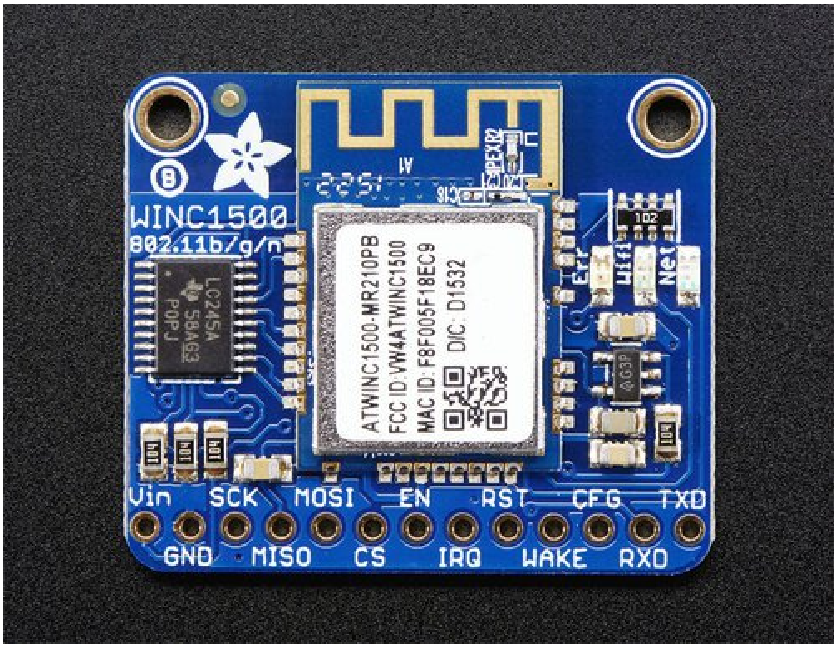
Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Adafruit ATWINC1500 WiFi Breakout

