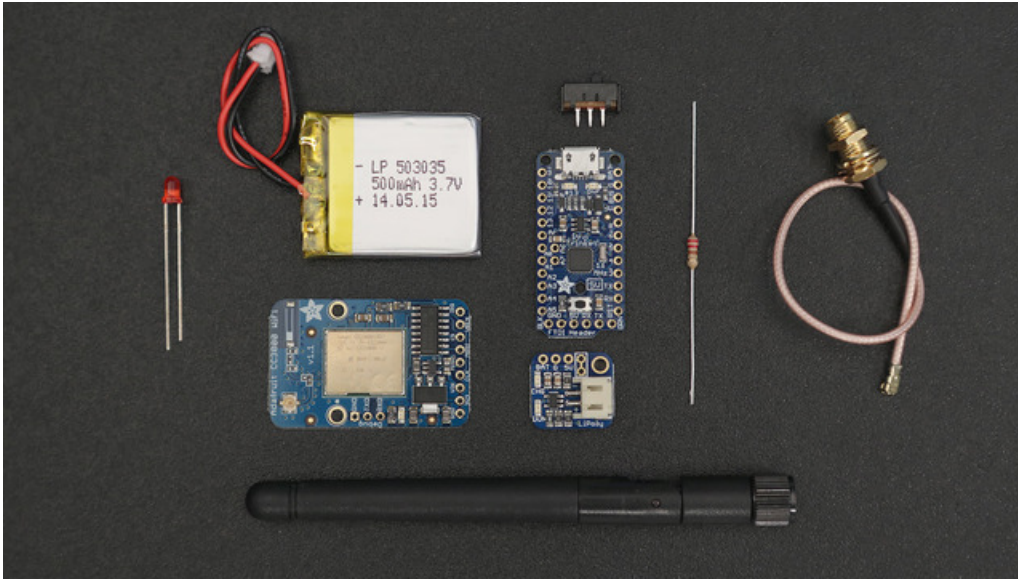
5 total modes indicate the status. When the LED starts flickering fast, it's scanning for WiFi networks. A slow blink indicates an open network is in range. 1 second blips mean closed networks only, and 4 second blips means no WiFi networks.



## Prerequisite Guides

We recommend walking-through the following tutorials below before starting this project. These guides will help you get familiar with the components and get setup and configured with the Arduino IDE and libraries.

- [Introducing Pro Trinket \(https://adafru.it/nXF\)](https://adafru.it/nXF)
- [Adafruit Pro Trinket LiPoly/Lilon Backpack \(https://adafru.it/iPd\)](https://adafru.it/iPd)
- [Adafruit CC3000 WiFi \(https://adafru.it/itc\)](https://adafru.it/itc)



## Parts

We have all the lovely components and tools to build this project. Be sure to check out the featured products on the right sidebar.

- [Pro Trinket \(http://adafru.it/2000\)](http://adafru.it/2000)
- [Trinket Lilon/LiPoly Backpack Add-On \(https://adafru.it/e0w\)](https://adafru.it/e0w)
- [CC3000 Breakout with external Antenna \(http://adafru.it/1510\)](http://adafru.it/1510) (2 - 5dBi) or [Internal antenna \(http://adafru.it/1469\)](http://adafru.it/1469) (~2dBi)
- [uFL Adapter Cable \(http://adafru.it/852\)](http://adafru.it/852) (you can also use the version of the CC3000 w/out external antenna and skip this part)
- [Breadboard-friendly SPDT Slide Switch \(http://adafru.it/805\)](http://adafru.it/805)
- [SMA Antenna \(https://adafru.it/e5m\)](https://adafru.it/e5m) (you can also use the version of the CC3000 w/out external antenna and skip this part)
- LiPoly battery. The standard case fits a [100 mAh battery \(http://adafru.it/1570\)](http://adafru.it/1570), or with minor editing this can hold a larger [500 mAh battery \(http://adafru.it/1578\)](http://adafru.it/1578), making a single charge last all day.

## Tools & Supplies

You'll need a couple of hand tools and accessories to assist you in the build.

- [Solder Iron \(http://adafru.it/1204\)](http://adafru.it/1204) + [Solder \(http://adafru.it/734\)](http://adafru.it/734)
- [Silicone Wire \(http://adafru.it/1877\)](http://adafru.it/1877)
- [PLA Filament \(https://adafru.it/dtp\)](https://adafru.it/dtp)
- [3D Printer \(https://adafru.it/tkf\)](https://adafru.it/tkf)



