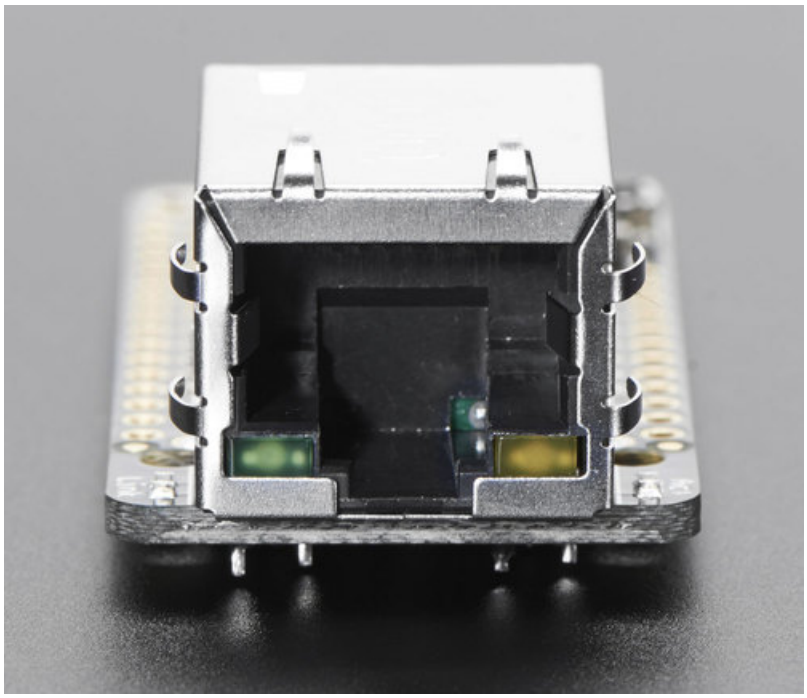
adafruit learning system

Adafruit Ethernet FeatherWing

Created by lady ada



Last updated on 2018-11-27 06:35:22 AM UTC

Guide Contents

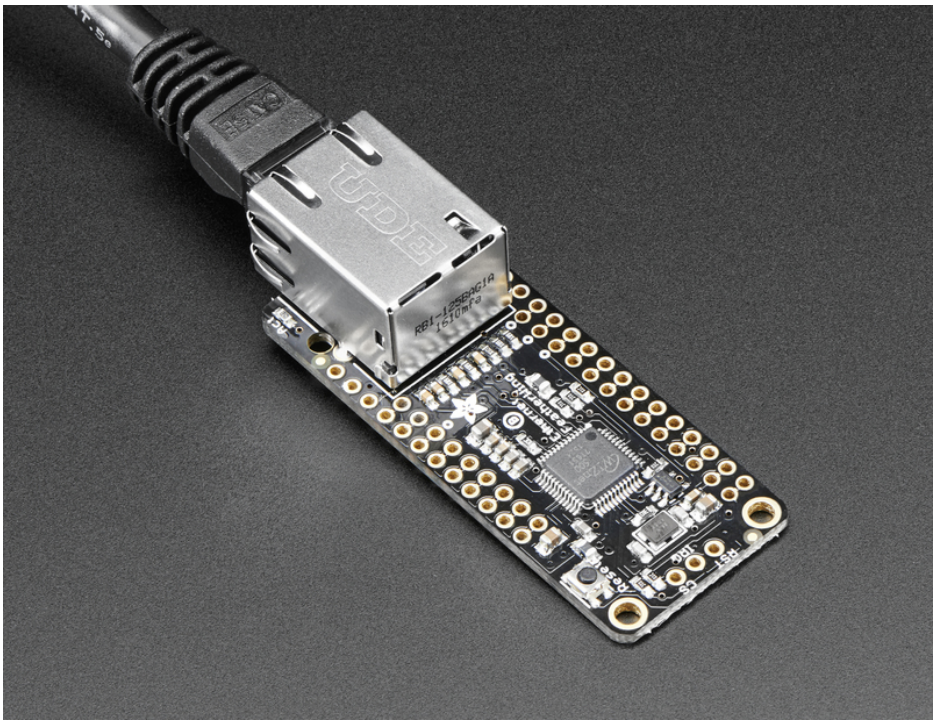
Guide Contents 2
Overview 3
Pinouts 5
Power Pins 5
SPI Data Pins 5
Other Breakouts 7
Assembly 8
- Prepare the header strip: 8
- Add the FeatherWing: 9
- And Solder! 9

Usage 13
Install Ethernet2 Library 13
Connect Cable 13
Run Example Code 14
Library Reference 17
Power over Ethernet 18
- PoE Splitter with MicroUSB Plug - Isolated 12W - 5V 2.4 Amp 18

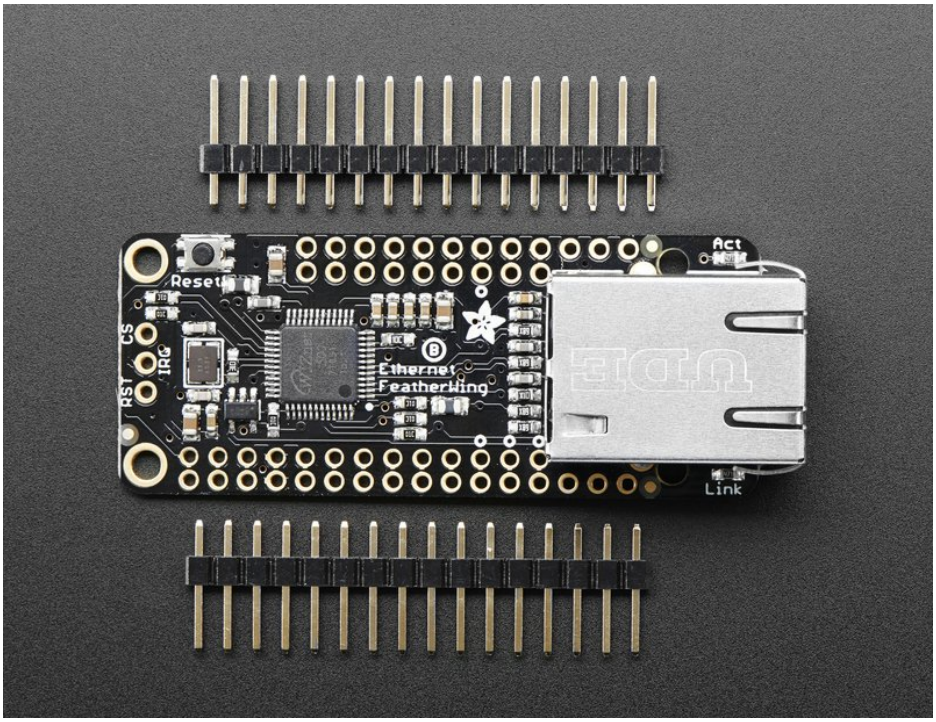
Passive PoE 18
- Passive PoE Injector Cable Set 18
- 5V 2A (2000mA) switching power supply - UL Listed 19
- MicroUSB Plug to 5.5/2.1mm DC Barrel Jack Adapter 19