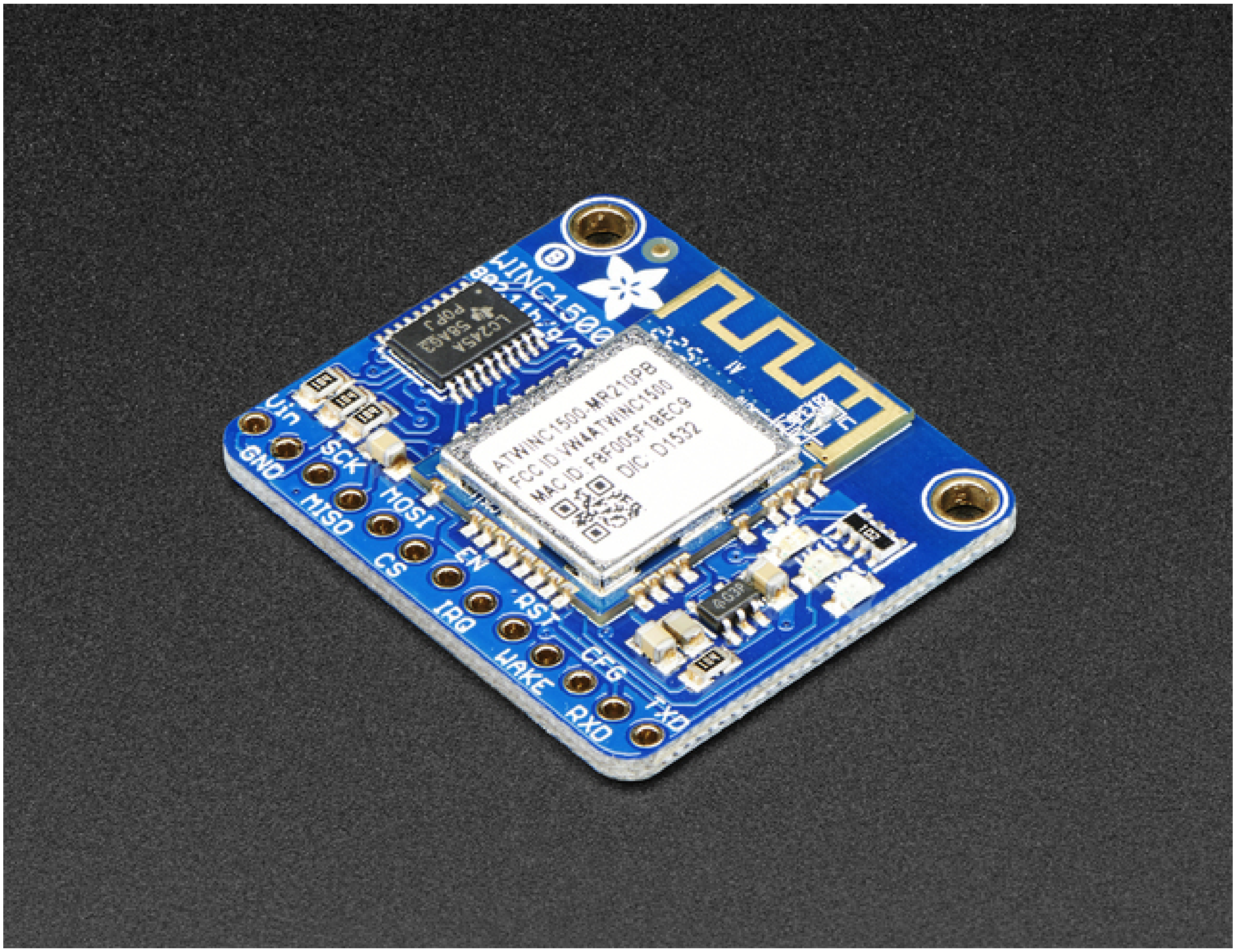
Created by lady ada



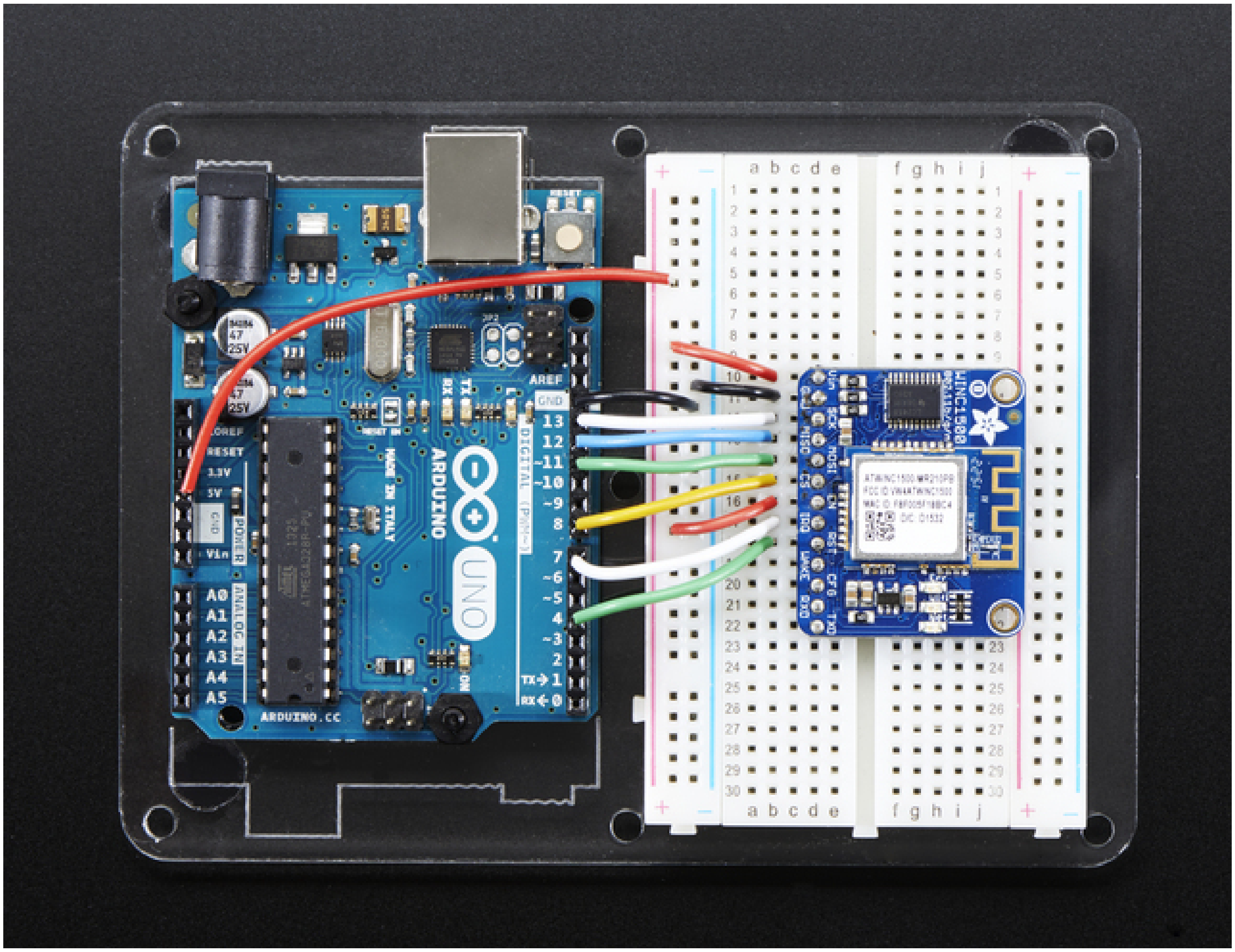
Last updated on 2017-03-30 05:15:24 PM UTC