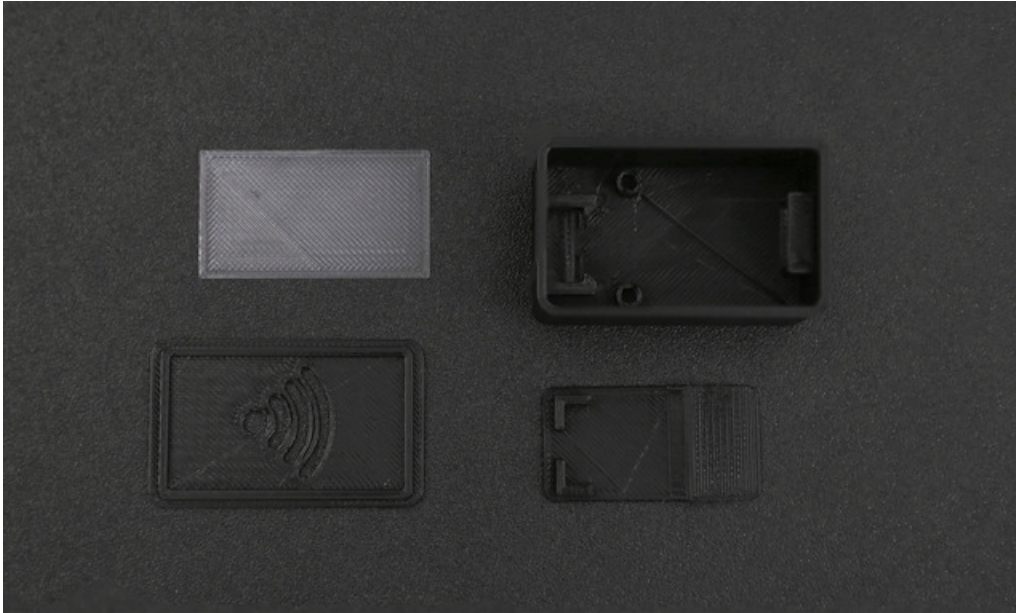
## 3D Printing

### FDM 3D Printing

These parts are optimized to print with desktop 3D Printers capable of printing in ABS or PLA material with a minimum build area of 100mm x 100mm x 90mm. The two parts are designed to print **without** any support material.

### Light Diffusion

The **wifiDif.stl** part should be printed in either transparent or a light colored material. This is the part that will diffuse the LED so that the WiFi logo in the cover illuminates.



The standard case design fits a tiny 100 mAh LiPoly battery for maximum portability. With some simple editing of the walls, this can be made to fit a larger 500 mAh battery, so a single charge lasts all day long.

<https://adafru.it/CdO>

<https://adafru.it/CdO>

<https://adafru.it/e80>

<https://adafru.it/e80>

wifiCase.stl	235c	about 2 hours to print all four parts
wifiClip.stl	10% infill	
wifiDif.stl	90 feed	
wifiCover.stl	120 travel	
	no supports or raft	

### Slicing Software

The recommend settings above should work with most slicing software. However, you are encouraged to use your own settings since 3D printers and slicing software will vary from printer to printer.

## PLA or ABS Material

We recommend using PLA material for an easier print with high quality. The tolerance has been tested with PLA filament but should also work with ABS. The parts do not require any support material or a raft.

## Code

Our WiFi-finder sketch depends on the Adafruit\_CC3000 and TimerOne libraries. Library installation is a common sticking point for beginners...our [All About Arduino Libraries \(https://adafru.it/dit\)](https://adafru.it/dit) guide explains how this is done.

```
https://adafru.it/cHe
```

```
https://adafru.it/cHe
```

```
https://adafru.it/e5h
```

```
https://adafru.it/e5h
```

After installing the libraries, restart the Arduino IDE.

The code will work on an **Adafruit Pro Trinket** or an **Arduino Uno**. It makes reference to some specific pins and hardware features (sleep, interrupts, etc.) that may not work on other boards without some code changes and deeper understanding of the hardware.

```
/* -----  
WIFI HOTSPOT DETECTOR for Adafruit Pro Trinket and CC3000 WiFi adapter  
(can also work with Arduino Uno, etc.). Periodically scans for open  
networks (those with no password required), LED on pin 9 shows status:  
  
Steady on    = One-time initialization  
Fast flicker = Scanning for WiFi networks  
Slow blink   = Open networks in range  
1 sec blips  = Closed networks only  
4 sec blips  = No WiFi networks (open OR closed) detected  
  
CC3000 boards with integrated chip antenna are most compact but have  
limited sensitivity. The uFL connector versions improve range, but  
require adding an adapter and antenna, making a larger package.  
5dBi antenna is more sensitive than most laptops; 2dBi is ideal.  
  
https://www.adafruit.com/product/2000 Pro Trinket 5V 16 MHz  
https://www.adafruit.com/product/2010 Pro Trinket 3V 12 MHz  
https://www.adafruit.com/product/2124 Pro Trinket LiIon/LiPoly backpack  
https://www.adafruit.com/product/1570 Lithium Polymer 100 mAh battery  
https://www.adafruit.com/product/1578 Lithium Polymer 500 mAh battery  
https://www.adafruit.com/product/1469 CC3000 breakout w/ chip antenna  
https://www.adafruit.com/product/1510 CC3000 breakout w/ uFL connector  
https://www.adafruit.com/product/852 uFL->RP-SMA adapter  
https://www.adafruit.com/product/944 4" 2dBi gain RP-SMA antenna  
https://www.adafruit.com/product/945 8" 5dBi gain RP-SMA antenna  
  
Requires Adafruit_CC3000 and TimerOne libraries:  
  
https://github.com/adafruit/Adafruit_CC3000_Library  
https://github.com/PaulStoffregen/TimerOne  
  
Adafruit invests time and resources providing this open source code,  
please support Adafruit and open-source hardware by purchasing products  
from Adafruit!  
  
Written by Phil Burgess and Tony DiCola for Adafruit Industries.  
BSD license all text above must be included in any redistribution
```



CC BY license, all text above must be included in any redistribution.

