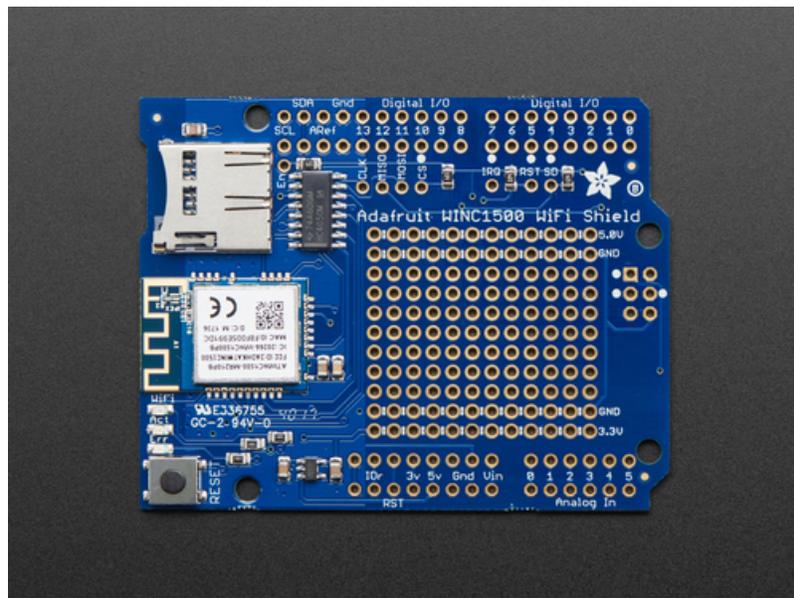


Adafruit WINC1500 WiFi Shield for Arduino

Created by lady ada



Last updated on 2018-11-11 10:17:45 PM UTC

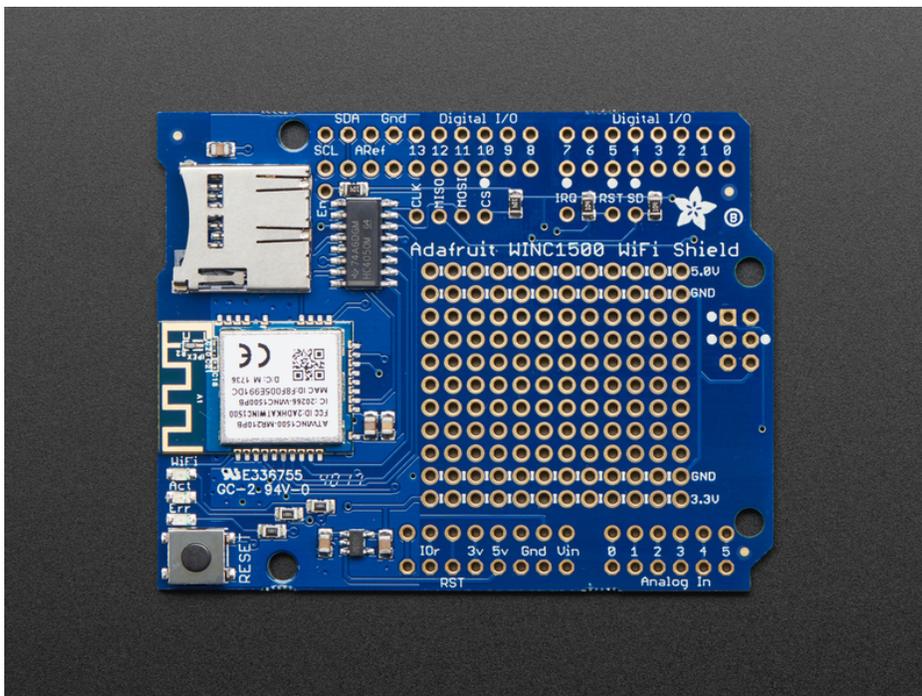
Guide Contents

Guide Contents	2
Overview	3
Pinouts	6
SPI Interface Pins	6
WiFi Control Pins	7
SD Card Interface	7
LEDs	8
Prototyping Area	8
Assembly	10
Installing Standard Headers	10
Stack Alert!	15
Wiring & Test	21
Install the Library	21
Check Connections & Version	21
Scanning WiFi	23
Connect & Read Webpage	23
Changing the control pins?	24
Updating Firmware	26
Updating SSL Certificates	31
What SSL/TLS support is available with the WINC1500?	34
Downloads	35
Datasheets & Files	35

Overview



Connect your Arduino-compatible to the Internet with this WiFi shield that features the FCC-certified ATWINC1500 module from Atmel. This 802.11bgn-capable WiFi module is the best new thing for networking your devices, with SSL support and rock solid performance - running our adafruit.io MQTT demo for a full weekend straight with no hiccups (it would have run longer but we had to go to work, so we unplugged it).

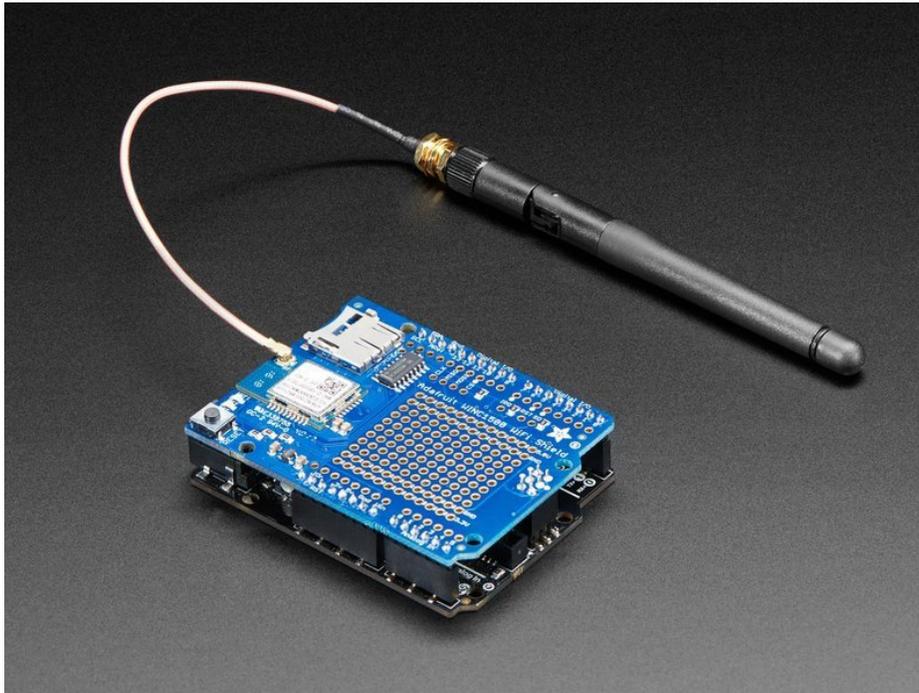


The **Adafruit ATWINC1500 WiFi Shield** uses SPI to communicate plus some GPIO for control, so with about 6 wires, you can get your wired up and ready to go. Right now the Atmel-supplied library works best with SAMD21-based

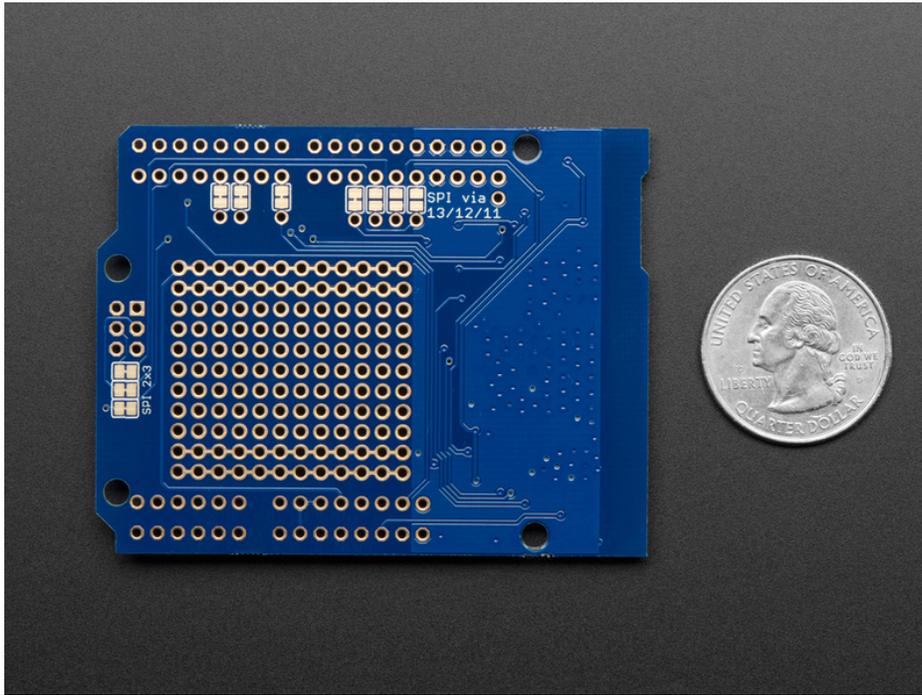
boards like the Arduino Zero or [Metro M0 Express \(https://adafru.it/xoa\)](https://adafru.it/xoa), or Megas or M4s. It will not work/fit on other Arduinos such as 328P or 32u4-based or attiny-based boards. You can clock it as fast as 12MHz for speedy, reliable packet streaming. And scanning/connecting to networks is very fast, a few seconds.

This module works with 802.11b, g, or n networks & supports WEP, WPA and WPA2 encryption. You can use it in Soft AP mode to create an ad-hoc network. For secure client connections, there is TLS 1.2 support!

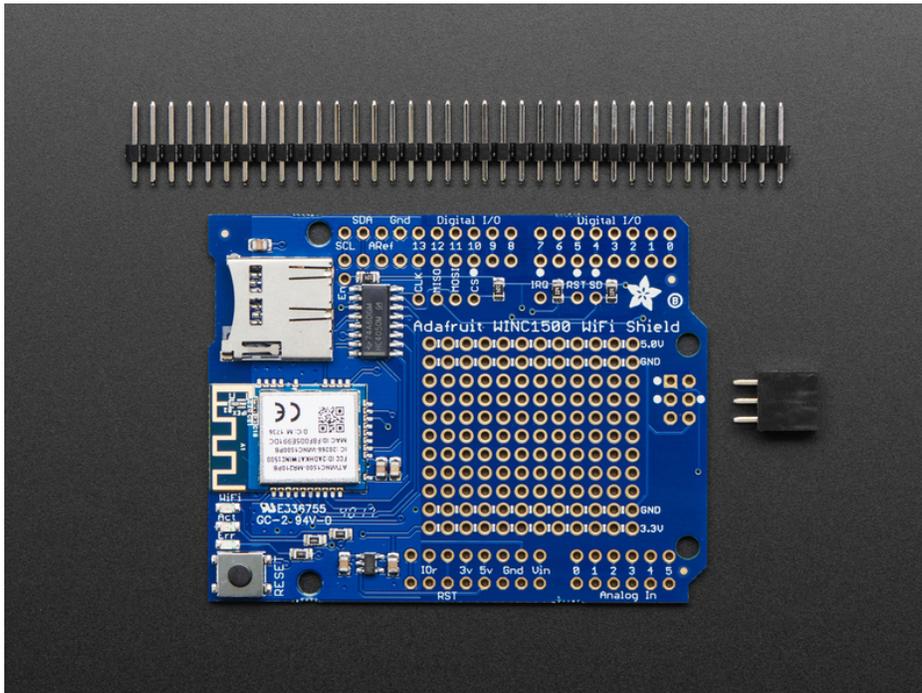
We have two versions. [One version comes with a built-in PCB trace antenna \(https://adafru.it/BtL\)](https://adafru.it/BtL). We also have a [uFL version for use with external antennas \(http://adafru.it/3654\)](http://adafru.it/3654)