Guide Contents

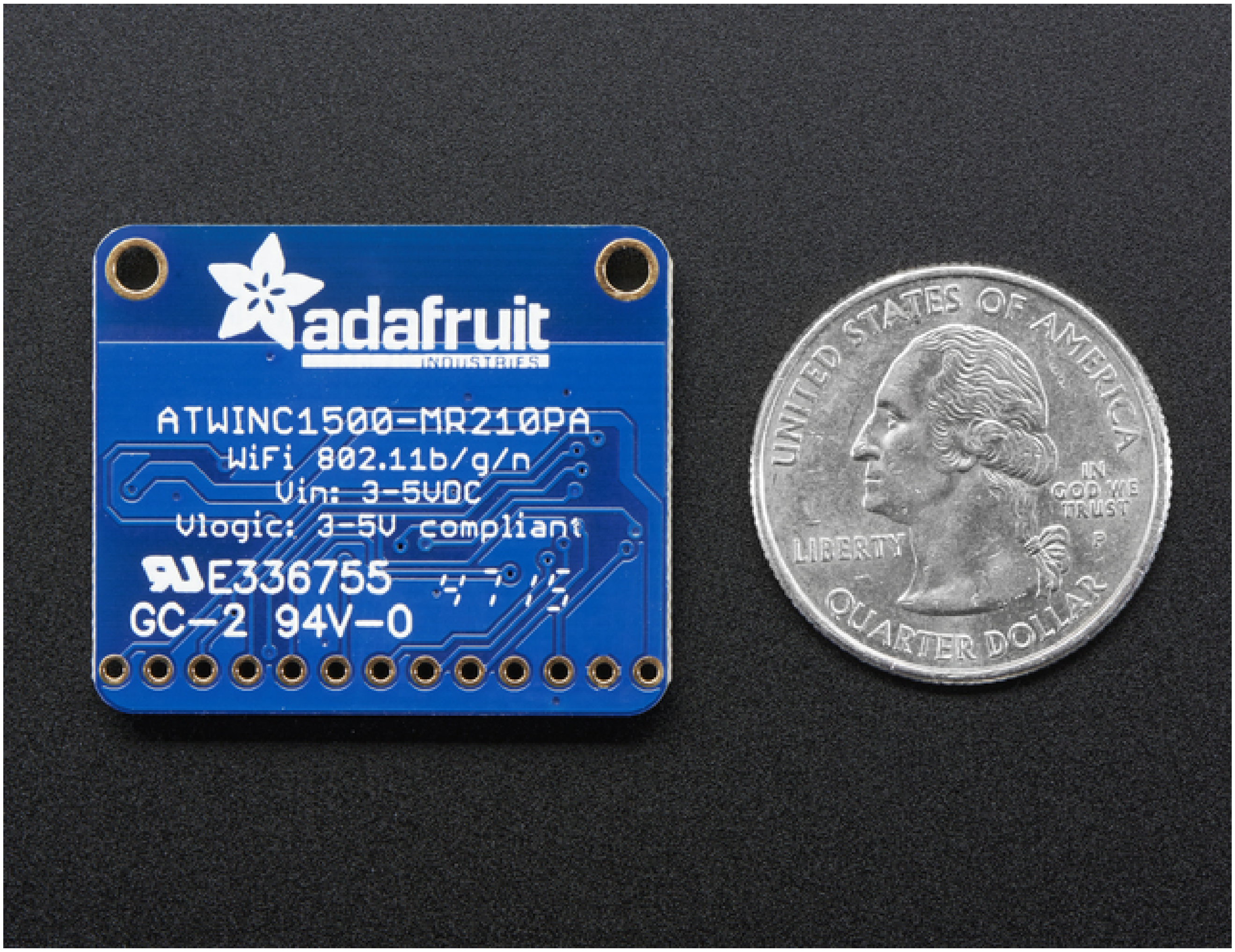
Overview



Connect your Arduino to the Internet with this fine new FCC-certified WiFi 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). We like these so much, they've completely replaced the CC3000 module on all our projects.