-----\*/

```
#include <Adafruit_CC3000.h>
#include <TimerOne.h>
#include <SPI.h>
#include <avr/sleep.h>
#include <avr/power.h>

// Hardware-based PWM is used to drive the status LED. This means it's tied
// to a specific pin and can't be moved, nor can this pin be used for CC3000
// interfacing or other functions. The exact pin number may vary from board
// to board -- in low-level AVR pin maps it's identified as OC1A. This is
// DIGITAL PIN 9 on the Adafruit Pro Trinket, Arduino Uno, Duemilanove,
// Diecimila and Leonardo. Looks like this is pin 14 on a PJRC Teensy 2.0 or
// pin 11 on Arduino Mega (but this is a tiny sketch and a waste of a Mega).
// This goes to the anode (+) leg of the LED, cathode (-) goes to GND.

// The CC3000 interrupt and control pins are configurable...
#define ADAFRUIT_CC3000_IRQ 3 // MUST be an interrupt pin!
#define ADAFRUIT_CC3000_VBAT 5 // These can be
#define ADAFRUIT_CC3000_CS 10 // any two pins.
// Hardware SPI for remaining pins. On Pro Trinket, SCK=13, MISO=12, MOSI=11
Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
    ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT, SPI_CLOCK_DIVIDER);

// AVR sleep mode is used to conserve power. Wake is initiated by the
// watchdog timer (WDT), set later in this code for 8-second intervals.
// Because the CC3000 library is still a bit experimental in spots, the WDT
// is also used to detect code lockups...the interrupt handler counts down
// and if it reaches zero the WDT is switched to system reset mode; the
// sketch resets the counter periodically to indicate continued "OK" state.
const uint8_t      // These are the number of 8-second WDT interrupts...
    sleepIntervals = 7, // ...while sleeping, between SSID scans
    runIntervals    = 7; // ...during SSID scan, max # before reset issued.
volatile uint8_t    // 'volatile' because interrupt may change this
    wdtCount        = runIntervals+1; // Init counter to SSID scan max time
// Although the WDT interval is 8 seconds, the first interval may already
// be partly elapsed when the counter is reset...so although the sleep
// time is set for 7 intervals -- 56 sec -- in reality it might be up to
// one full interval shorter -- 48 sec. The SSID scan typically just needs
// a few seconds to run, but 7 intervals are allotted for good measure.
// If it's finished before then, that's fine and normal sleep commences.
// If it's NOT finished, code is presumed locked up and the system resets.

// -----

void setup(void) {

    // Watchdog timer setup. Abbreviated here; original from Tony DiCola's
    // excellent Low Power WiFi Datalogger guide:
    // https://learn.adafruit.com/low-power-wifi-datalogging
    noInterrupts();
    MCUSR  &= ~_BV(WDRF);
    WDTCSR = _BV(WDCE) | _BV(WDE); // WDT change enable
    WDTCSR = _BV(WDIE) | _BV(WDP3) | _BV(WDP0); // Interrupt enable, 8 sec.
    interrupts();

    // Initial pin 9 LED state is always-on. If the CC3000 init fails,
    // the LED remains in this state to indicate there's trouble.
```

```

Timer1.initialize();
Timer1.pwm(TIMER1_A_PIN, 1023); // 1023 = 100%

// Serial console messages help with debugging, but aren't seen in
// normal use. Pro Trinket requires FTDI adapter for Serial I/O.
Serial.begin(57600);
Serial.print(F("Initializing CC3000..."));
if(!cc3000.begin()) {
  Serial.println(F("failed. Check your wiring?"));
  for(;;);
}
Serial.println(F("OK.));

// AVR peripherals that aren't used by this code are disabled to further
// conserve power, and may take certain Arduino functionality with them.
// If you adapt this code to other projects, may need to re-enable some.
power_adc_disable(); // Disable ADC (no analogRead())
power_twi_disable(); // Disable I2C (no Wire library)
power_timer2_disable();

set_sleep_mode(SLEEP_MODE_IDLE);
}

// -----
const uint8_t PROGMEM
  mode0[] = "Open",
  mode1[] = "WEP",
  mode2[] = "WPA",
  mode3[] = "WPA2",
  *modes[] = { mode0, mode1, mode2, mode3 };

void loop(void) {
  uint32_t count;
  uint8_t rssi, sec;
  char ssid[33];

  // PWM set to 10 Hz during scan w/ short 'blip' duty cycle.
  // This line can be commented out for a more discreet scan,
  // LED will retain prior blink until new state is known.
  Timer1.pwm(TIMER1_A_PIN, 10, 100000);

  Serial.print(F("Scanning..."));
  if(cc3000.startSSIDscan(&count)) { // Initiate SSID scan
    // Default LED state is 'no networks found' -- a short 'blip'
    // every 4 seconds provides a quiet heartbeat:
    Timer1.pwm(TIMER1_A_PIN, 1, 4000000); // 4 sec, tiny duty cycle
    if(count) { // Networks found! How many?
      Serial.print(count); Serial.print(F(" network"));
      if(count > 1) Serial.print('s');
      Serial.println(F(" found:"));
      // Networks found, but open/closed state not yet known.
      // Presume closed only, shorten heartbeat to 1 sec.
      Timer1.pwm(TIMER1_A_PIN, 1, 1000000); // 1 sec, tiny duty cycle

      while(count--) { // For each network...
        if(cc3000.getNextSSID(&rssi, &sec, ssid)) { // Valid?
          if(sec > 3) sec = 3;
          // Dump basic info to Serial console for -all- networks
          Serial.print(F("SSID: ")); Serial.println(ssid);
        }
      }
    }
  }
}