Downloads 20
Files & Datasheets 20
Schematic 20
Fabrication Print 20

Overview

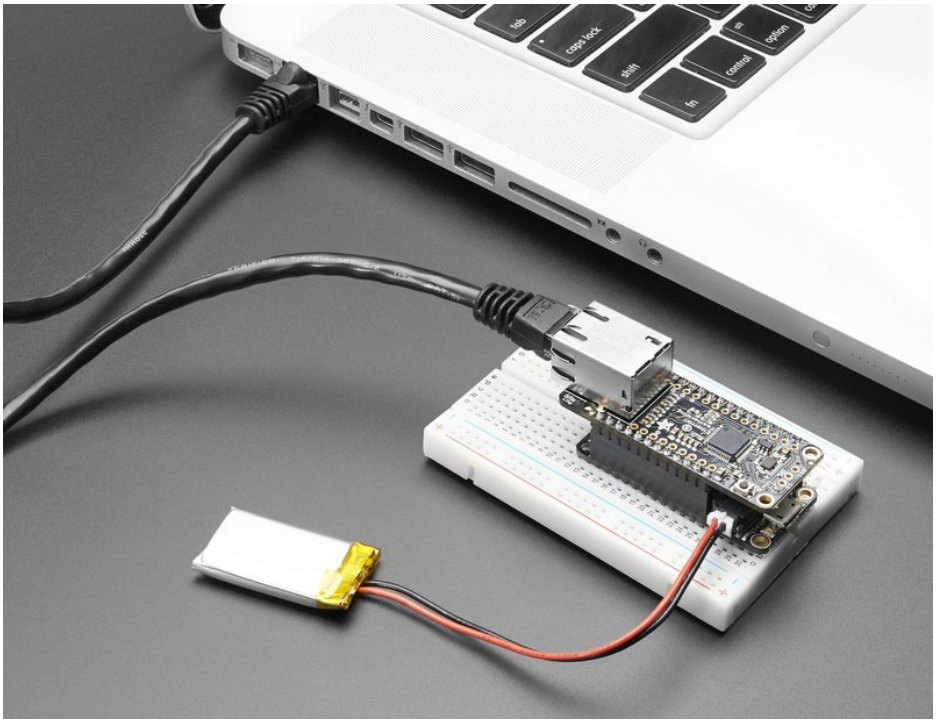


Wireless is wonderful, but sometimes you want the strong reliability of a wire. If your Feather board is going to be part of a permanent installation, this Ethernet FeatherWing will let you add quick and easy wired Internet. Just plug in a standard Ethernet cable, and run the Ethernet2 library for cross-platform networking.

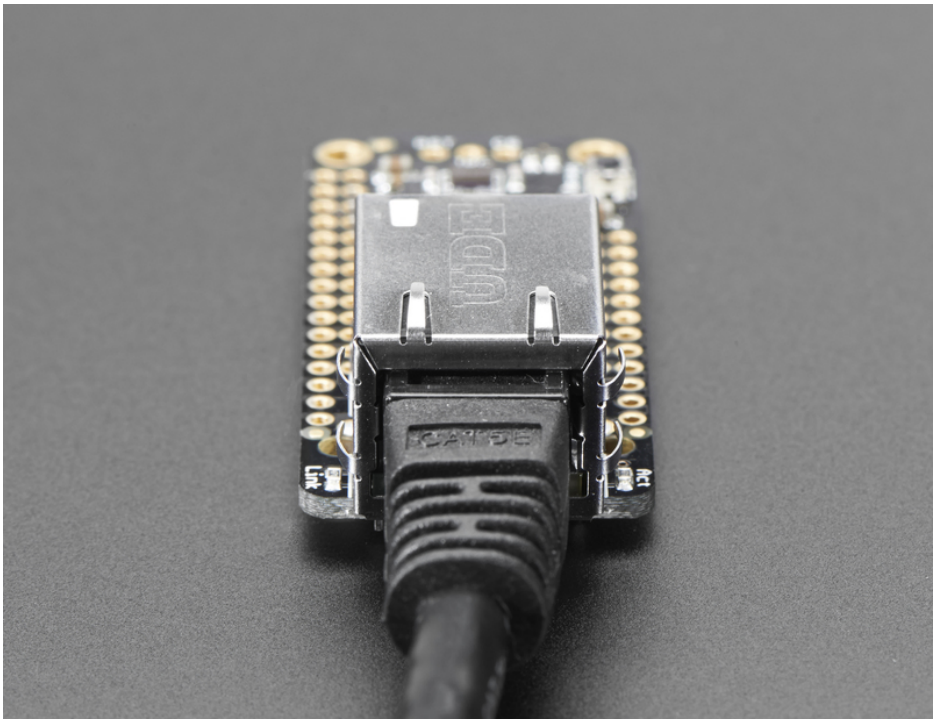


Ethernet is a tried-and-true networking standard. It's supported by every hub and switch, and because there's a physical connection you don't have to noodle around with SSIDs, passwords, authentication schemes or antennas. It

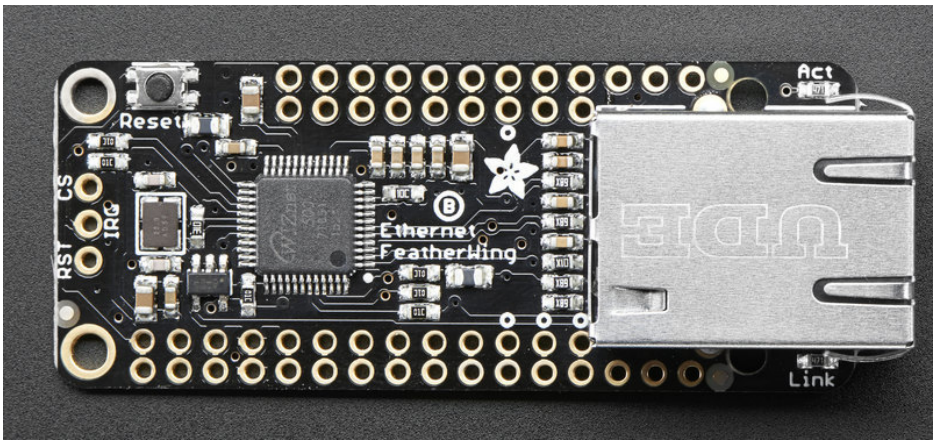
works great with any of our Feathers, the WIZ5500 chip communicates over SPI plus a single CS pin. The Arduino Ethernet2 library works great, and within a few seconds after connecting, will do the DHCP setup for you. As a nice extra, the RJ-45 jack has both link and activity lights that will light/blink to let you know the current connection status.



Each order comes with one assembled and tested FeatherWing, plus some header. You will need to solder in the header yourself but its a quick task.