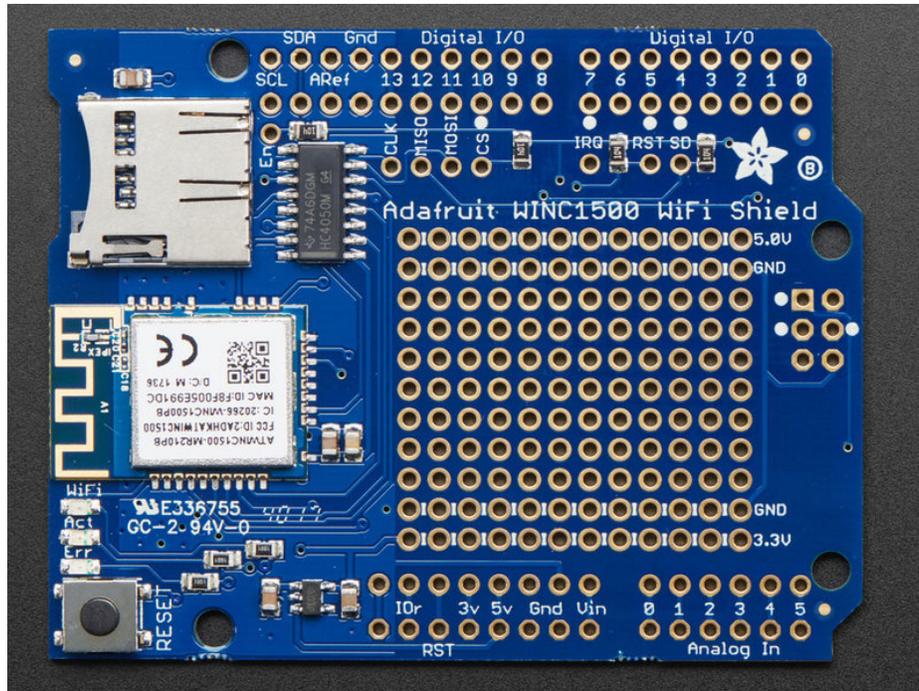
These SPI-protocol WiFi module are currently the best way to add WiFi to your Arduino-compatible so we've decided to make a handy shield for it. The shield comes with level shifting on all the input pins so you can use it with 3V or 5V logic boards. A 3.3V voltage regulator that can handle the 300mA spikes lets you power from 3-5.5VDC. There's also 3 LEDs that you can control over the SPI interface (part of the library code) or you can have controlled by the Arduino library. They'll light up when connected to an SSID, or transmitting data.



We also tossed on a micro SD card socket, you can use that to host or store data you get from the Internet. Arduino's based on the ATmega328 (like the UNO) cannot use both the WiFi module and SD library at the same time, they're too small. Upgrade to a [Metro M0 Express](https://adafru.it/xoa) to get a big memory bump!

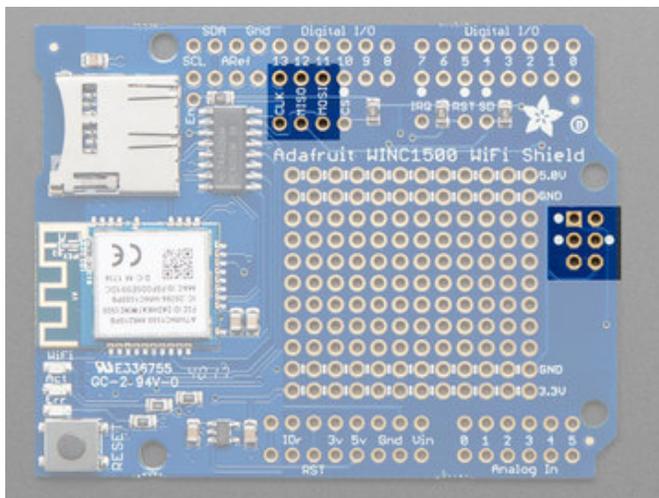


Pinouts



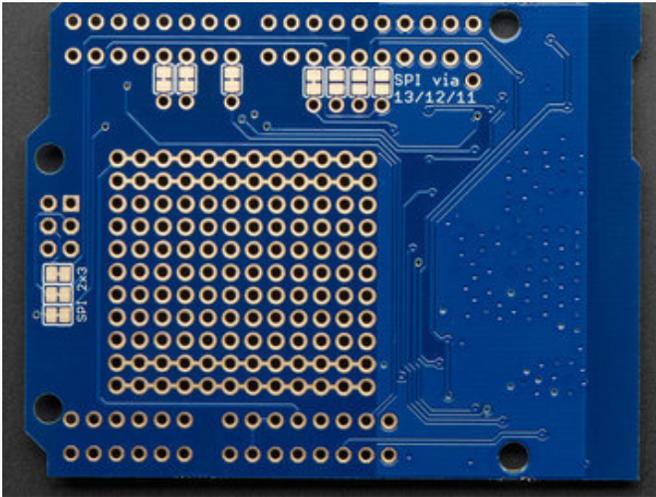
There's a lot jam-packed into this shield! Lets take a look at what we've got going on.

SPI Interface Pins



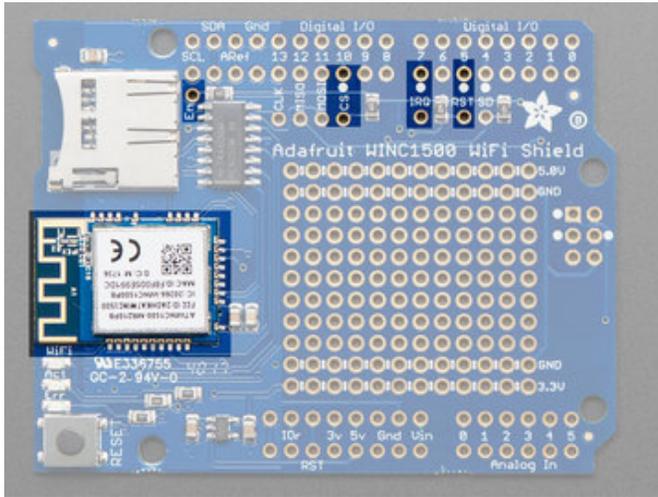
Both WiFi and SD card use SPI to send and receive data. These pins are labeled **CLK MISO MOSI** and have level shifting so you can use this shield with 3.3V or 5V microcontroller boards.

By default the 2x3 pin **ICSP** header on the right hand side is where the SPI signals are found. However, on some older or custom Arduinos that have Atmega328P processors, the SPI pins are found on digital 11, 12, and 13



If you happen to want to use 11/12/13 rather than the SPI header, turn the board over, you can cut the 3 jumpers next to the 2-3 ISP header and solder the open jumpers between the SPI pins and D11/12/13

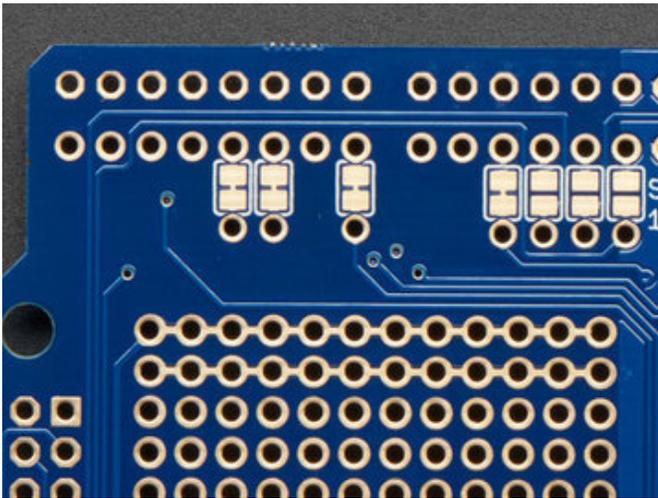
WiFi Control Pins



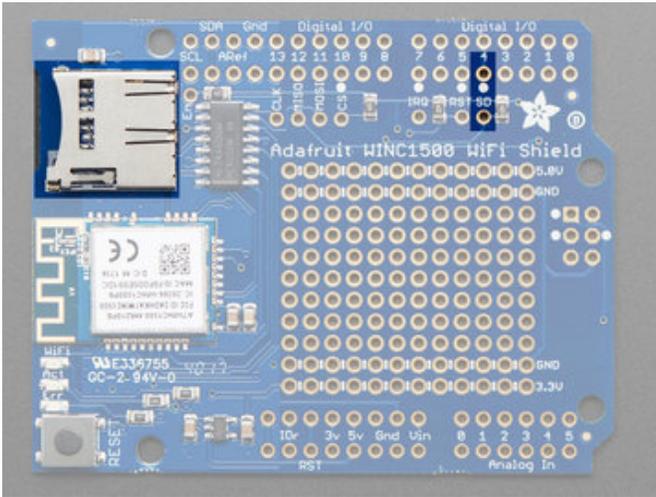
The WiFi module is where the magic happens. In addition to the SPI pins, the WINC module also requires three GPIO for control:

- **CS** chip select - can be any pin
- **IRQ** interrupt out - *can be any pin despite being called an IRQ pin*
- **RST** hardware reset - can be any pin
- **EN** - this is not a required pin, but can be used to completely disable power to the WINC for ultra-power-savings!

If you need to change the pins, cut the jumpers underneath the shield and rewire the pins.



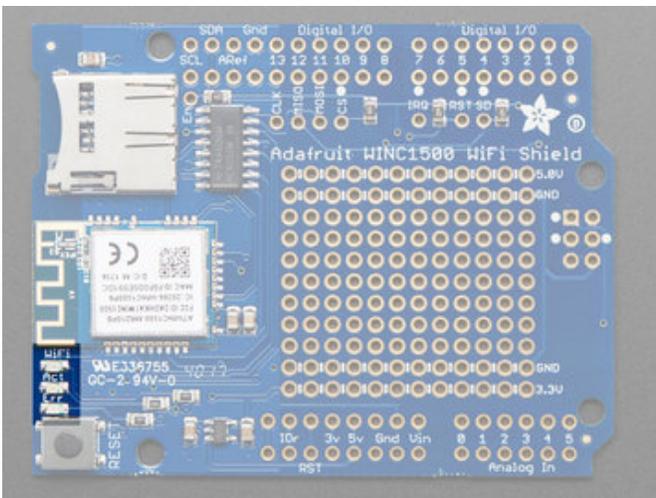
SD Card Interface



There's a lot of space available on this shield so we also stuck on a micro SD card holder, great for datalogging or storing data to transmit over WiFi.

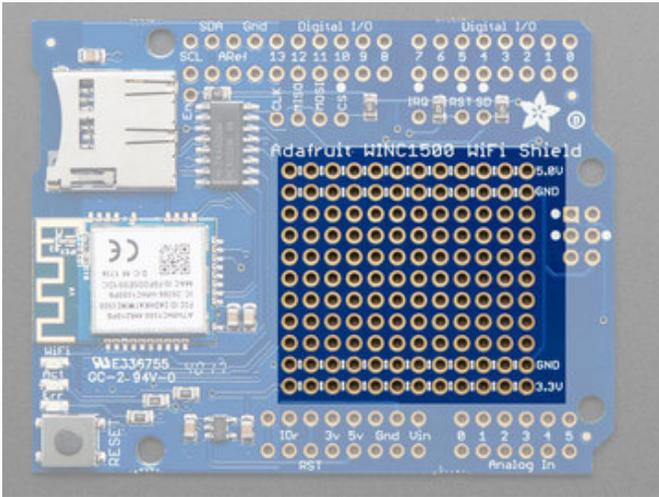
In addition to the shared SPI pins, the **SDCS** (chip select) pin is also used. It can be re-assigned to any pin by cutting the trace underneath the board and rewiring. If the SD card is not used, the **SDCS** pin can be used for any other purpose

LEDs

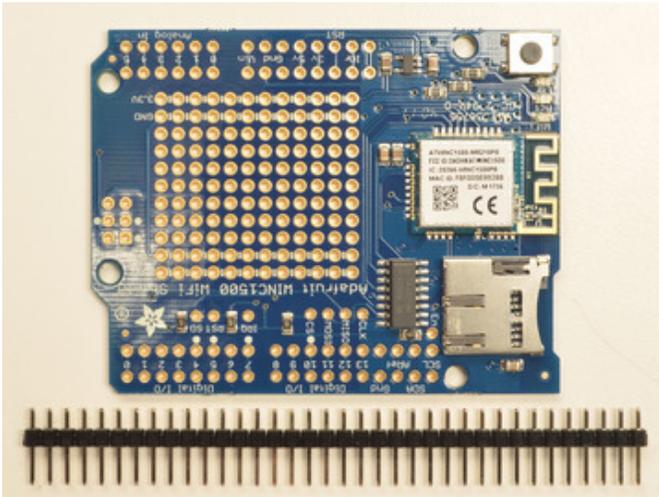


There are 3 LEDs that are controlled by the Arduino library - they will let you know when there is **WiFi** access-point connectivity, **Activity** or an **Error** when setting up.