```

```

Serial.print(F(" RSSI      : ")); Serial.println(rssi);
Serial.print(F(" Security: "));
Serial.println((__FlashStringHelper *)pgm_read_word(&modes[sec]));
if(sec == WLAN_SEC_UNSEC) { // If open network...
  // Switch LED to conspicuous 'open networks' flash immediately
  Timer1.pwm(TIMER1_A_PIN, 511, 1000000); // 1 sec, 50% duty cycle
  // "Open hotspot" is as good as the indicator gets and the scan
  // can stop now, get into power-saving sleep mode ASAP.
  // If you're using the Serial console and want to see all
  // networks displayed, comment out this line:
  break;
}
}
} else Serial.println(F("no networks found."));
} else Serial.println(F("failed!"));

cc3000.stopSSIDscan();

Serial.print(F("Sleeping..."));
wlan_stop();

// Additional AVR peripherals are now shut down, most everything except
// Timer/Counter 1, used for the status LED. 'Idle' mode is the deepest
// sleep level that still allows this timer to continue running.
power_spi_disable(); // Disable SPI (CC3000 comms)
delay(50);           // Stall for last serial output
power_timer0_disable(); // Disable millis() etc.
power_usart0_disable(); // Disable Serial

// WDT interrupt counts down 8-second intervals from sleepIntervals+1
// to 1 (NOT 0 -- that's when a lockup-busting system reset is performed).
for(wdtCount=sleepIntervals+1; wdtCount>1; ) sleep_mode();
// Done sleeping, set counter to max SSID scan time before sys reset
wdtCount = runIntervals+1;

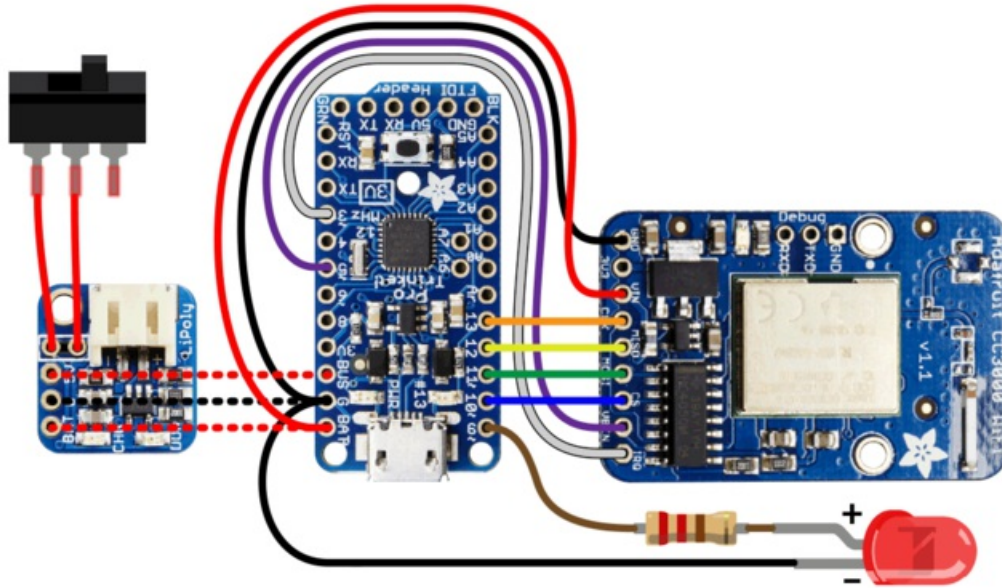
// Restore AVR peripherals used elsewhere in the code
power_usart0_enable(); // Enable Serial
power_timer0_enable(); // Enable millis() etc.
power_spi_enable(); // Enable SPI (for CC3000)
wlan_start(0); // Wake up CC3000
Serial.println(F("done."));
}

ISR(WDT_vect) { // Watchdog interrupt @ 8 sec. interval
  if(!--wdtCount) { // Decrement sleep interval counter...
    // If it reaches zero, WDT switched from interrupt to system reset mode...
    noInterrupts();
    WDTCSR = _BV(WDCE) | _BV(WDE); // WDT change enable
    WDTCSR = _BV(WDP2) | _BV(WDP1) | _BV(WDE); // System reset enable, 1 sec
    for(;;); // Wait for reset
  }
}
}

```

## Assembly

This diagram shows the CC3000 WiFi Breakout with [onboard ceramic antenna](https://adafru.it/e5k) (<https://adafru.it/e5k>), but you can also build it with the [uFL connector version](https://adafru.it/e5l) (<https://adafru.it/e5l>) to add an [adapter](http://adafru.it/852) (<http://adafru.it/852>) and [antenna](https://adafru.it/e5m) (<https://adafru.it/e5m>) for extra sensitivity. The wiring is identical, the latter just requires plugging the extra parts into the connector on the CC3000 board. A lithium-polymer battery (100 mAh, or 500 mAh with some editing of the case design) plugs into the LiPoly backpack after assembly.