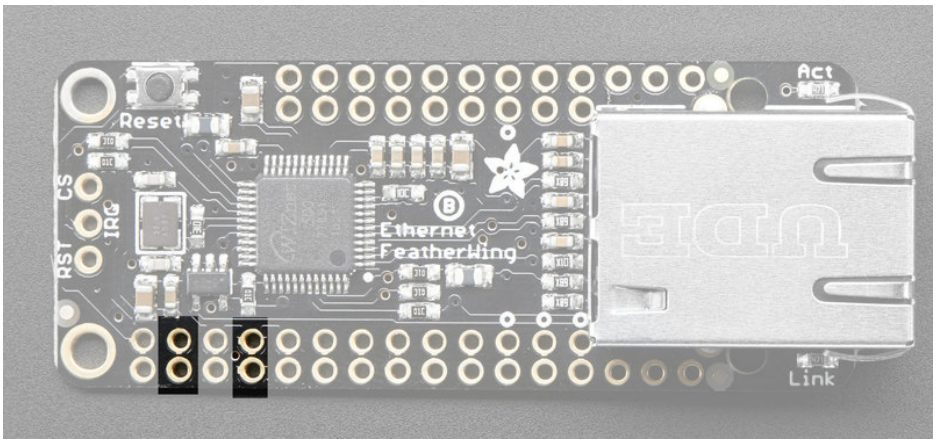


Pinouts



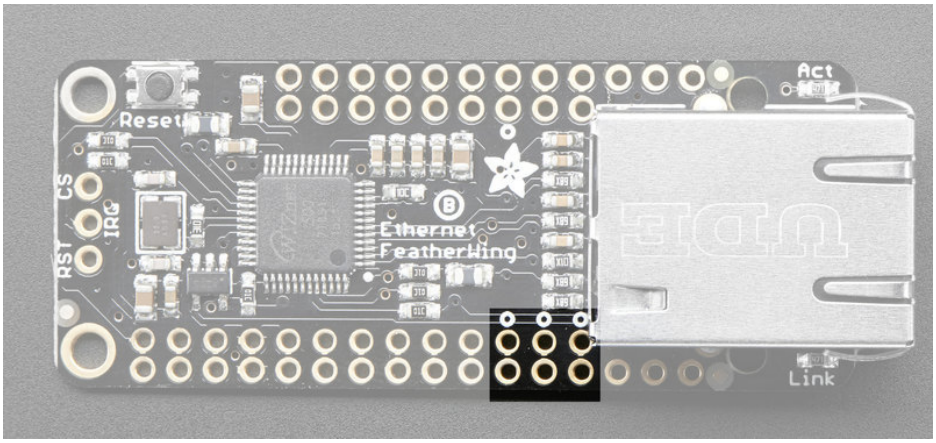
The Ethernet FeatherWing plugs into any Feather and adds hard-wired networking. To make it as cross-platform compatible as possible, we use only SPI plus one extra, configurable pin. It's not going to be gigabit (or even multi-megabit) speeds but it will work!

Power Pins



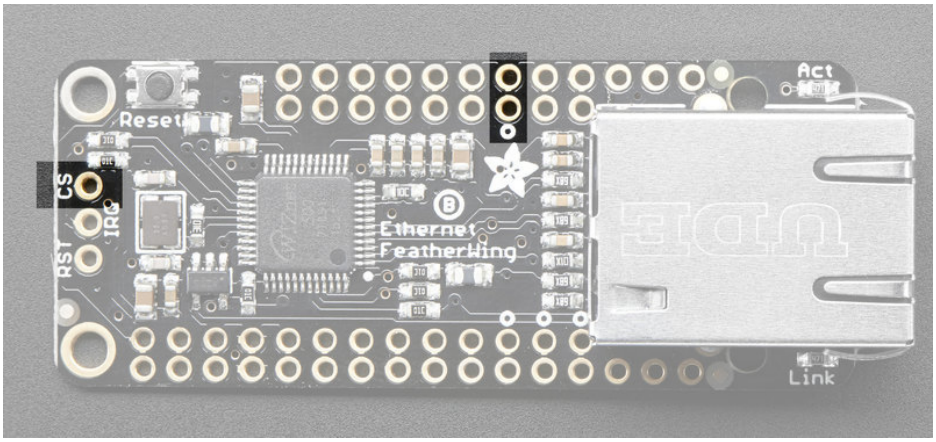
The WIZ5500 Ethernet chip (and really any ethernet chip) are fairly high current devices, requring about 150mA whether or not actively transmitting. The chip is powered from the **3.3V** and **GND** pins indicated and does not have an internal regulator.

SPI Data Pins



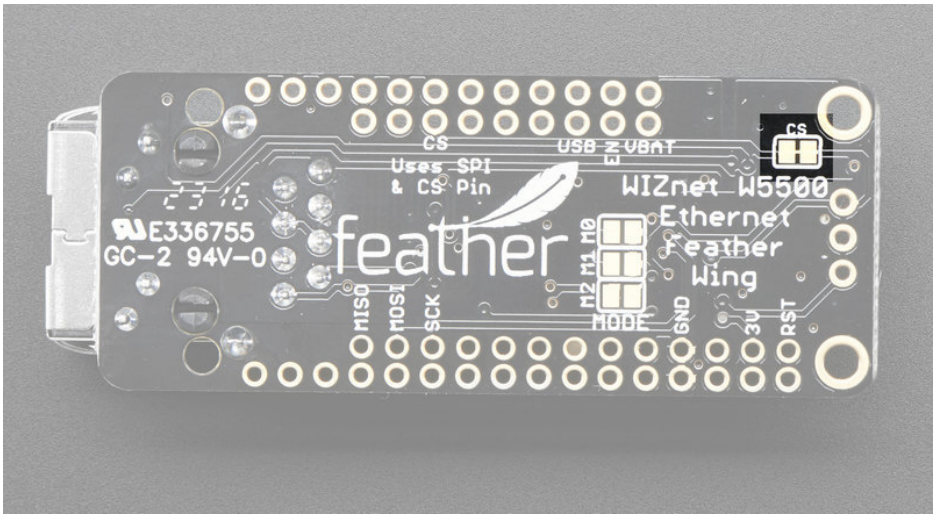
Configuration, setup, and data is received/transmitted over SPI. All Feathers have the default SPI port on the same pins, so we use **MOSI/MISO/SCK** in this location. They are 3.3V logic, 5V tolerant.