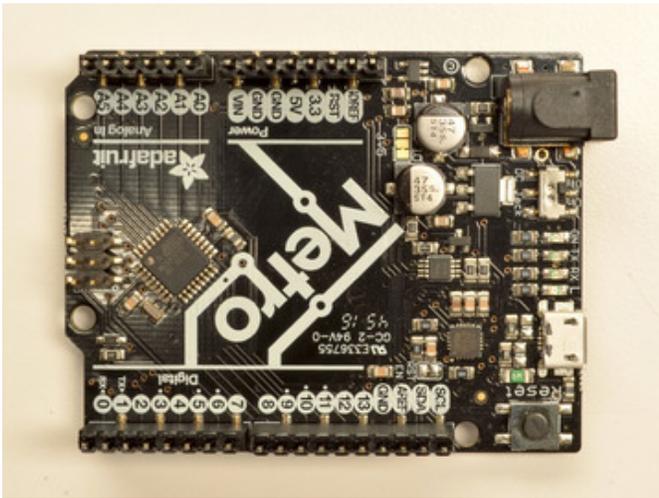
Prototyping Area

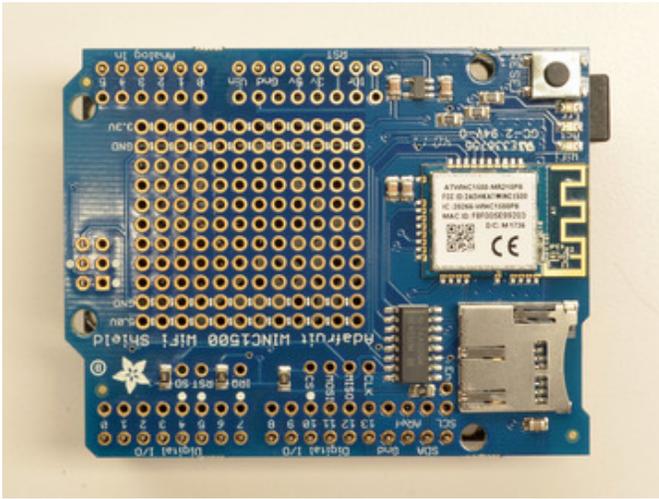


We have a big grid of prototyping holes and power rails if you want to make some custom circuitry!

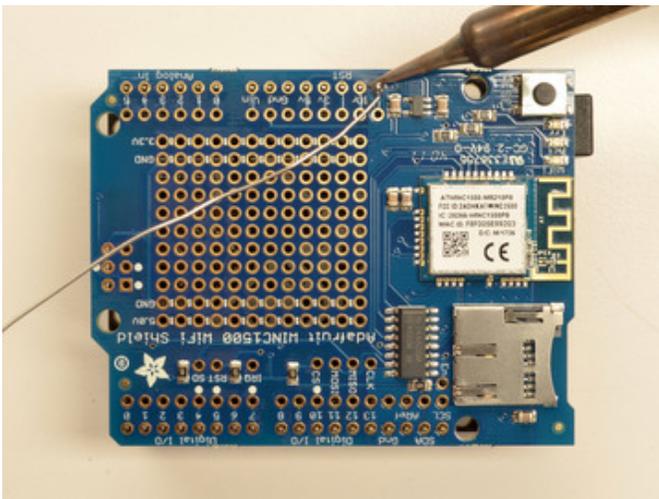


Break apart the 0.1" header into 6, 8 and/or 10-pin long pieces and slip the long ends into the headers of your Arduino

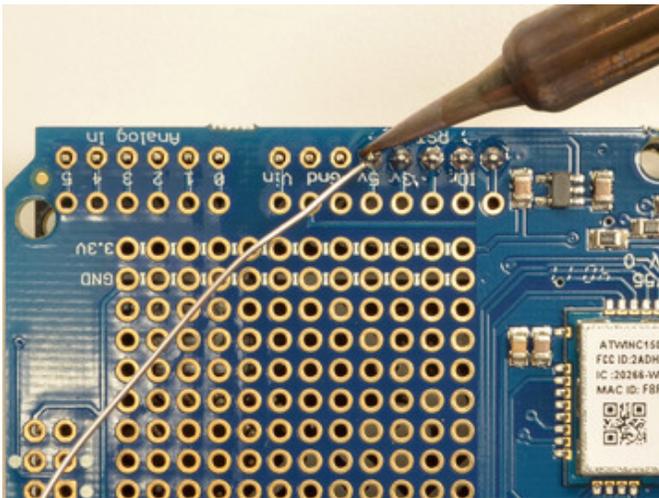


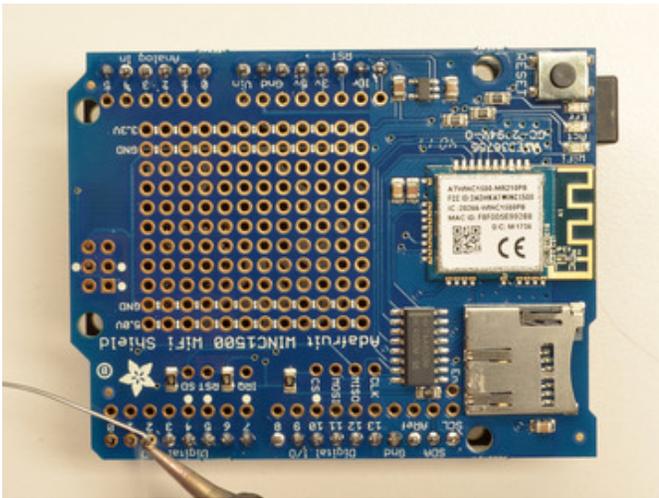
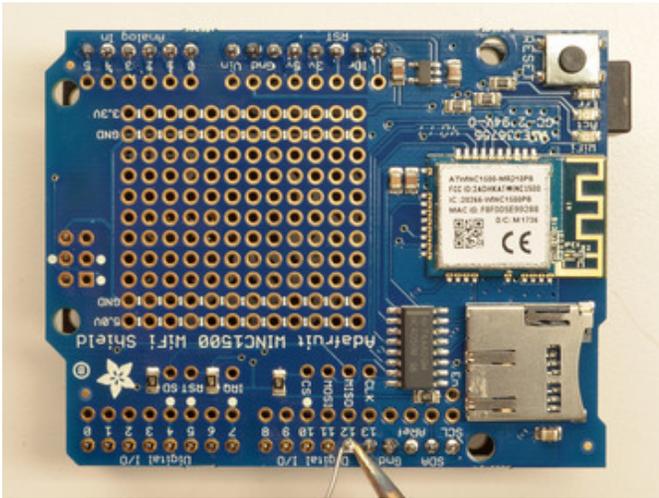
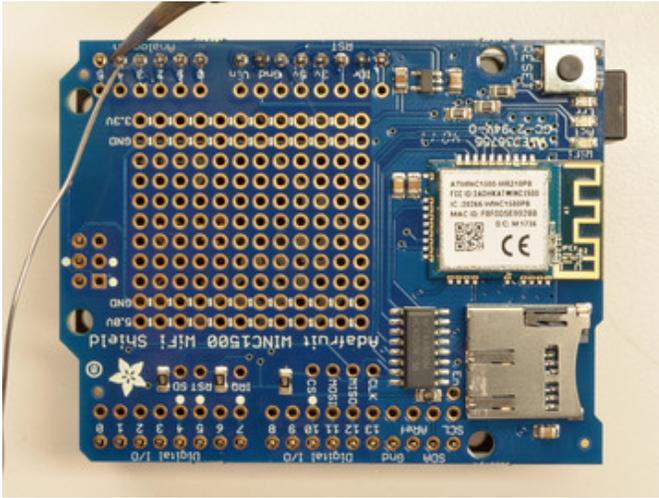


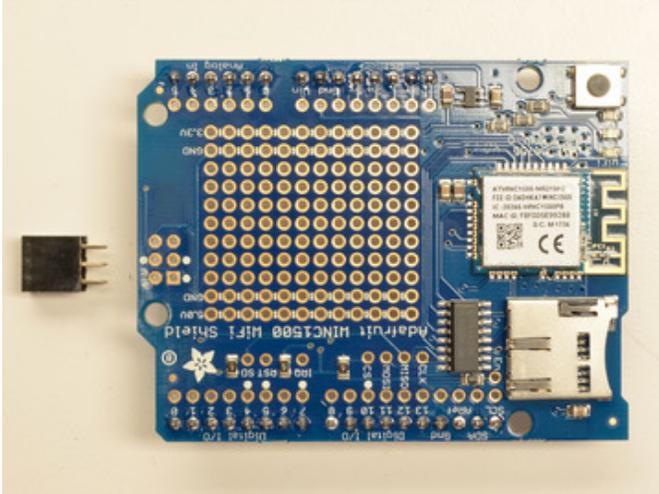
Place the assembled shield on top of the header-ed Arduino so that all of the short parts of the header are sticking through the outer set of pads



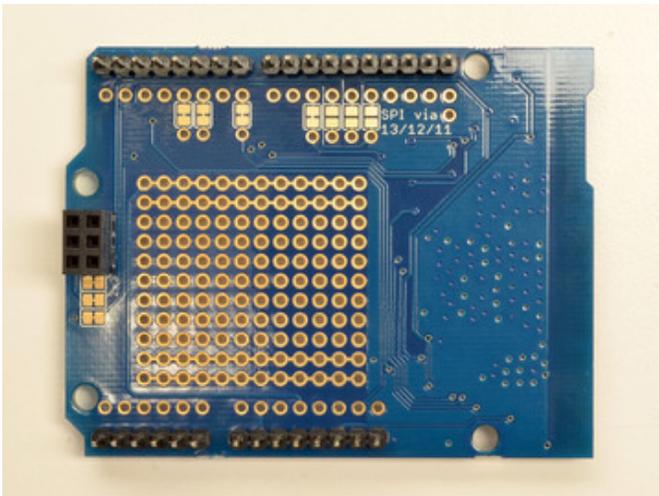
Solder each one of the pins into the shield to make a secure connection