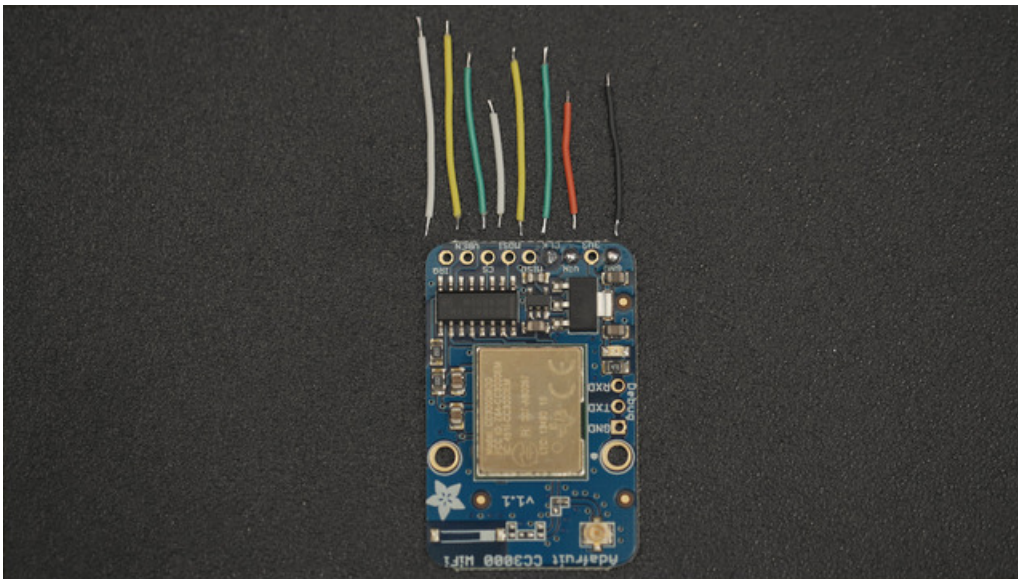


The LiPoly backpack sits atop the Pro Trinket on the **BUS**, **G** and **BAT+** pins, and the power jumpers are modified to connect a switch.

An LED (any color) and 220 Ohm resistor in series are connected to Pro Trinket pin 9 and any available **GND** pin (you can use **GND** on the FTDI header if it's easier).

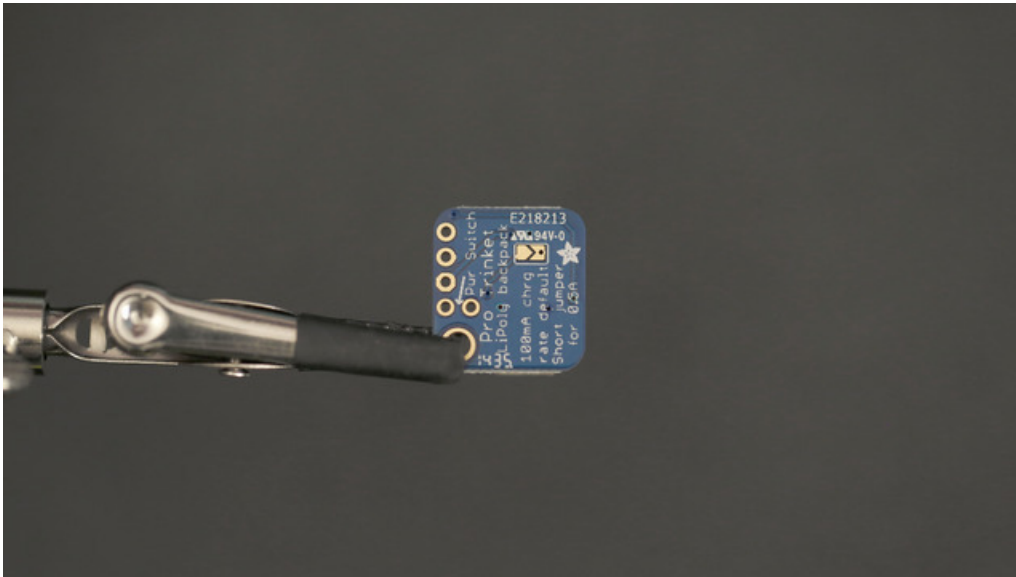
The remaining connections are:

Pro Trinket Pin	CC3000 Pin
3	IRQ
5	VBEN
10	CS
11	MOSI
12	MISO
13	CLK
G	GND
BAT+	VIN