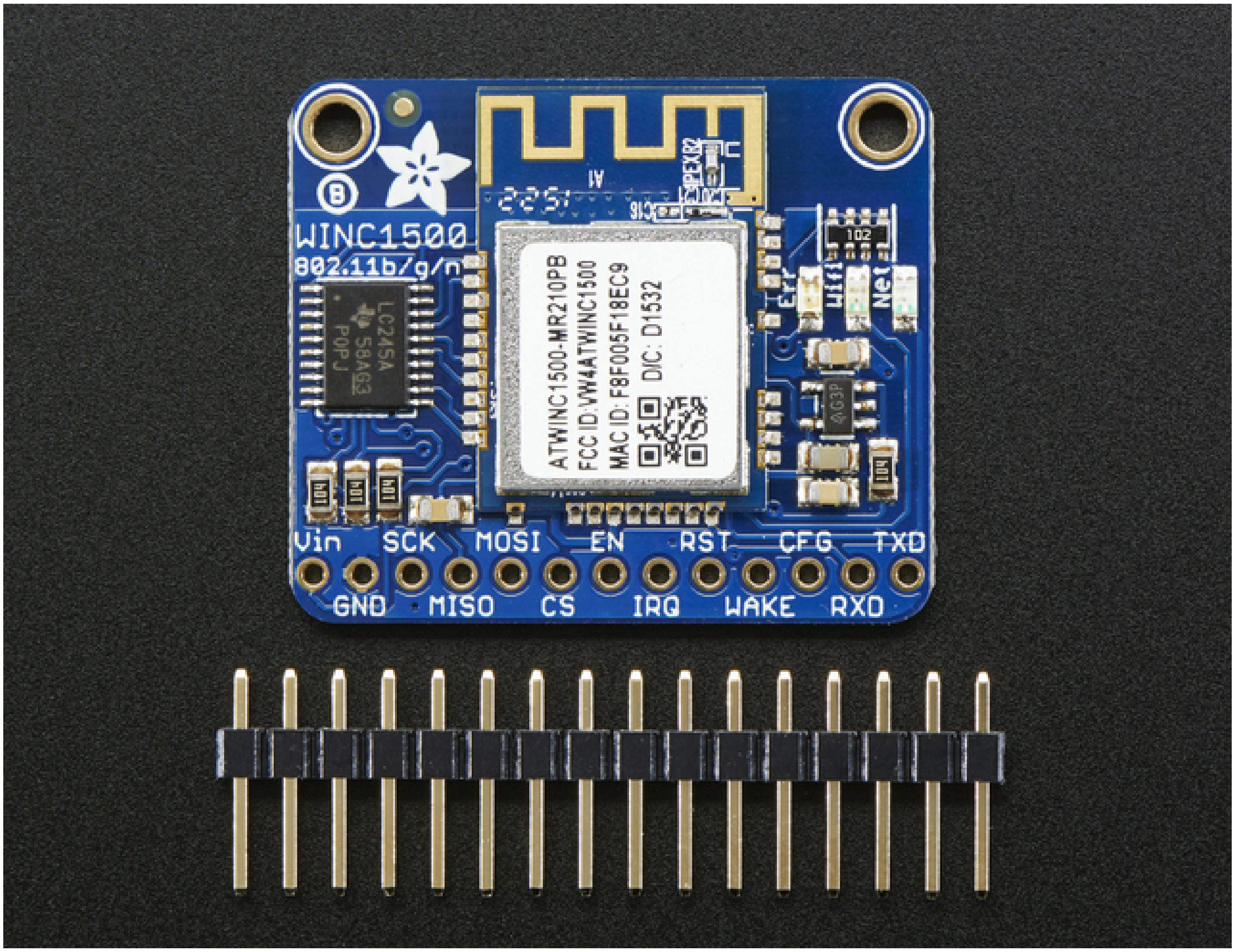


The ATWINC1500 uses SPI to communicate, so with about 5 or 6 wires, you can get your wired up and ready to go. Right now the Atmel-supplied library works great with Arduino Uno or Zero, but may not work on other Arduinos. You can clock it as fast as 12MHz for speedy, reliable packet streaming. Communication is done through the standard Client & Server interface so all your Ethernet & older WiFi code is easy to adapt. Scanning/connecting to networks is very fast, a few seconds.