A Chip Select pin is also required. By default we use this pin:

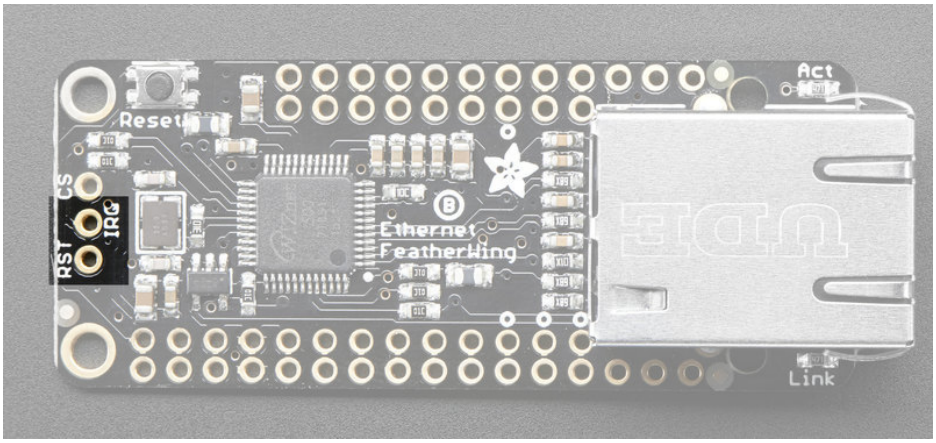


Which is connected to pin **#10** on Feather Teensy 3, 32u4, 328p, M4 and M0, pin **#15** on ESP8266, and **PB4** on WICED.

You can also get to the CS pin from the left group of breakouts. To change CS pin, solder this pad to a different pin and cut this jumper:

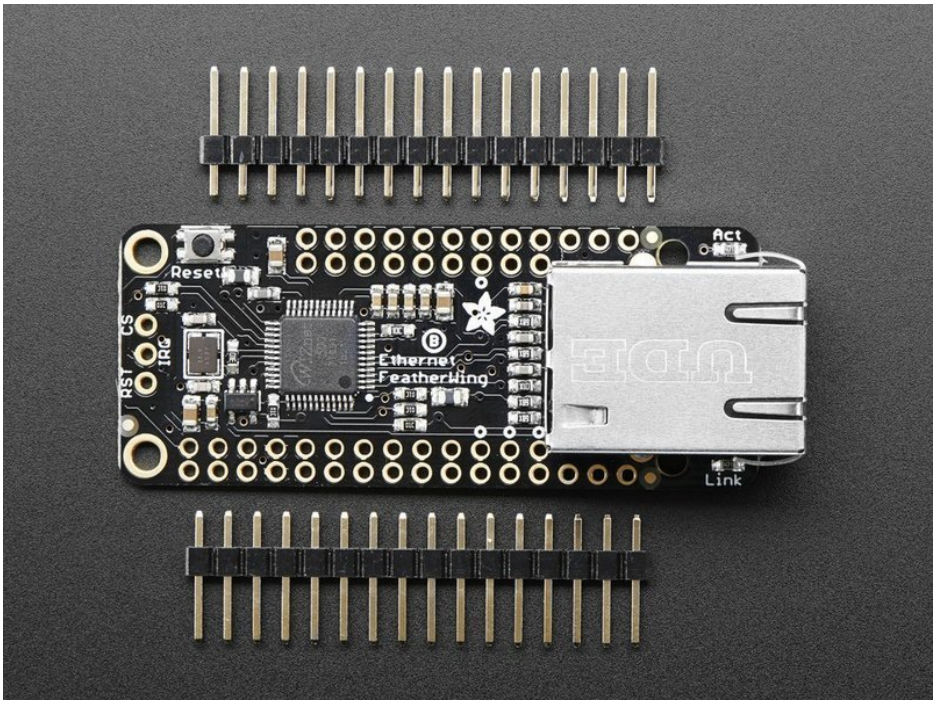


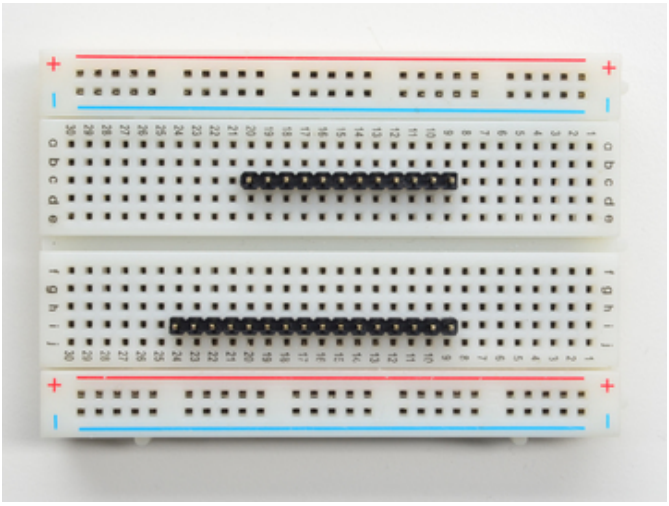
Other Breakouts



You can also connect to and use the **IRQ** and WIZ **ReSeT** pins. These are not used in the library and are there for advanced usage. They are 3.3V logic, 5V tolerant.

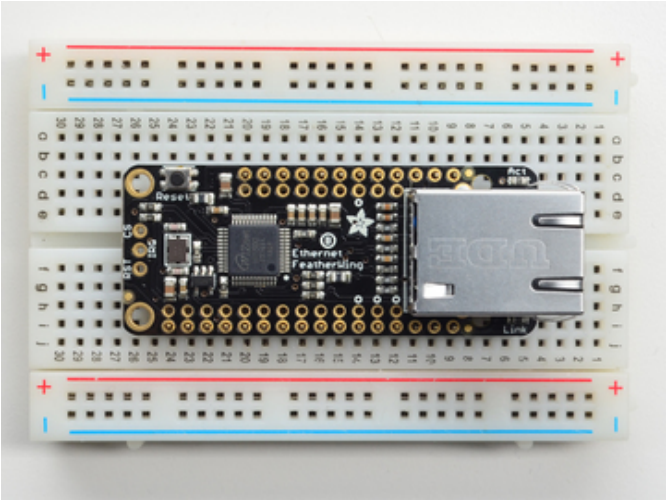
Assembly





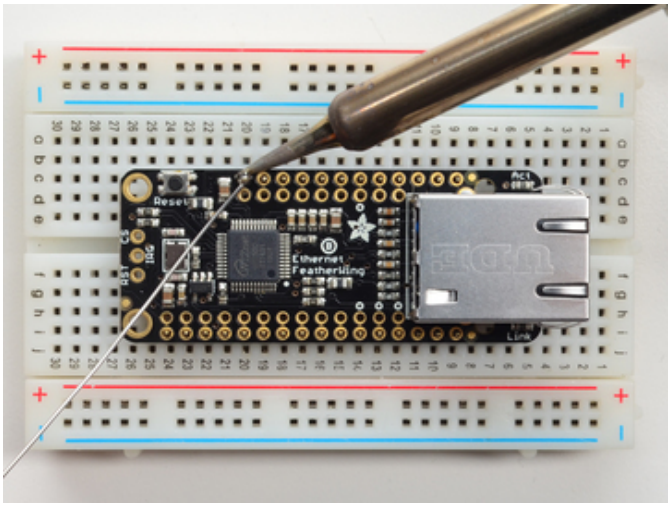
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



Add the FeatherWing:

Place the featherwing over the pins so that the short pins poke through the two rows of breakout pads

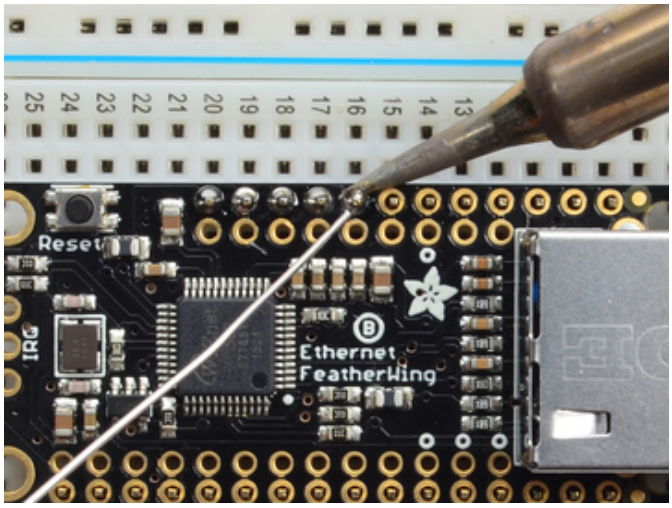


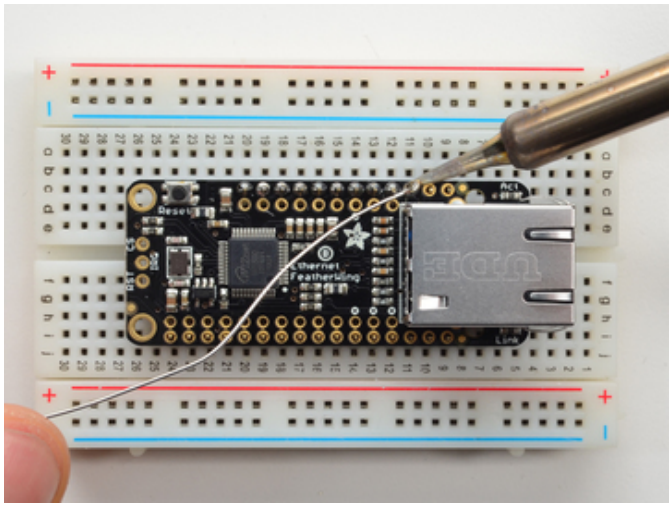
And Solder!

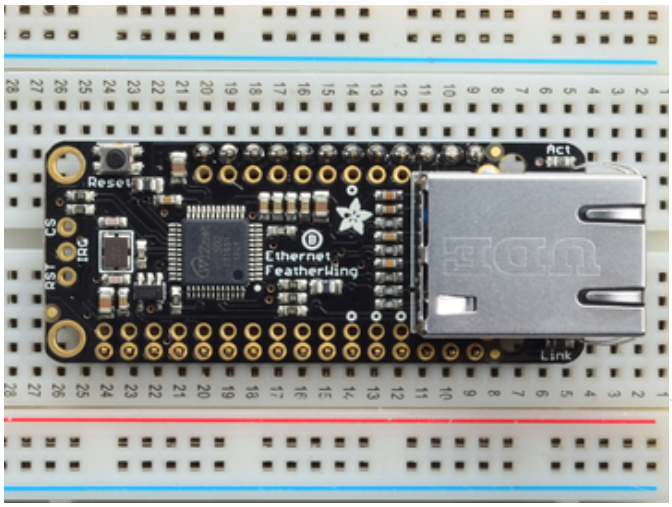
Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering (https://adafru.it/aTk).

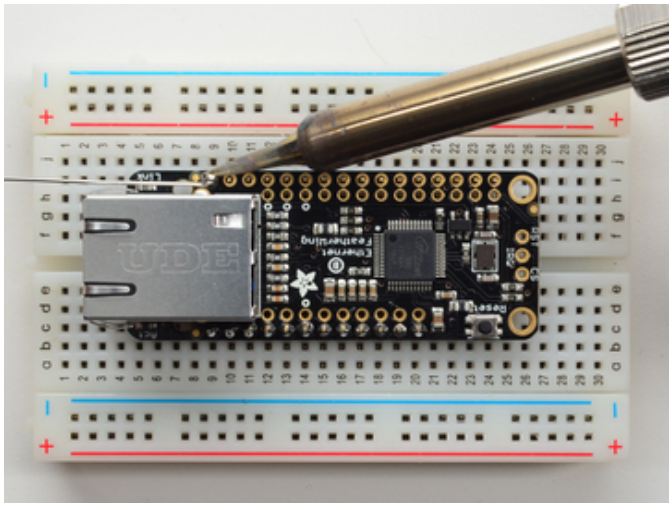
Start by soldering the first row of header