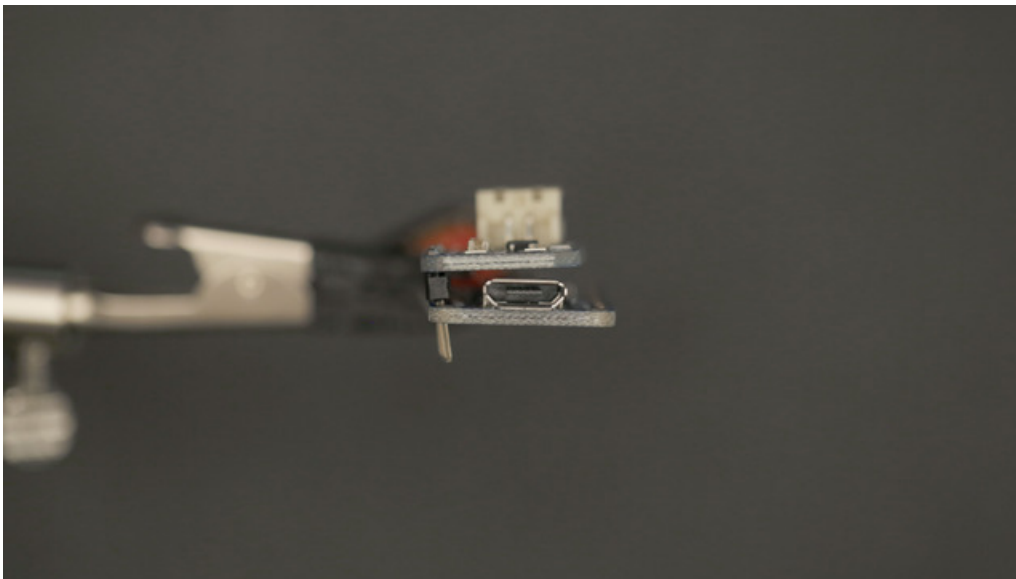
## Measure wire

Align the Pro Trinket on top of the CC3000 and measure how much length each wire needs to reach the through-holes.



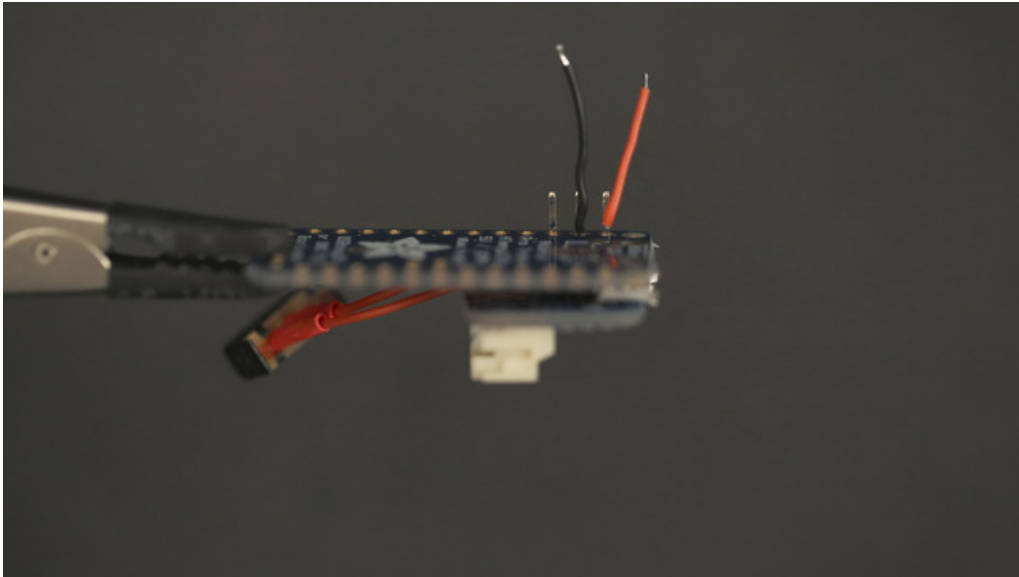
## LiPoly backpack

Cut the trace with an hobby knife that connects the two through-holes for the power switch. This will allow us to use a slide switch to power on and off our circuit.