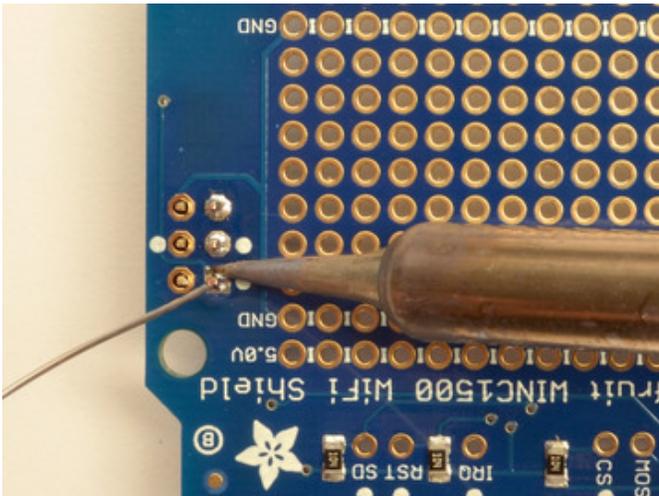
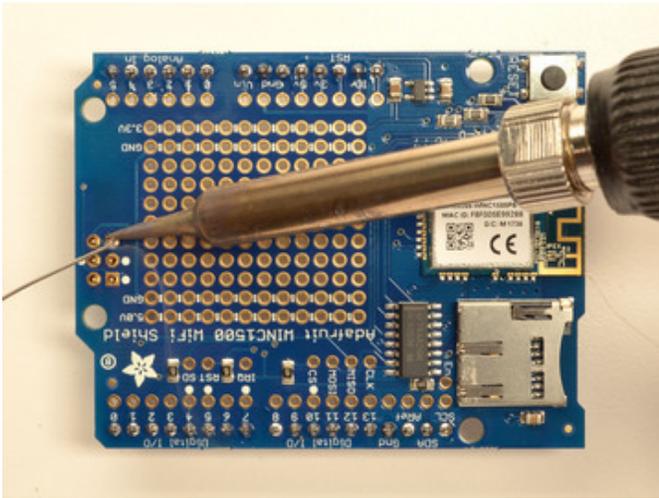
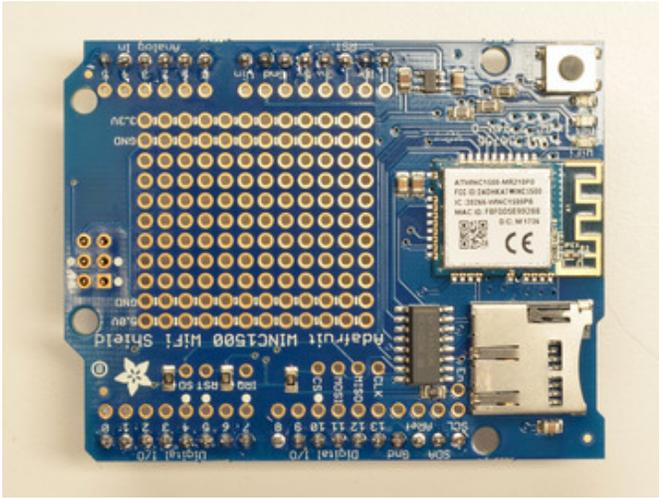


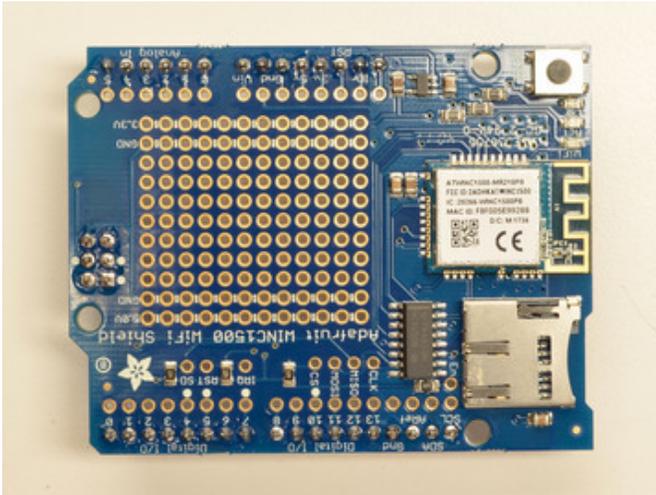


That's it! Now you can install the 2x3 header



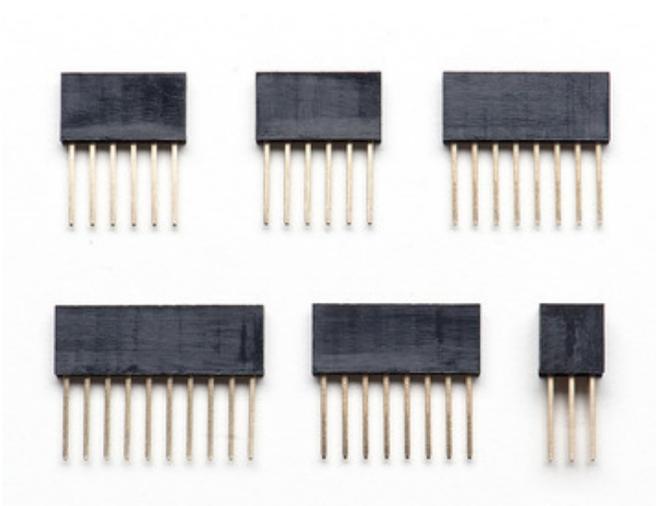
Solder the 2x3 header so that it's pointing downwards



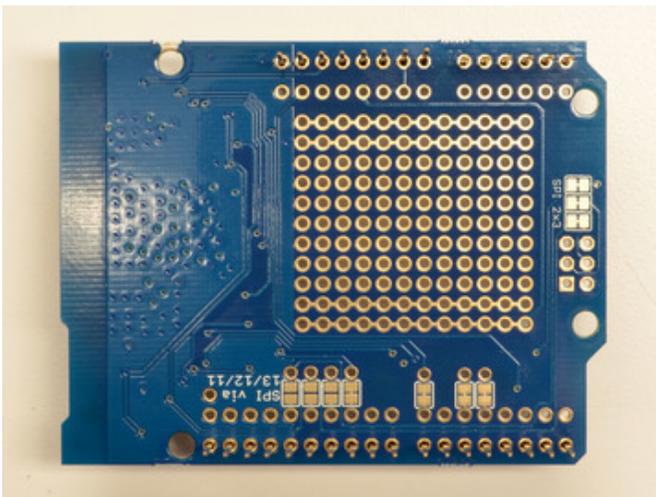


Stack Alert!

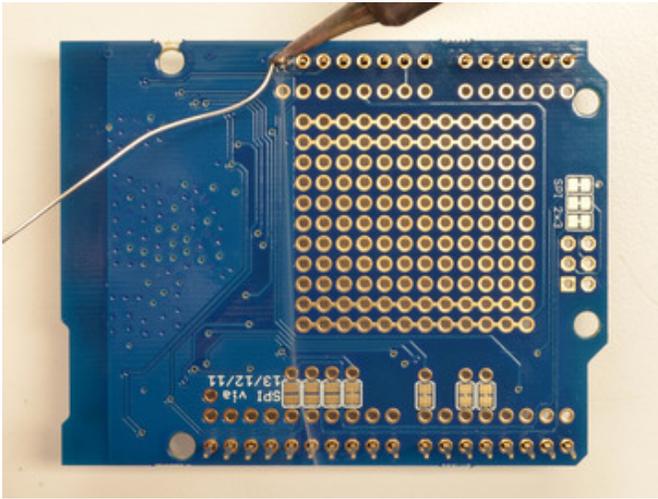
If you want to stack a shield on top of the WiFi Shield, you'll want to pick up some stacking headers and use those instead of the plain header shown here!



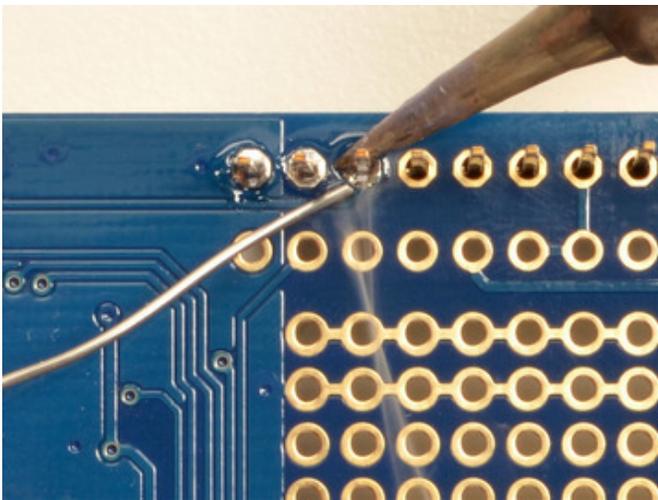
Wanna stack? This tutorial shows how to use the plain header to connect to an Arduino. [If you want to use stacking headers \(https://adafru.it/dsu\)](https://adafru.it/dsu), don't follow these steps!



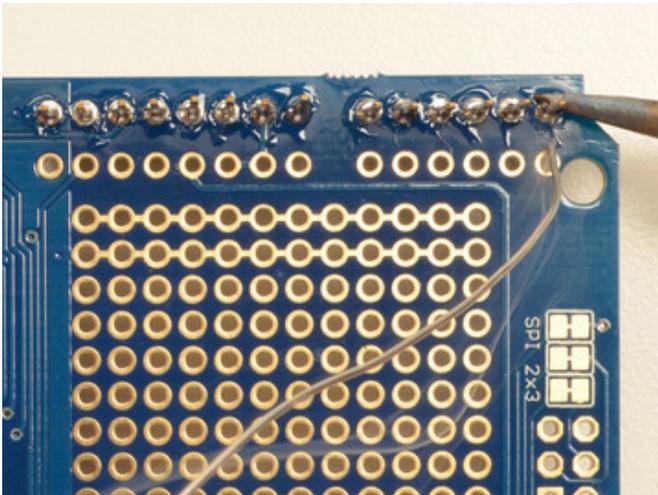
Start by sliding the 10 pin, 2 x 8 pin and 6-pin stacking headers into the outer rows of the shield from the top. Then flip the board over so its resting on the four headers. Pull on the legs if necessary to straighten them out.

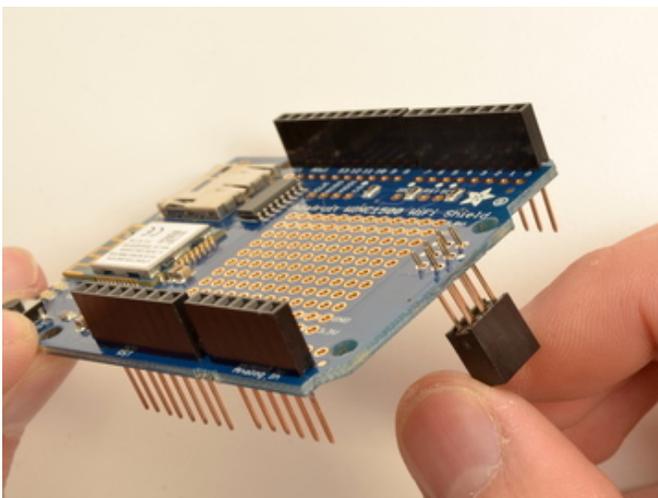
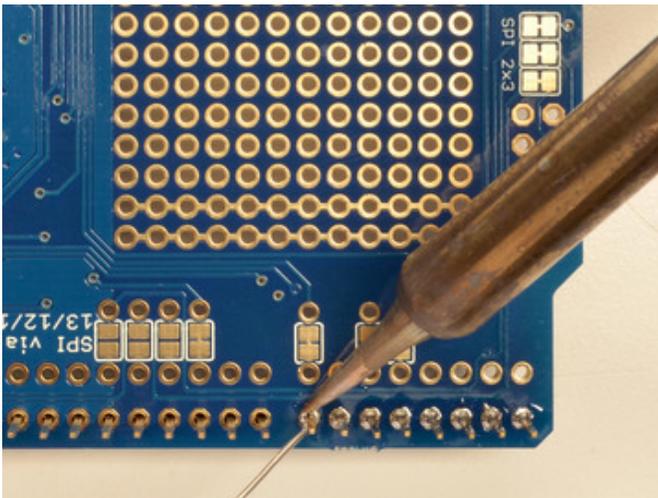
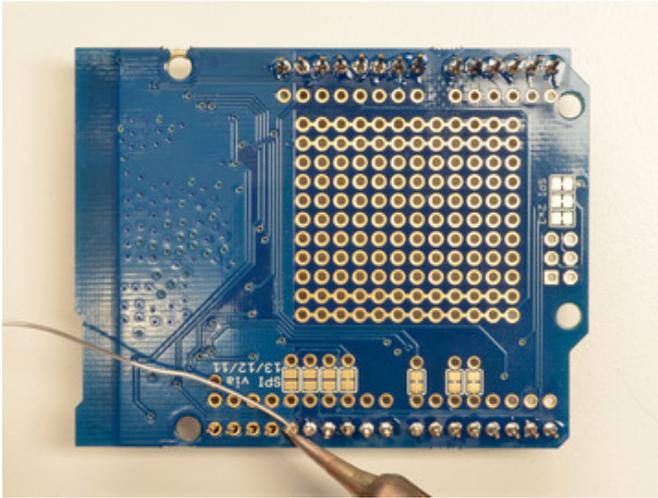