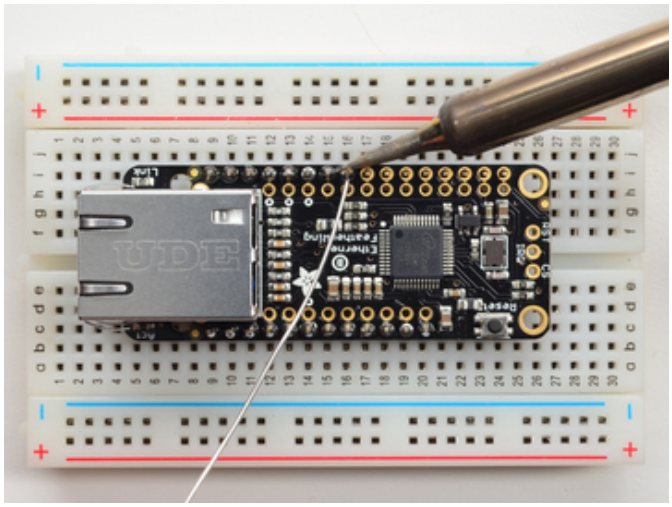


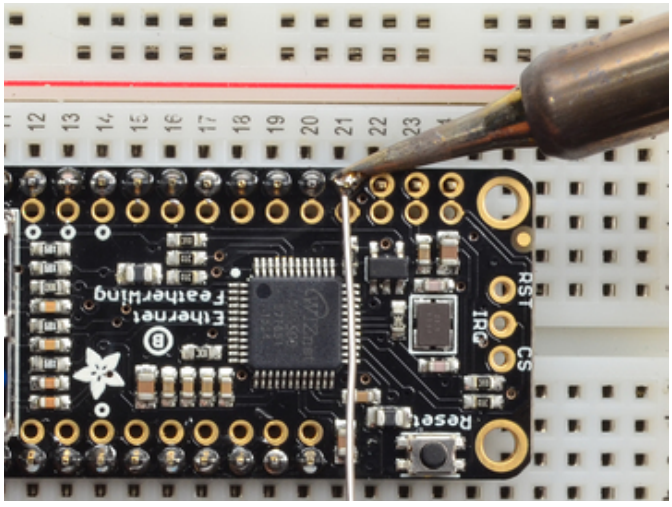


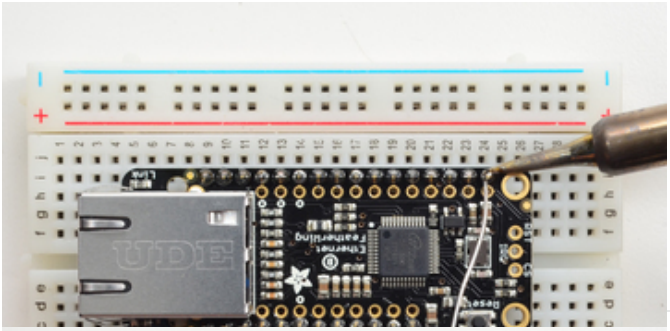


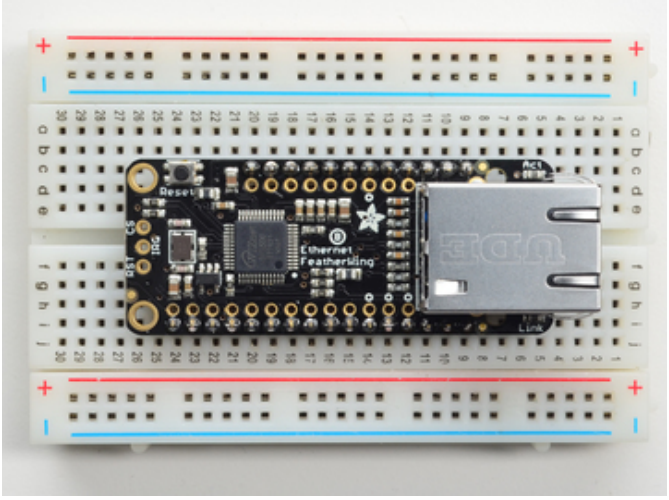
Now flip around and solder the other row completely



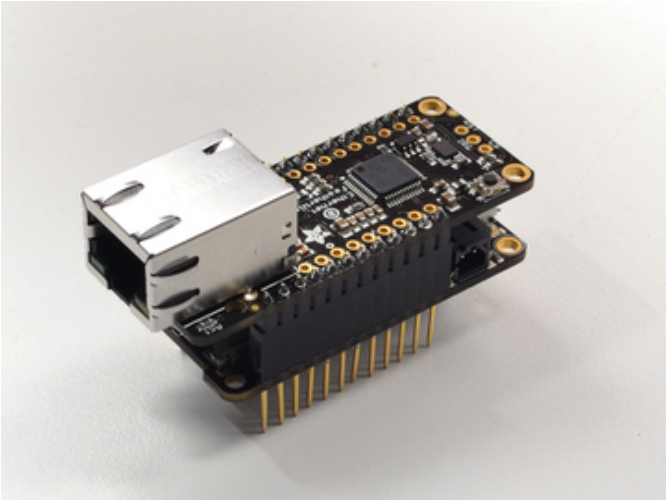


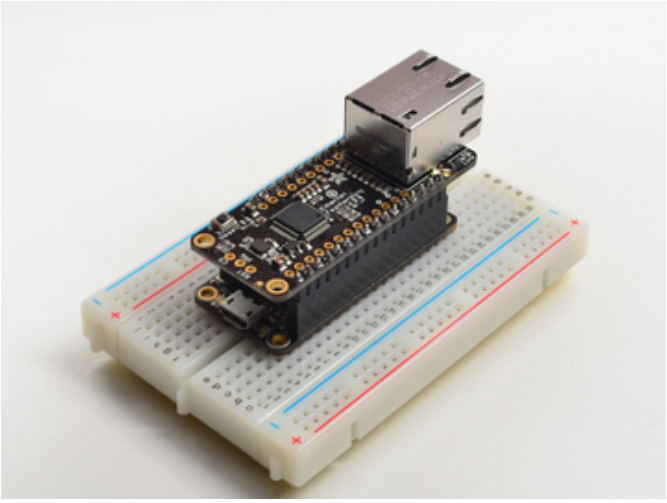

Ethernet
FeatherWing
RST
IRQ
CS
Reset





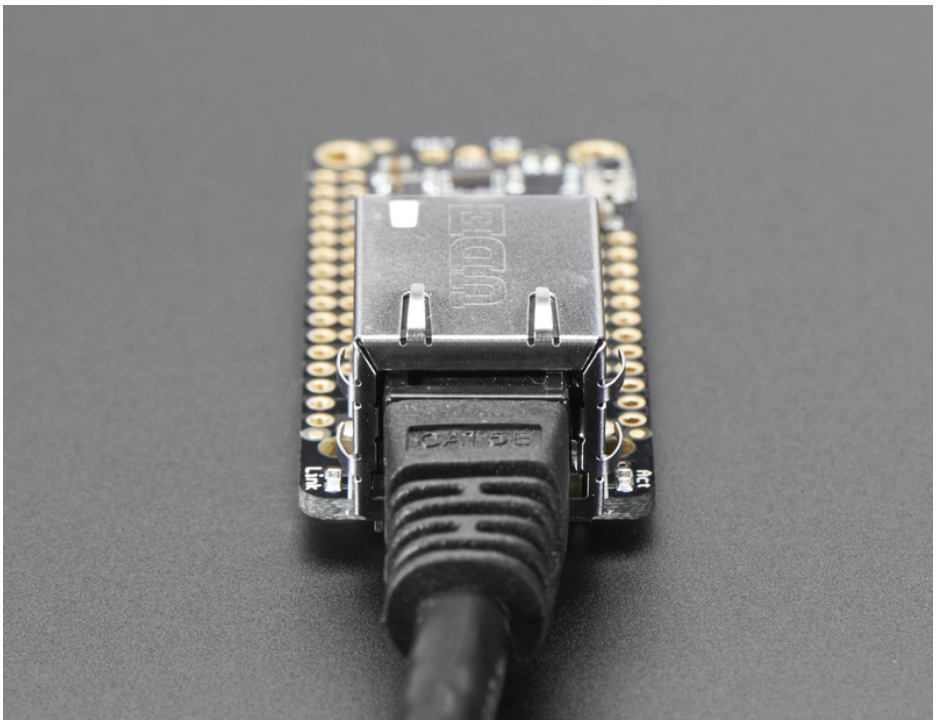
You're done !





Attach an ethernet cable, & plug it into your feather of choice!

Usage



The Ethernet chipset we are using (Wiz5500) is well supported and works for all Feathers, all you need is a little library support and you will be sending packets in no time!

Install Ethernet2 Library

Start by installing the support library for the Wiz5500. We have the Adafruit fork of the Ethernet2 library repository on GitHub (https://adafru.it/plf) if you're interested in looking at the code.