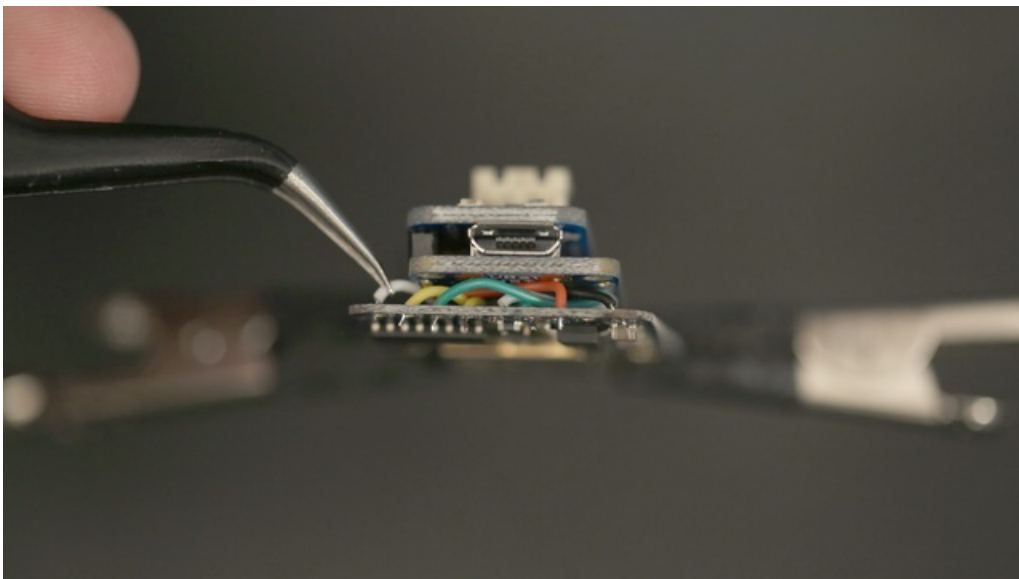
## LiPoly backpack headers

Solder the included header to the LiPoly backpack, **BAT+**, **G**, and **5V**.



### Share Power cable with headers

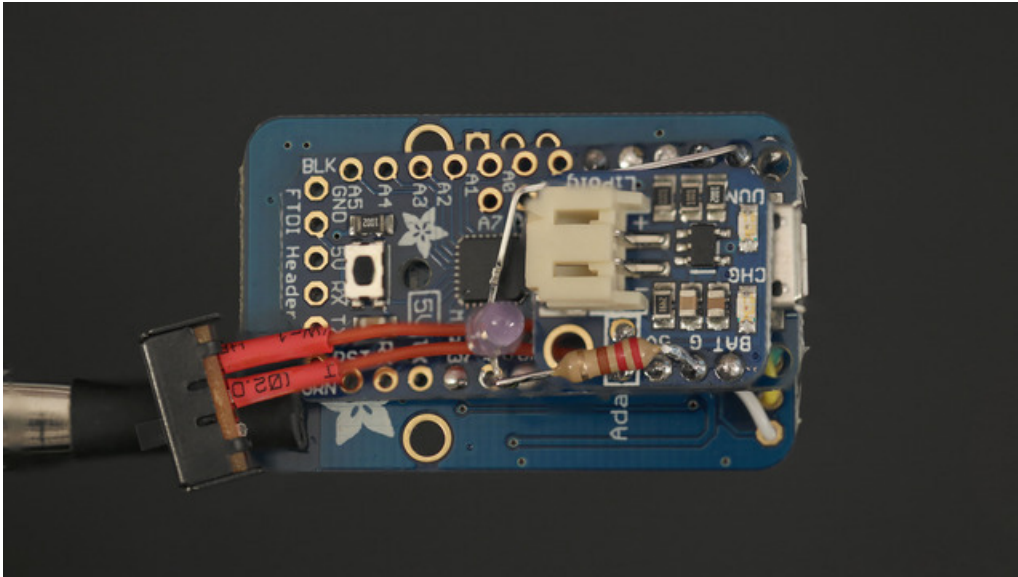
Squeeze in the power and ground cables that will power the CC3000. **BAT+** to **BAT+**, **G** to **G** and **5V** to **BUS**. Tin and use flat pliers to fit the the two cables with the headers.



### Solder CC3000

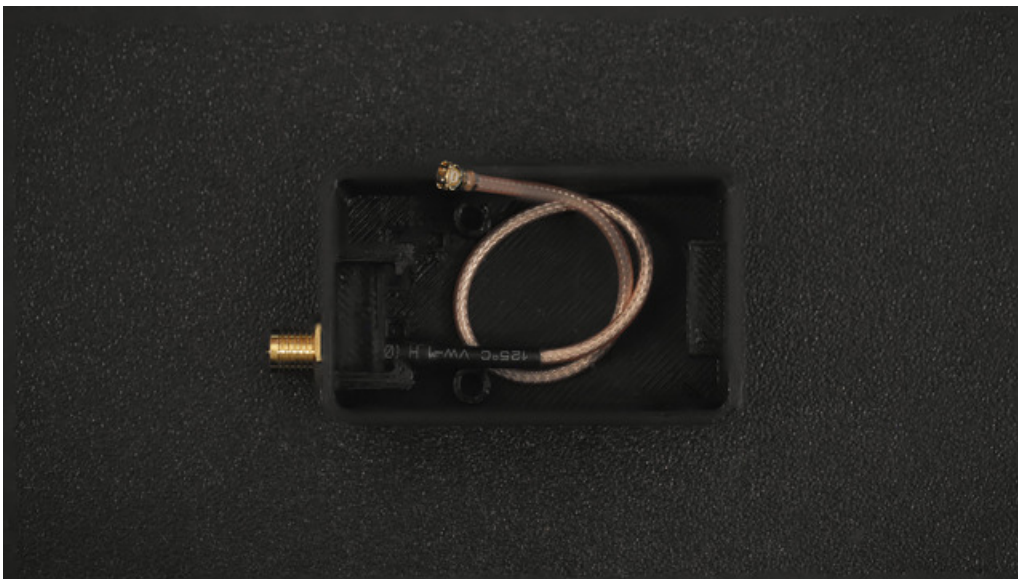
Solder and adjust any of wires so they can reach each through-hole without too much slack. Use tweezers to bend the tips to keep them in place while soldering.