Tack one pin of each header, to get them set in place before more soldering. If the headers go crooked you can re-heat the one pin while re-positioning to straighten them up



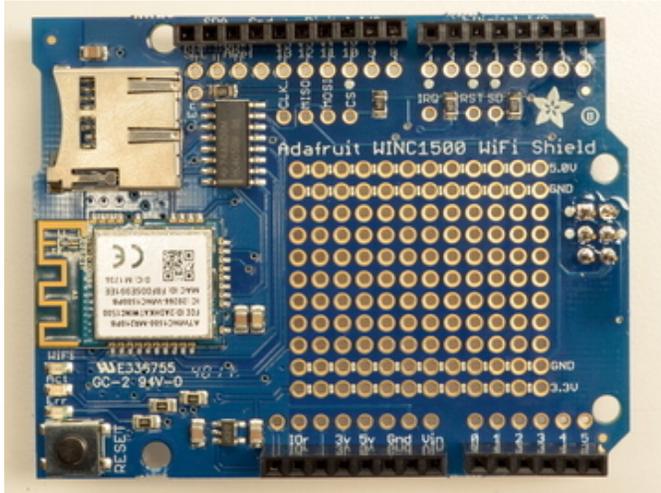
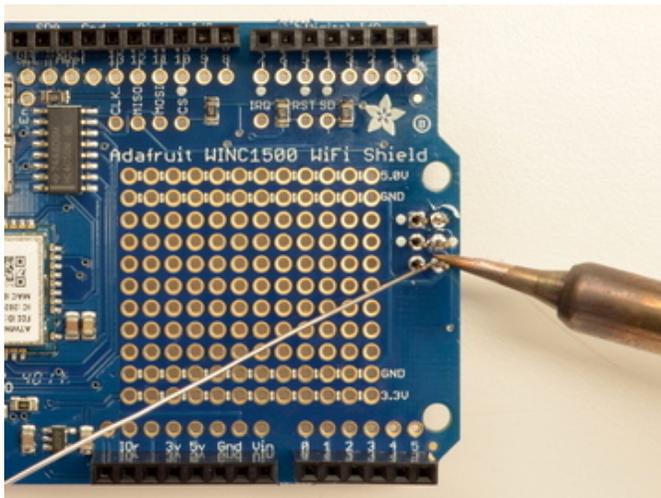
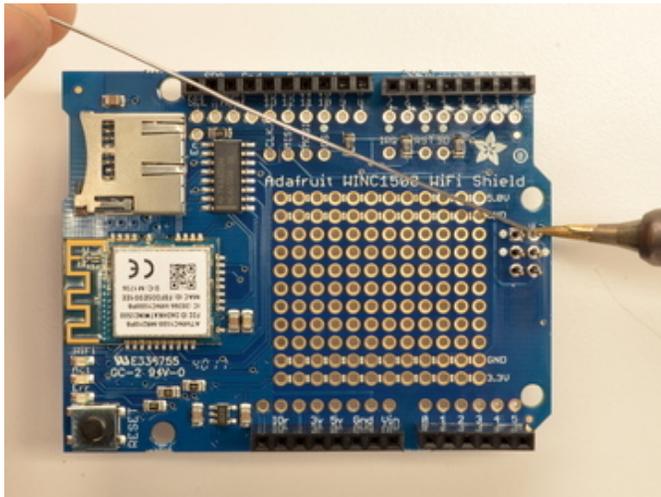
Once you've tacked and straightened all the headers, go back and solder the remaining pins for each header.

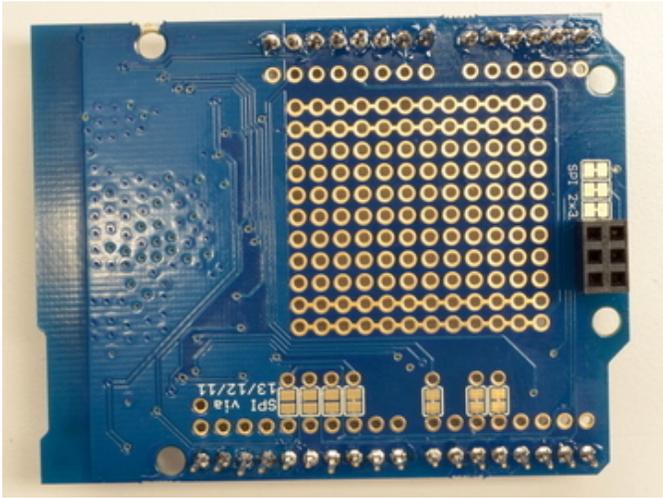




Insert the 2x3 stacking header as shown.

Solder into place.





Wiring & Test

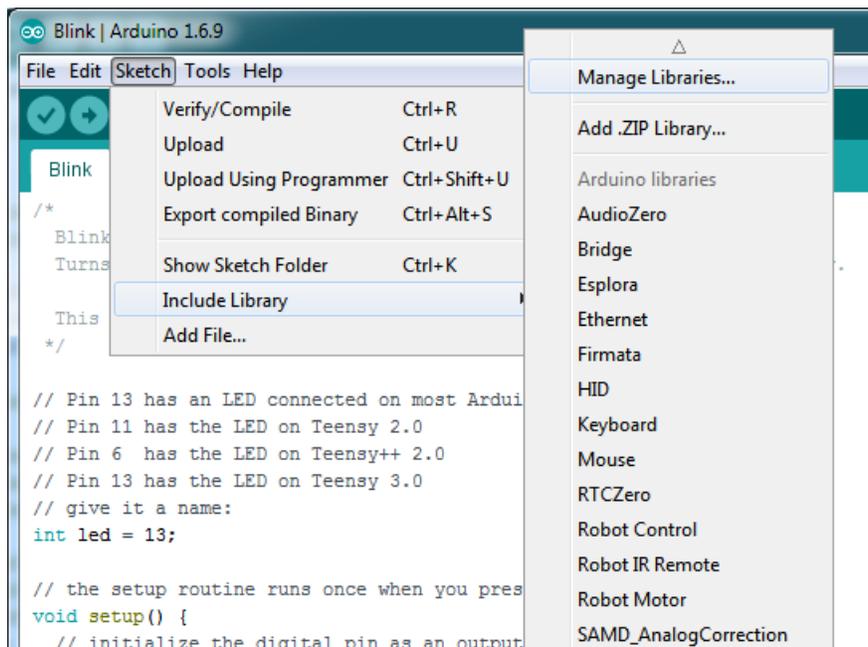
Once you have soldered headers onto the shield you can plug it into your microcontroller

For the initial demo you can use an Atmega328P or ATSAM21-based Arduino compatible. That means a UNO/Metro or Zero/Metro M0. We strongly recommend a ATSAM21/Zero based board as the library is quite large and takes up nearly all the memory on a '328.

Install the Library

We will start by installing the [official Arduino WiFi101 library \(https://adafru.it/kUF\)](https://adafru.it/kUF).

We want the latest version so visit the Library Manager



Type in wifi101 and when the library comes up, click **Install** or **Update** to make sure its the most recent one!

If you're not familiar with installing Arduino libraries, please visit our tutorial: [All About Arduino Libraries \(https://adafru.it/aYM\)](https://adafru.it/aYM)!

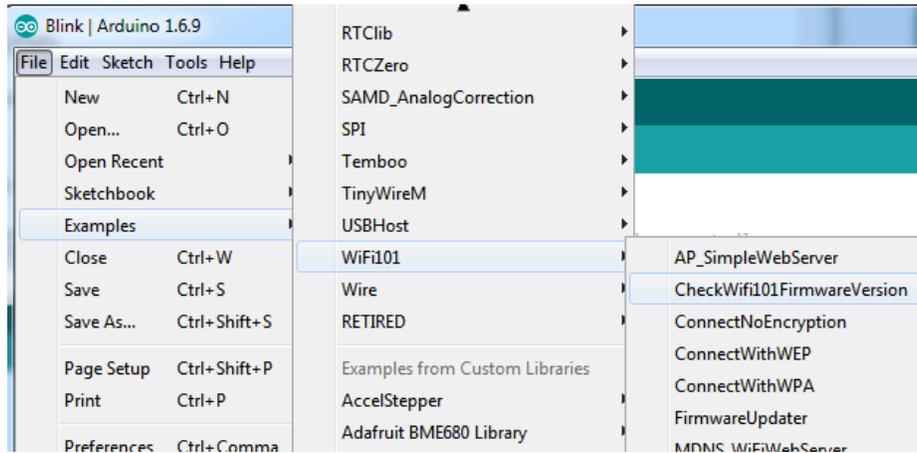
Restart the Arduino IDE.

You may need to use Arduino 1.6.5 or later

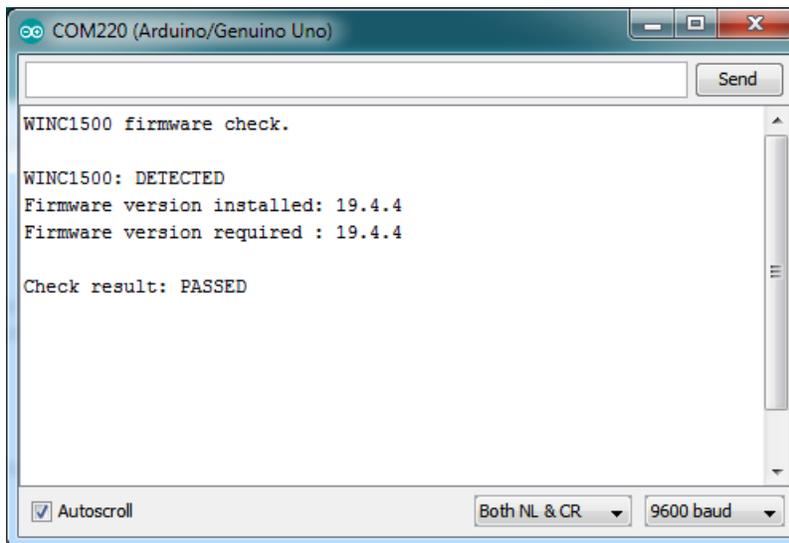
Check Connections & Version

Before we start, its important to verify you have the right setup & firmware version.