Start by downloading the library. You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip:

https://adafru.it/plA

https://adafru.it/plA

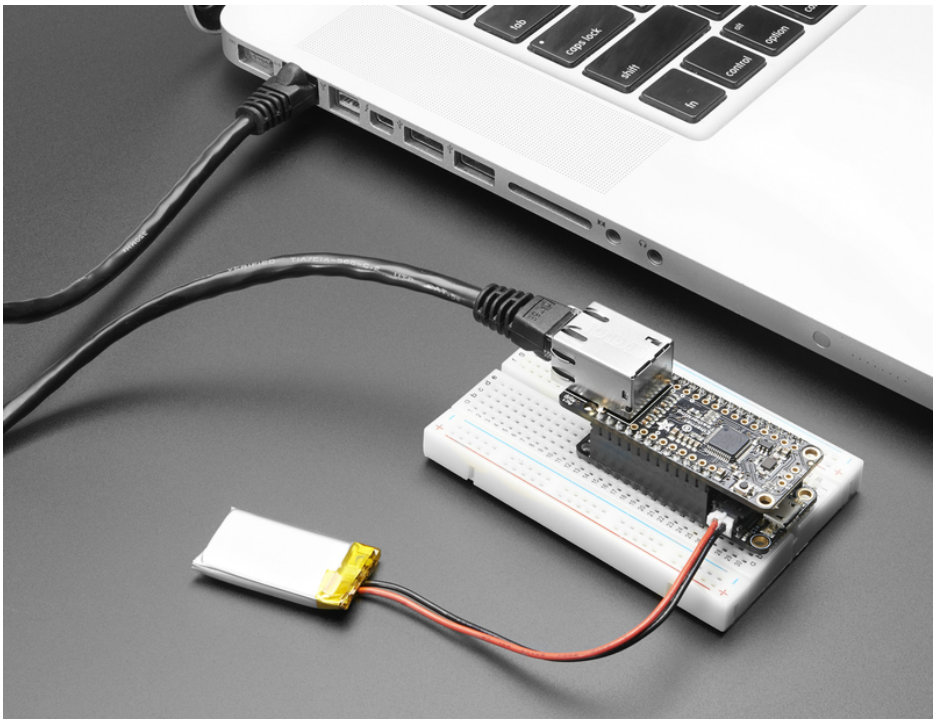
Rename the uncompressed folder **Ethernet2** and check that the **Ethernet2** folder contains **src**, **keywords.txt** and **library.properties**

Place the **Ethernet2** library folder your ***arduinosketchfolder*/libraries/** folder.
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (https://adafru.it/aYM)

Connect Cable

Make sure you have the FeatherWing plugged into your Feather, and an Ethernet cable connected to your hub, router or switch.



Run Example Code

We have a basic demo that works with all Feathers, so restart the IDE and compile/upload this sketch

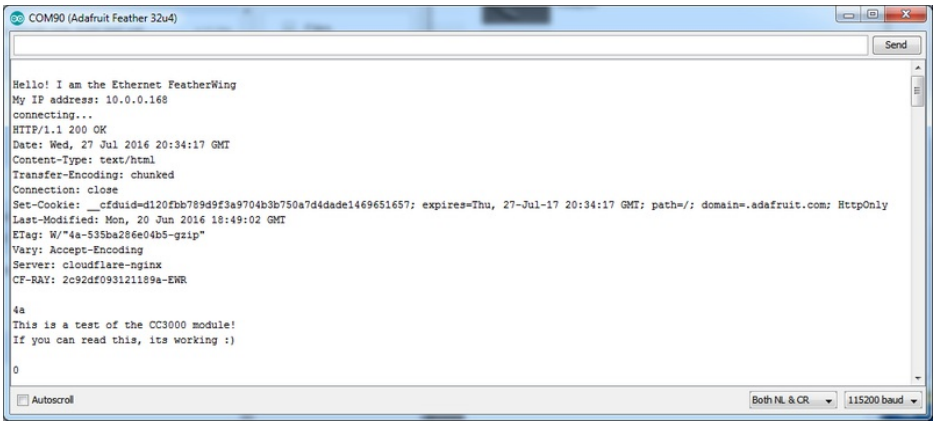
```
/*
  Web client

 This sketch connects to a website (http://www.adafruit.com)
 using an Arduino Wiznet Ethernet shield/FeatherWing.

 created 18 Dec 2009
 by David A. Mellis
 modified 9 Apr 2012
 by Tom Igoe, based on work by Adrian McEwen

 */

#include <SPI.h>
#include <Ethernet2.h>

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
//IPAddress server(74,125,232,128);  // numeric IP for Google (no DNS)
#define URLHOST "wifitest.adafruit.com"    // name address for Adafruit (using DNS)
#define URLPATH "/testwifi/index.html" // url to grab

// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 0, 177);

// Initialize the Ethernet client library
// with the IP address and port of the server
```

```
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;

unsigned long lastConnectionTime = 0;             // last time you connected to the server, in millisecon
const unsigned long postingInterval = 10L * 1000L; // delay between updates, in milliseconds
// the "L" is needed to use long type numbers

//#define WIZ_RESET 9

#if defined(ESP8266)
  // default for ESPressif
  #define WIZ_CS 15
#elif defined(ESP32)
  #define WIZ_CS 33
#elif defined(ARDUINO_STM32_FEATHER)
  // default for WICED
  #define WIZ_CS PB4
#elif defined(TEENSYDUINO)
  #define WIZ_CS 10
#elif defined(ARDUINO_FEATHER52832)
  #define WIZ_CS 11
#else   // default for 328p, 32u4, M4 and M0
  #define WIZ_CS 10
#endif


void setup() {
#if defined(WIZ_RESET)
  pinMode(WIZ_RESET, OUTPUT);
  digitalWrite(WIZ_RESET, HIGH);
  delay(100);
  digitalWrite(WIZ_RESET, LOW);
  delay(100);
  digitalWrite(WIZ_RESET, HIGH);
#endif

#if !defined(ESP8266)
  while (!Serial); // wait for serial port to connect.
#endif

  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  delay(1000);
  Serial.println("\nHello! I am the Ethernet FeatherWing");

  Ethernet.init(WIZ_CS);

  // give the ethernet module time to boot up:
  delay(1000);

  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to congifure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }

  // print the Ethernet board/shield's IP address:
```

```
  Serial.print("My IP address: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // if there's incoming data from the net connection.
  // send it out the serial port.  This is for debugging
  // purposes only:
  if (client.available()) {
    char c = client.read();
    Serial.write(c);
  }

  // if ten seconds have passed since your last connection,
  // then connect again and send data:
  if (millis() - lastConnectionTime > postingInterval) {
    httpRequest();
  }

}

// this method makes a HTTP connection to the server:
void httpRequest() {
  // close any connection before send a new request.
  // This will free the socket on the WiFi shield
  client.stop();

  // if there's a successful connection:
  if (client.connect(URLHOST, 80)) {
    Serial.println("connecting...");
    // send the HTTP PUT request:
    client.println("GET " URLPATH " HTTP/1.1");
    client.println("Host: " URLHOST);
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();

    // note the time that the connection was made:
    lastConnectionTime = millis();
  }
  else {
    // if you couldn't make a connection:
    Serial.println("connection failed");
  }
}
```

If you're using a LoRa/RFM69 Feather, make sure to add a pinMode(8, INPUT_PULLUP) to disable the LoRa or RFM69 radio

Then open up the serial console of your Feather to see the connection and data:



If you don't get an IP address, check you have a green link light, and that your Ethernet is going out to an internet connected router. You may also have to set up the MAC address to allow it access, check with your system admin if you're not sure.