## LED and resistor

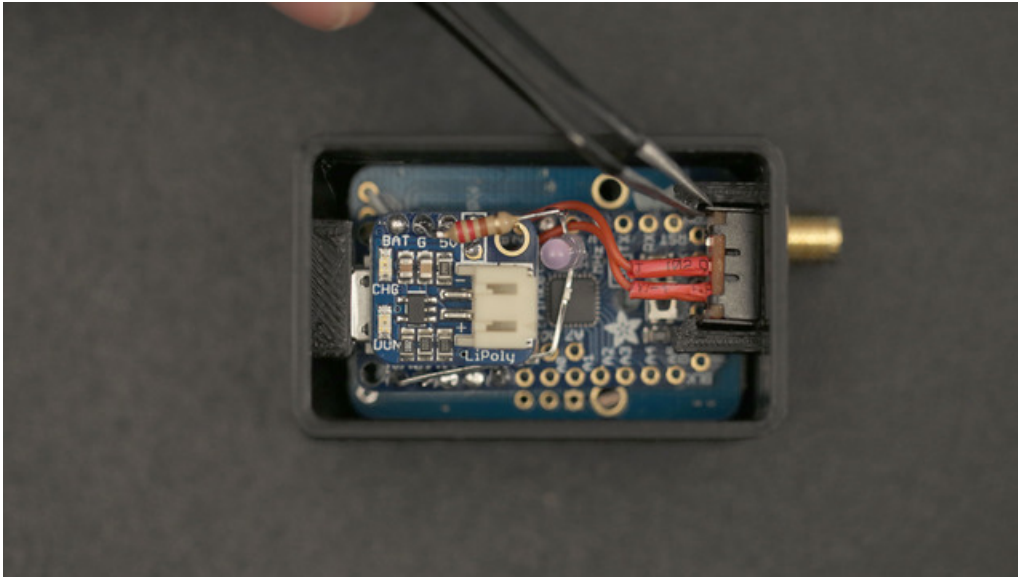
Follow the circuit diagram to see which pins to insert the LED and resistor. Keep the legs of the terminals long. Bend these leads using pliers and arrange the LED so that it shines through the cover design.



## uFL cable

Coil the uFL cable to fit inside the enclosure. Unscrew the included nut and insert into the enclosure. Screw the nut back on to secure the uFL cable to the enclosure.





## Insert Circuit into Enclosure

Carefully insert the circuit into the enclosure at an angle with the opposite side of the USB port going in first. Ensure the uFL cable isn't uncoiled (otherwise it won't fit) while inserting the circuit. The micro USB connector should be facing the port opening. This can be a tedious task, you may want to give it multiple attempts to get the fittings just right.

## Slide switch

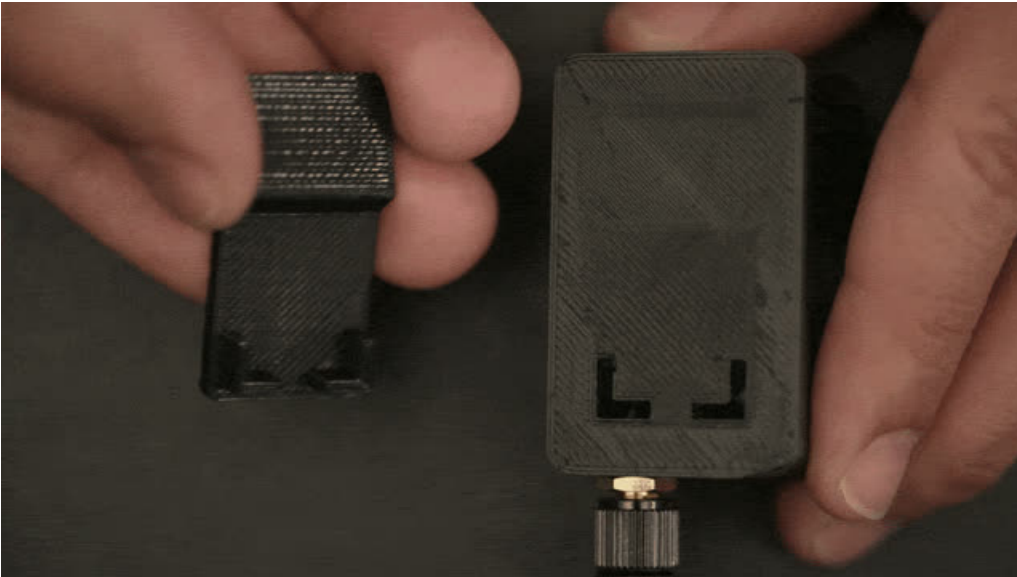
Insert the slide switch at an angle with tweezers. A bottom platform and side clips will keep the slide switch in place.



## Battery

This enclosure was designed to fit a 110mAh lithium polyer battery. Coil the battery cable and fit on either side of the enclosure.

The case design can be edited to accommodate a larger 500 mAh battery for all-day use.



### Wearable clip

Align the clip corners and apply a small amount of force to attach to the enclosure.



### Wear it!

Clip on to a backpack or on your persons to see where open wifi spot are hiding!