Load up the **WiFi101->CheckWifi101Firmware** sketch



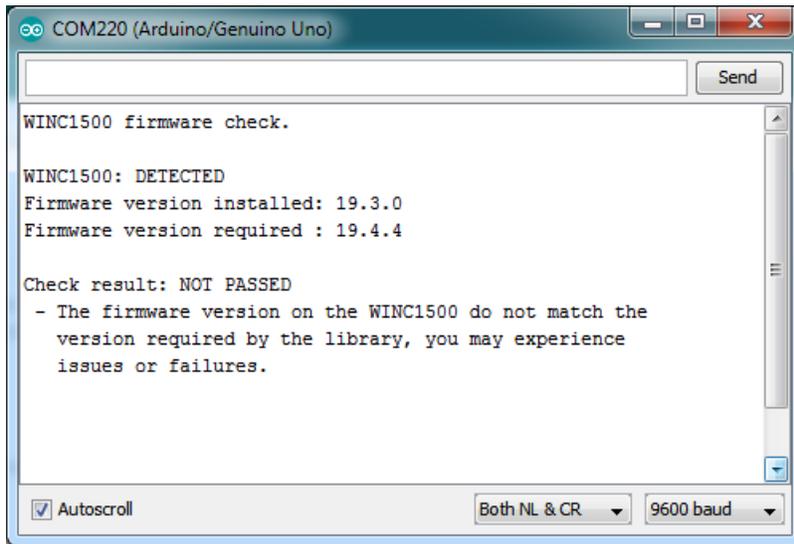
Upload to your Arduino and open up the Serial Console at 9600 baud:



You should see the firmware version. If your version has not **PASSED** no big deal - simply [follow this page to update your firmware!](https://adafru.it/vff) (<https://adafru.it/vff>)

If you have version 19.3 or less, the firmware is too old

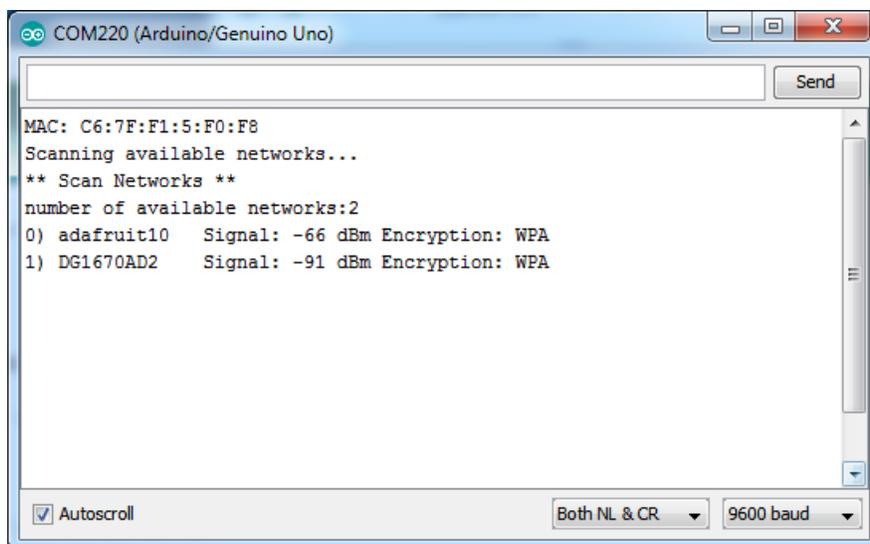
If you get not response, the firmware is either waay to old, or something is amiss with your wiring!



Scanning WiFi

Now that you have the right firmware version, lets scan for network!

Run the **WiFi101->ScanNetworks** example to see a list of available visible networks



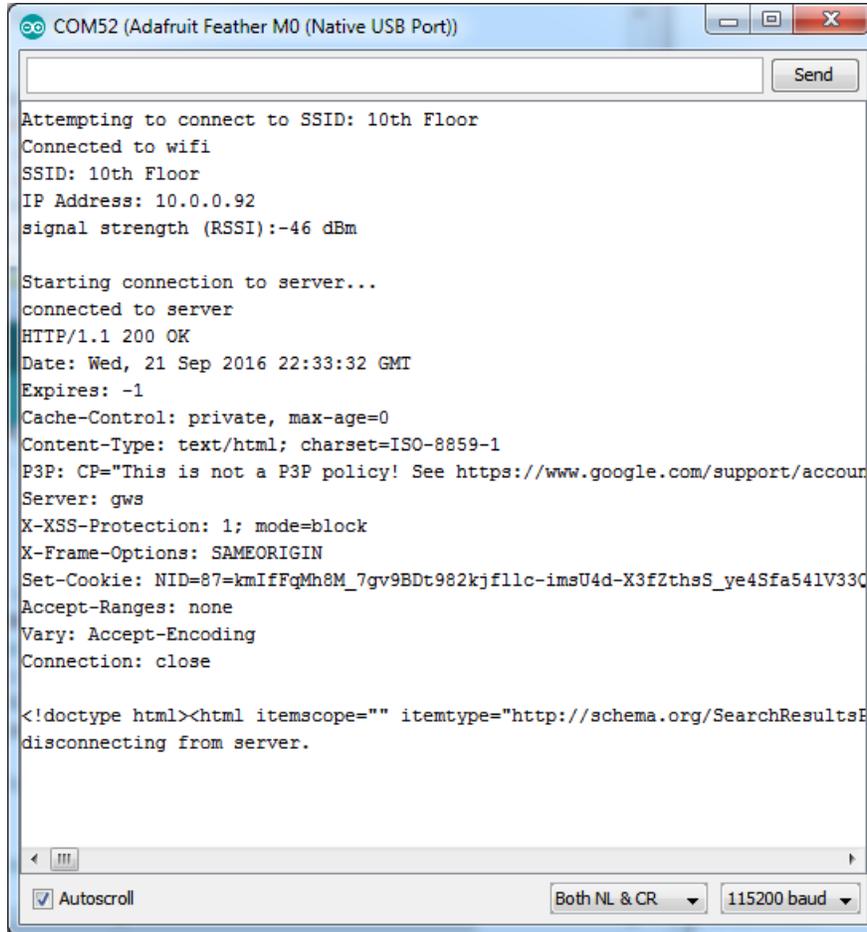
Connect & Read Webpage

OK finally you get to connect and read some data!

Open up the **WiFi101->WiFi101WebClient** example, then edit the **ssid** and **pass** variables to contain your network and password

```
34
35
36 char ssid[] = "adafruit"; // your network SSID (name)
37 char pass[] = "supersekret"; // your network password (use for WPA, or use as key for WEP)
38 int keyIndex = 0; // your network key Index number (needed only for WEP)
39
40 int status = WL_IDLE_STATUS;
41 // if you don't want to use DNS (and reduce your sketch size)
42 // use the numeric IP instead of the name for the server:
```

It will connect to the website in **server** and read the **webpage** manually:



That's it! pretty easy, huh? There's other examples you can try such as server mode, UDP data transmission & SSL

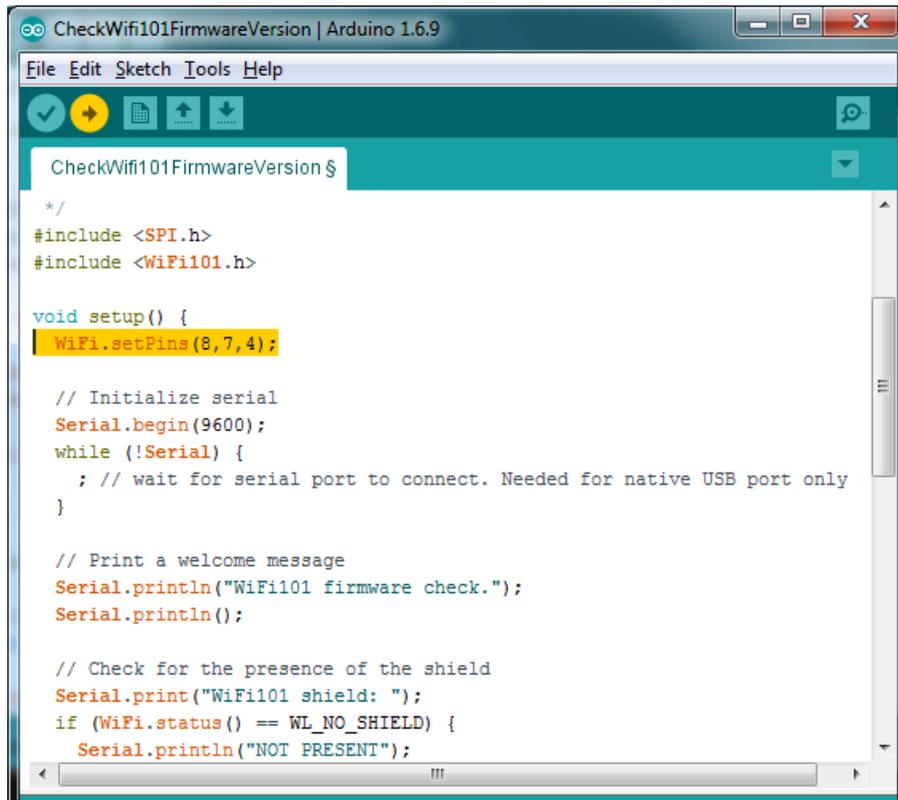
Changing the control pins?

If you are not using the default pin configuration on the shield, you'll need to **setPins** to tell the library where to find the control pins.

You can call **setPins** with the new **CS**, **IRQ** and **Reset** pin. If you are using the **Enable** pin, pass that in fourth:

```
WiFi.setPins(int8_t cs, int8_t irq, int8_t rst, int8_t en = -1);
```

Like so:



```
CheckWifi101FirmwareVersion | Arduino 1.6.9
File Edit Sketch Tools Help
CheckWifi101FirmwareVersion $
*/
#include <SPI.h>
#include <WiFi101.h>

void setup() {
  WiFi.setPins(8,7,4);

  // Initialize serial
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

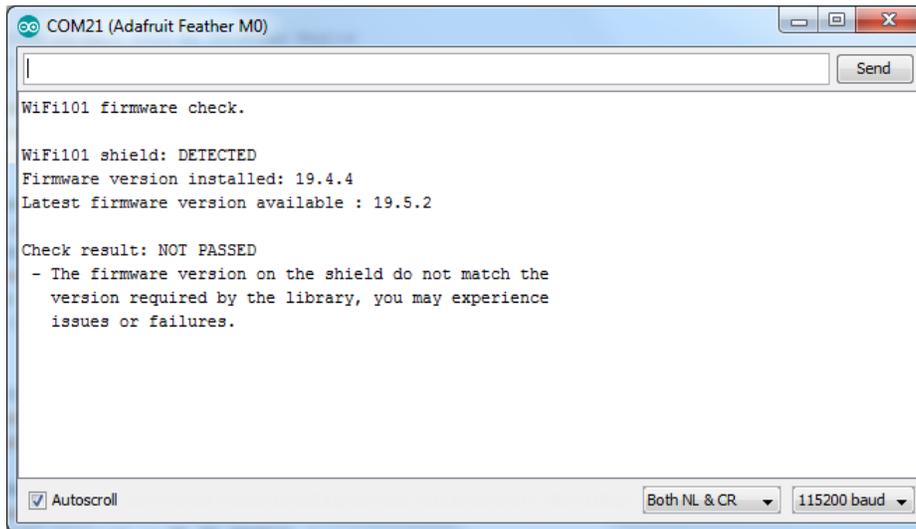
  // Print a welcome message
  Serial.println("WiFi101 firmware check.");
  Serial.println();

  // Check for the presence of the shield
  Serial.print("WiFi101 shield: ");
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("NOT PRESENT");
  }
}
```