Library Reference

Ethernet2 is fully compatible with the Client (https://adafru.it/lFj) interface and also back compatible with the Arduino Ethernet library so you can use that documentation (https://adafru.it/lGA) when coding. Just include **Ethernet2.h** instead of Ethernet and **init()** with the chip select pin so it knows what pins to use.

Power over Ethernet

To keep the FeatherWing small, simple and inexpensive, we do not include PoE support. However, it's a common request that some people want to simplify cabling when the device is far away.

This all-in-one solution will let you plug an Ethernet cable to a PoE switch/hub then split into data and a Feather-friendly micro USB connector on the other

PoE Splitter with MicroUSB Plug - Isolated 12W - 5V 2.4 Amp

$10.95
IN STOCK

ADD TO CART

Passive PoE

If you don't even have a PoE hub, here's a solution that doesn't even require a PoE hub: passive PoE injectors! (http://adafru.it/435)

Passive PoE Injector Cable Set

$5.95
IN STOCK

ADD TO CART

These cables don't have any active PoE negotation circuitry, intstead, we just use the extra unused wires in the Cat-5+ wires to carry power and ground. They're a simple and afforable solution.

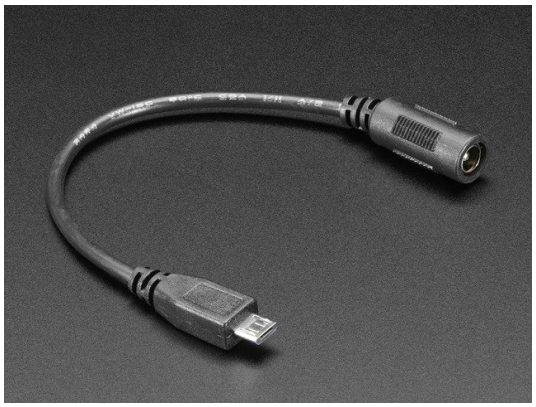
On one end, plug in the Ethernet cable to your router, and a 5V wall adapter into the 2.1mm jack (http://adafru.it/276)

5V 2A (2000mA) switching power supply - UL Listed

$7.95
IN STOCK

ADD TO CART

Then run any standard Ethernet cable to the other end where you will use a 2.1mm to microUSB adapter cable (http://adafru.it/2727)



MicroUSB Plug to 5.5/2.1mm DC Barrel Jack Adapter

$1.95
IN STOCK

ADD TO CART

And plug that directly into the Feather. For long power runs there might be a slight voltage droop but since the regulator on board will reduce the input voltage to 3.3V for the chip and peripherals (all Feathers are 3.3V logic) it should not be noticable

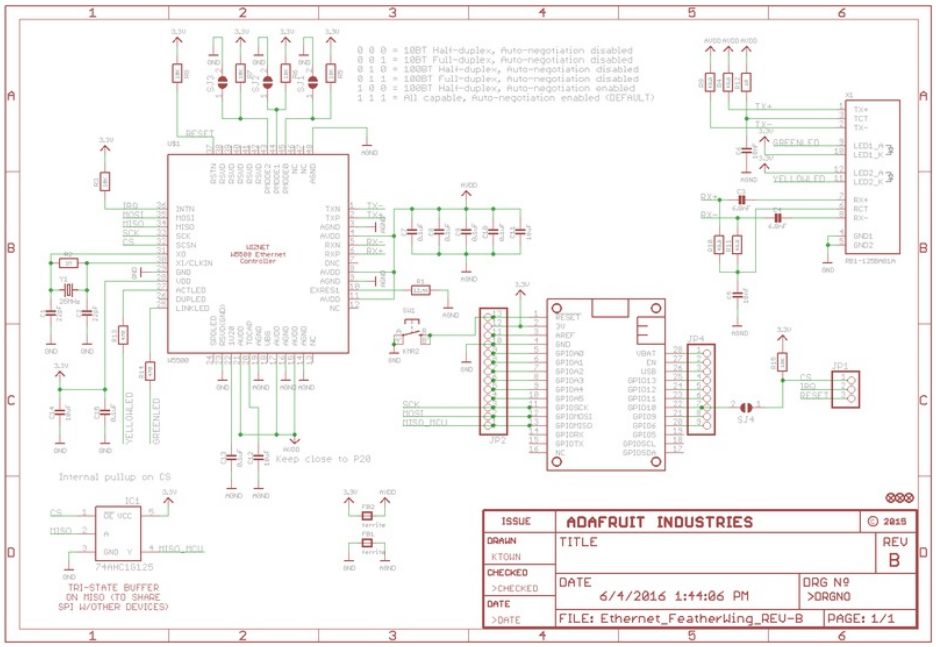
Downloads

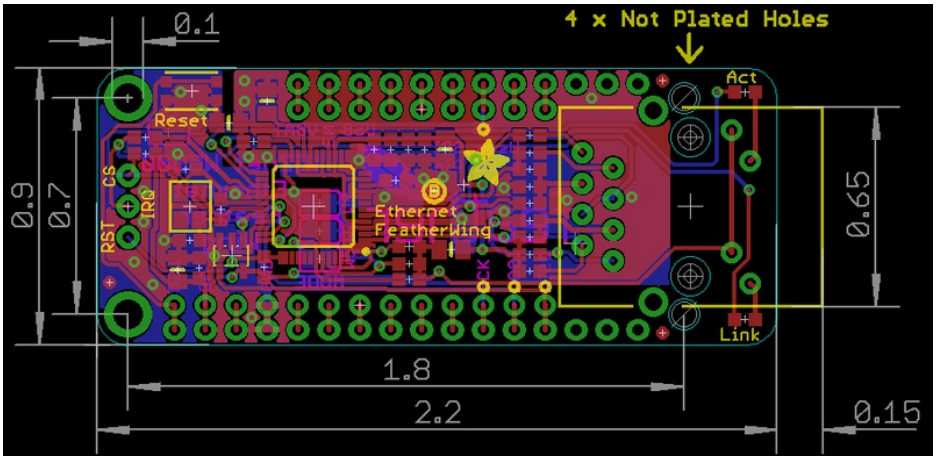
Files & Datasheets

- WIZ5500 Datasheet (https://adafru.it/pID)
- EagleCAD PCB files on GitHub (https://adafru.it/pIE)
- Fritzing object in Adafruit Fritzing repo (https://adafru.it/aP3)

Schematic

Click to embiggen



Fabrication Print

dimensions in inches