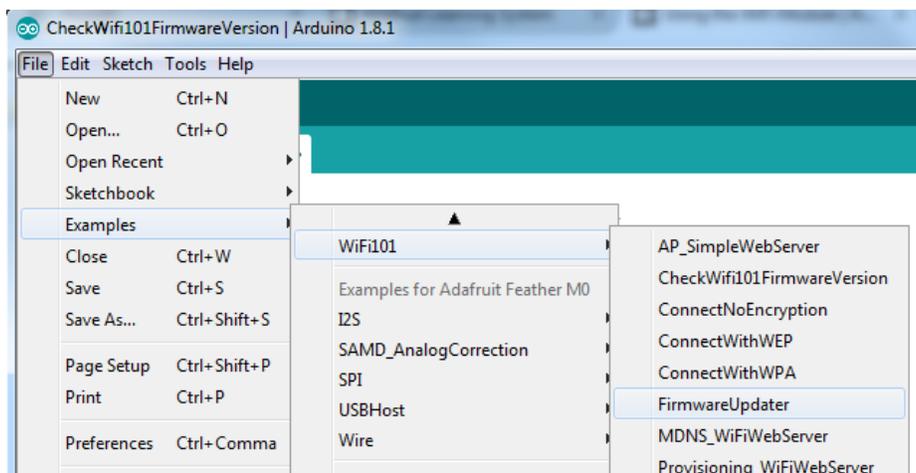
Updating Firmware

If you're running 19.5.2 there is no need to update to 19.5.4, despite what the WiFi101 library says it should work just fine and you could brick your device by updating, it is not perfectly foolproof! Thanks :)

As new versions of the WiFi101 library come out, you may end up getting a complaint that the library and WINC1500 firmware are out of sync:

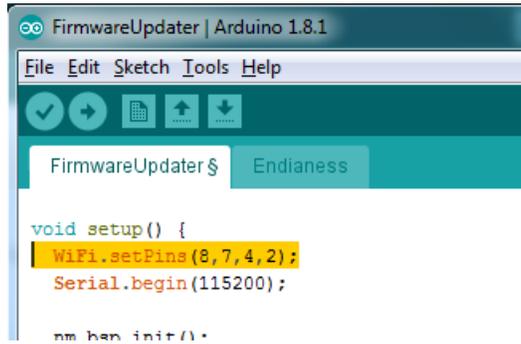


No problem - you can update the firmware through your Arduino/compatible! Start by loading up the **FirmwareUpdater** sketch



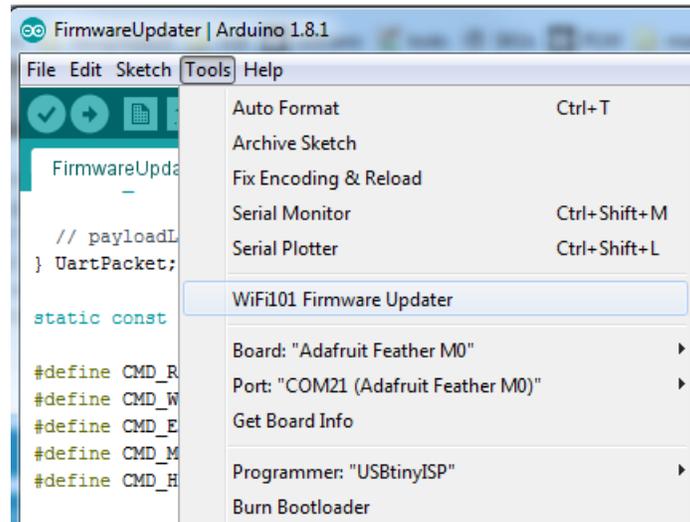
If you are using a Feather M0 or WINC1500 breakout, don't forget to update the pins as necessary with `setPins()`!

If you are using a WiFi101 or WINC1500 shield, you do not have to add `setPins()` code

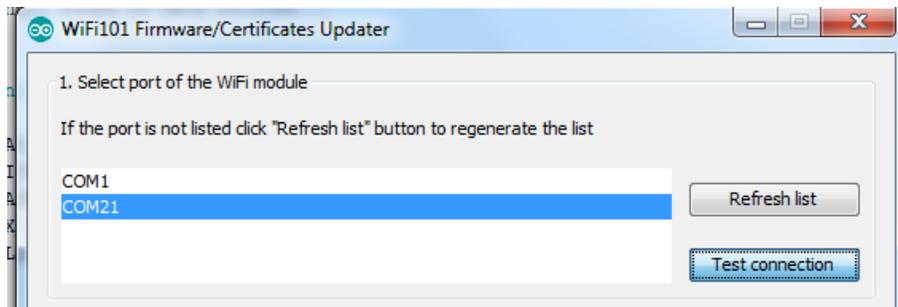


Upload it to your board. Make sure the Serial console is not open before or after uploading.

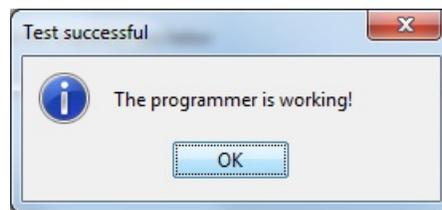
Then select the Updater tool built into the IDE



Select the right COM port, and click Test Connection

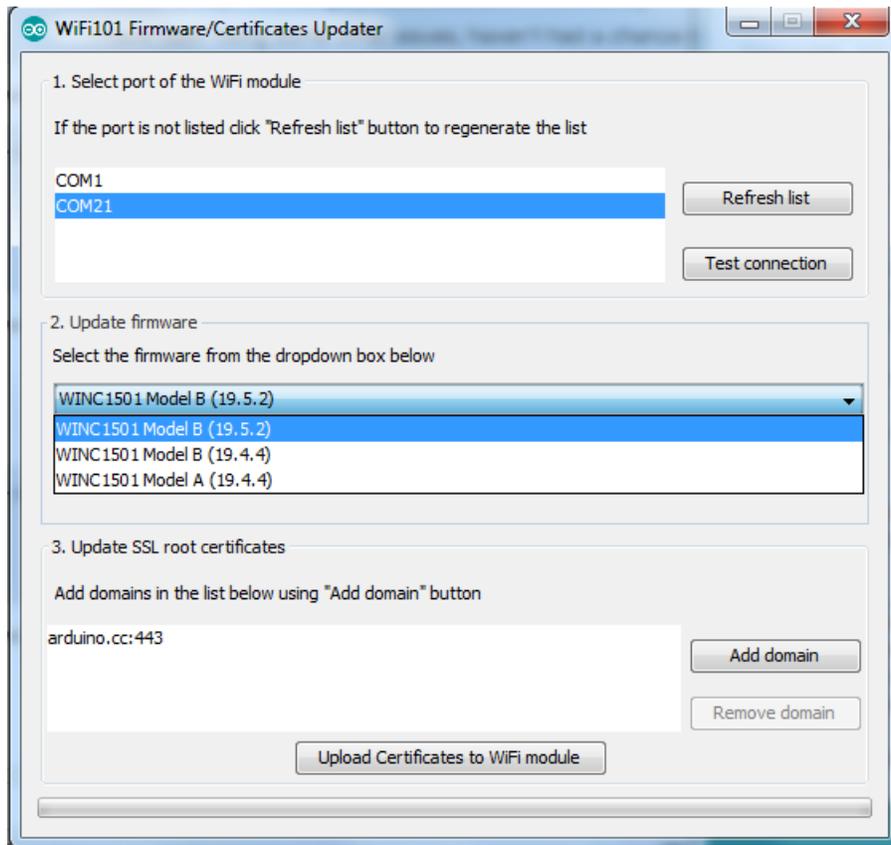


If all is good you'll get a confirmation

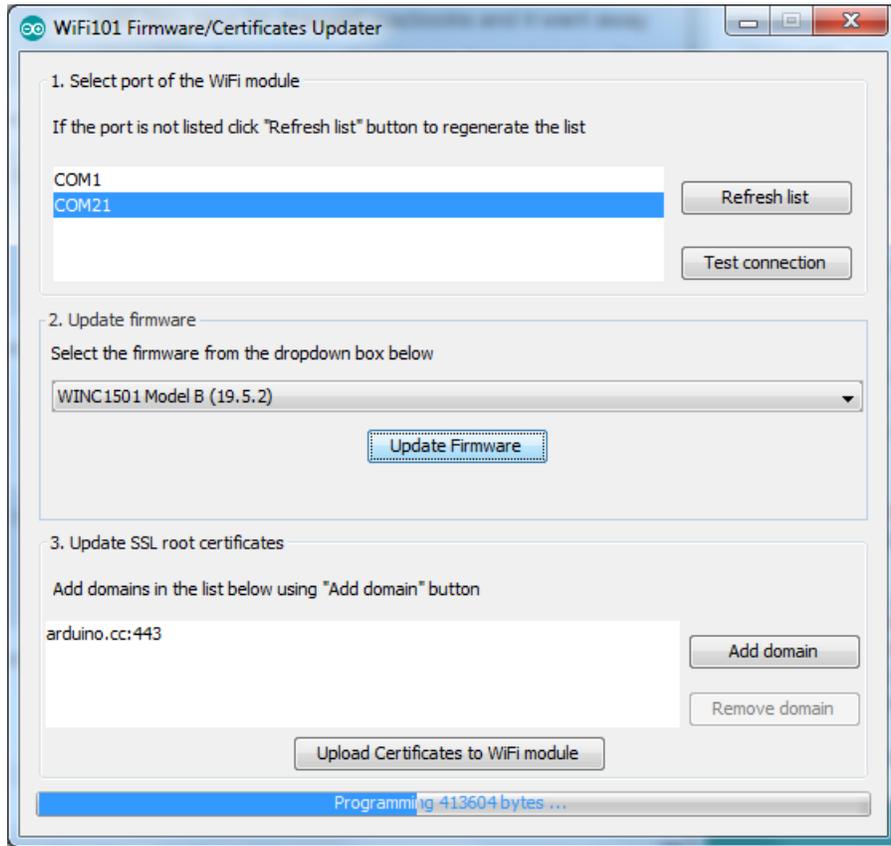


Next, select the firmware - we of course recommend the latest version!

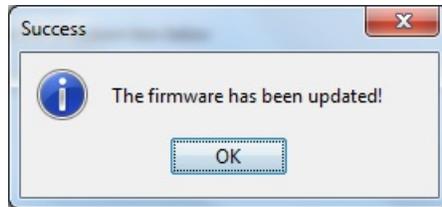
If you don't see the right/matching version you may need to update the IDE



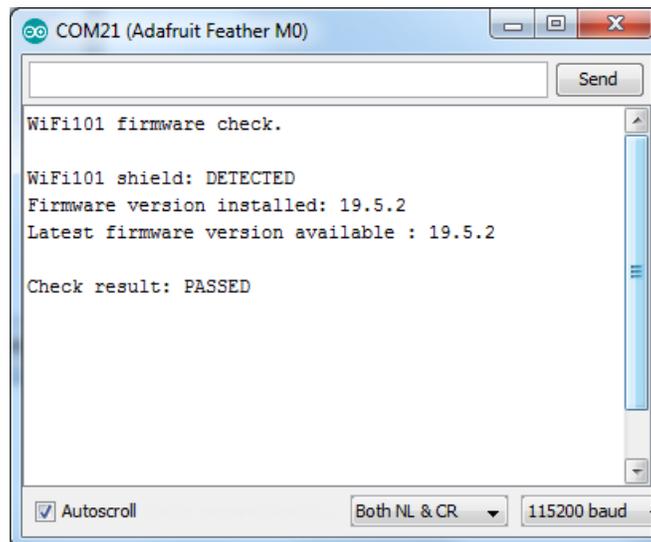
Once you feel ready - make sure the USB cable is connected solidly! Click **Update Firmware**



And a minute or two later...



Now you're ready to rock! Reload the Firmware Check sketch from before, this time you will see:

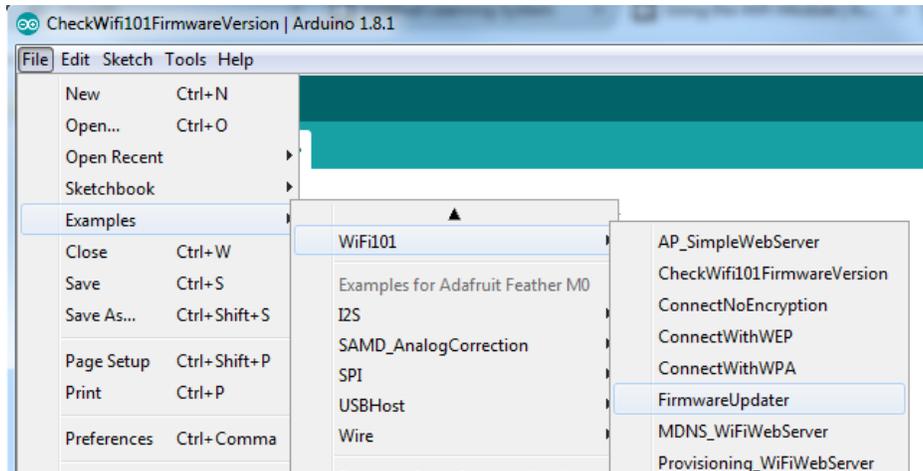


Updating SSL Certificates

If you're trying to connect to a computer or service via SSL and the connection is failing, you may need to update the certificates built into the WINC1500. By default it comes with many of the most popular SSL certificates but you may bump into a site that requires one that isn't included.

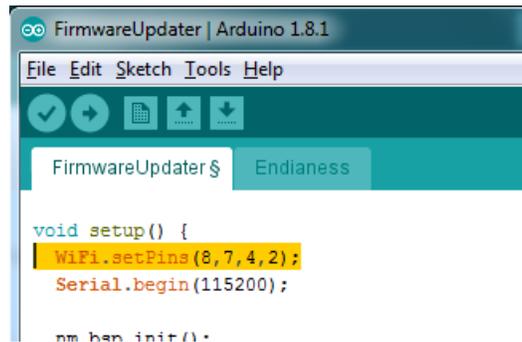
It's quite easy to update the certificates, you'll need to upload some code and run the uploader but it only has to happen once.

Start out by uploading the **FirmwareUpdater** sketch from **WiFi101**



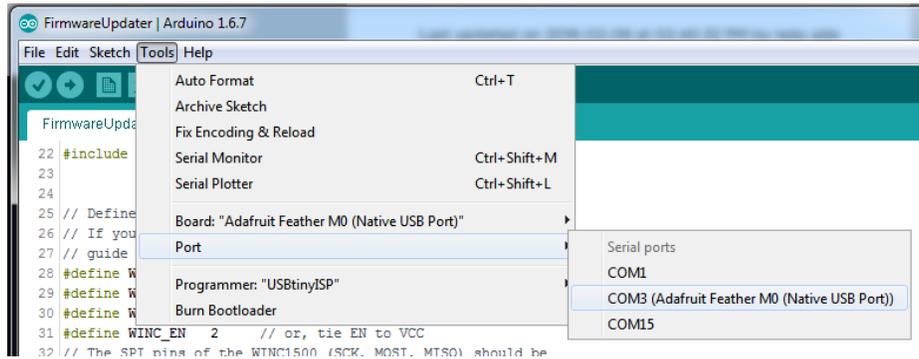
If you are using a Feather M0 or WINC1500 breakout, don't forget to update the pins as necessary with `setPins()`!

If you are using a WiFi101 or WINC1500 shield, skip this step



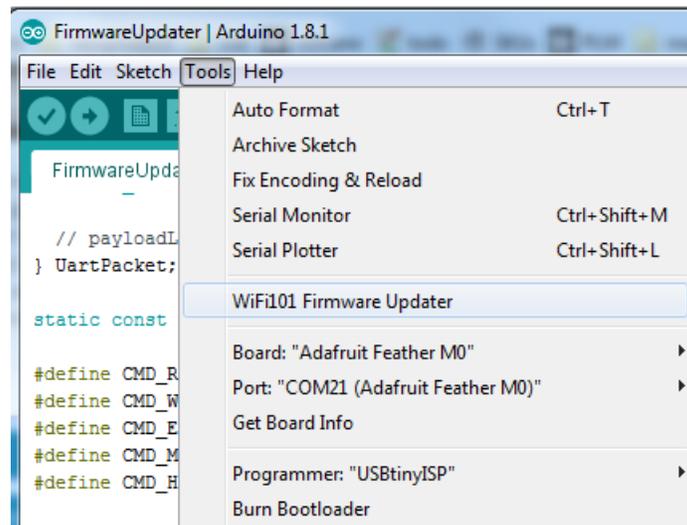
and upload it!

After uploading be sure to note what is the name of the COM or Serial port for the Arduino Zero or Feather...You'll need this for the next step

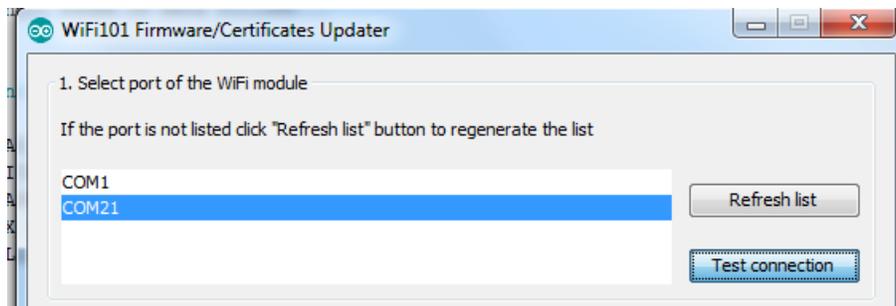


Upload it to your Feather. Make sure the Serial console is not open before or after uploading.

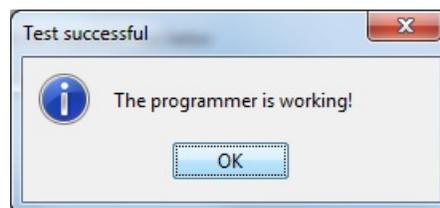
Then select the Updater tool built into the IDE



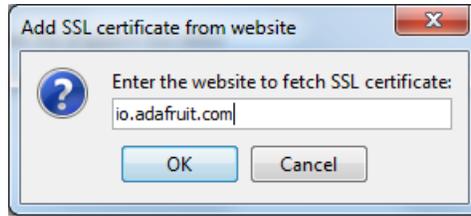
Select the right COM port, and click Test Connection



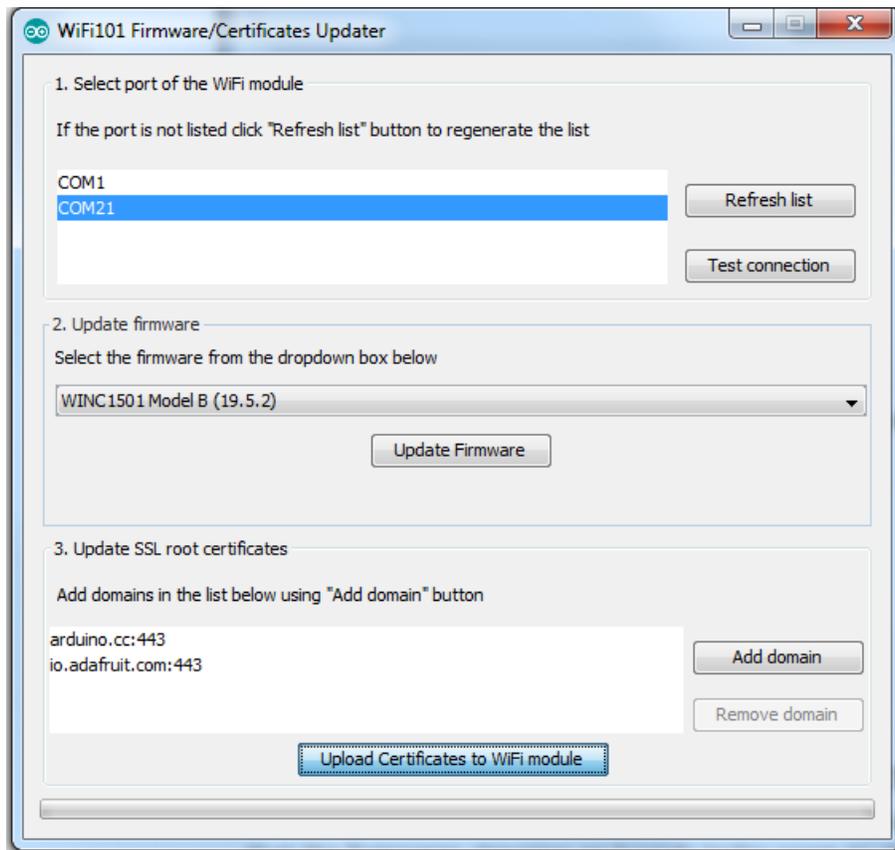
If all is good you'll get a confirmation



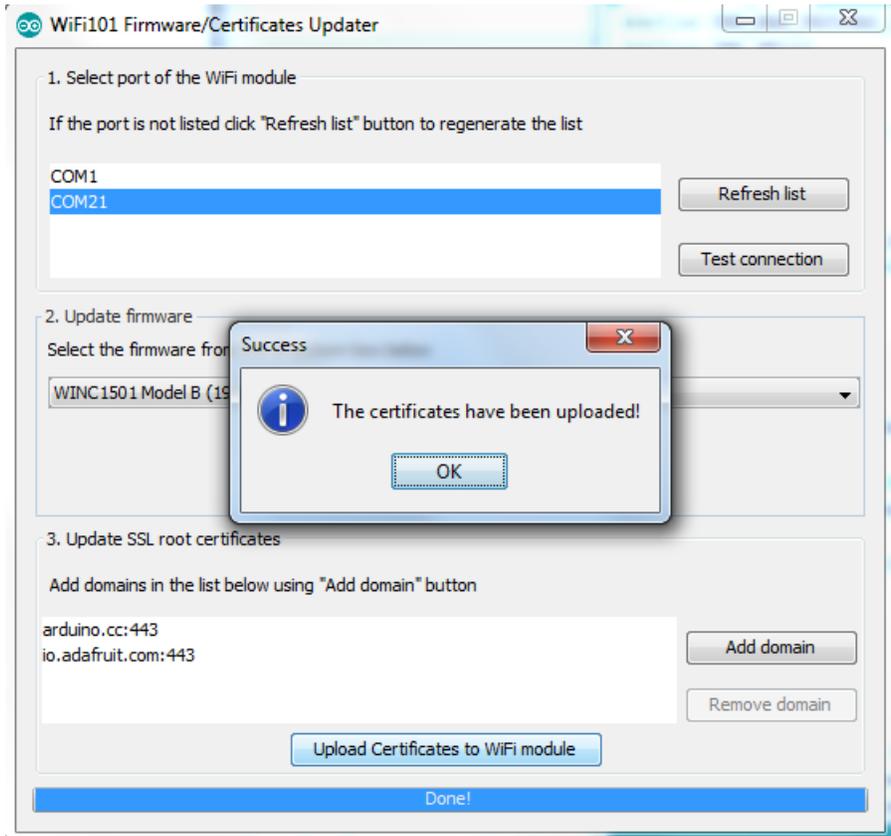
Now at the bottom of the page, click **Add Domain** and type in the URL of the site you want to access:



Then click **Upload Certificates**



A few moments later...success!



What SSL/TLS support is available with the WINC1500?

Officially Atmel lists TLS 1.0 & 1.1, however we have noticed that the firmwares shipping on boards today seem to also support TLS 1.2 (verified by checking the results of www.howssmyssl.com).

The supported ciphers are:

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA

Downloads

Datasheets & Files

- All the details you could ever want are over at the Atmel website (<https://adafru.it/kUD>)
- EagleCAD PCB files on GitHub (<https://adafru.it/BtP>)
- Fritzing objects in the Adafruit Fritzing library (<https://adafru.it/aP3>)