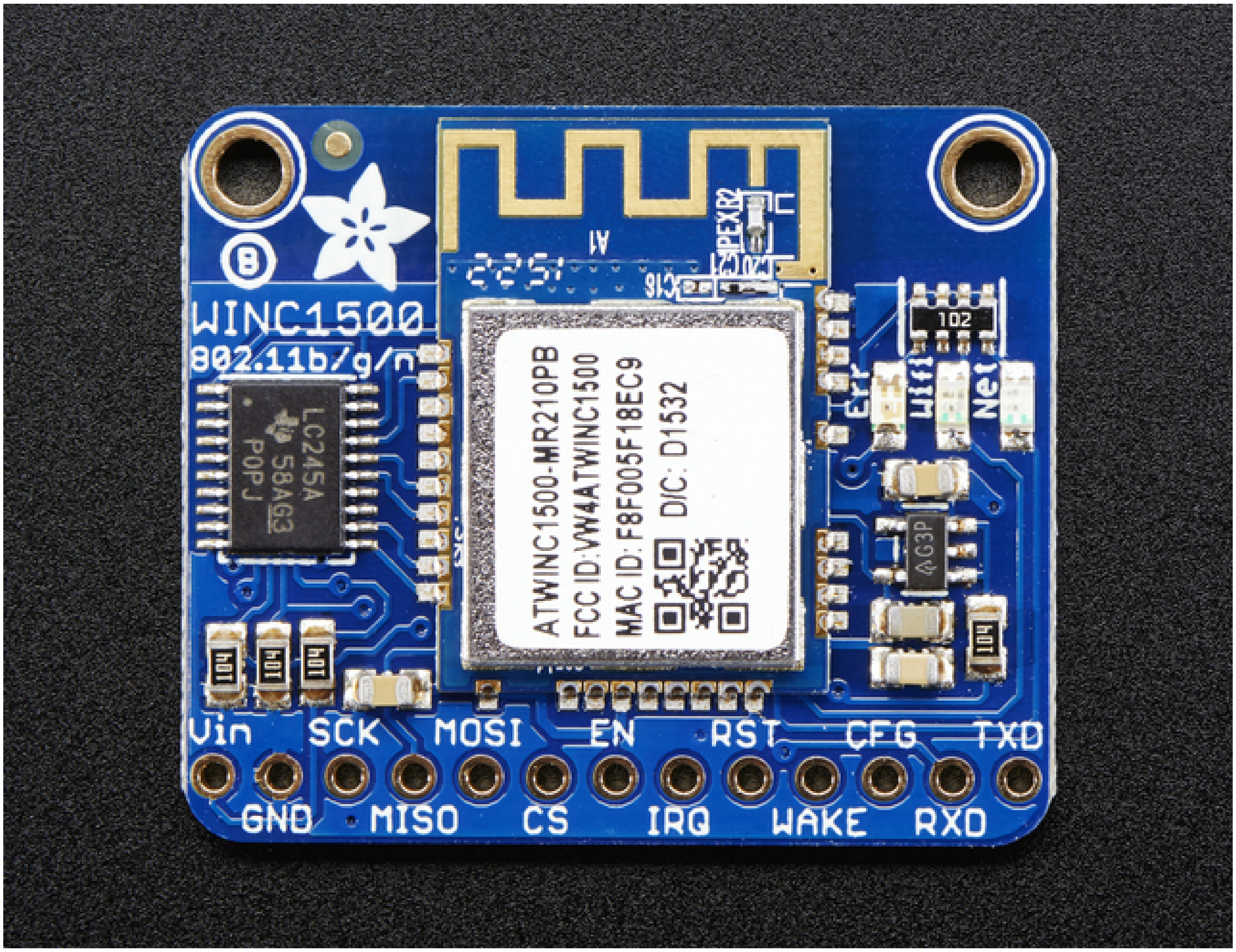
This module works with 802.11b, g, or n networks & supports WEP, WPA and WPA2 encryption. The datasheet says it can do Soft-AP mode but we don't have any code to actually use that.

Since this is our new favoritest SPI-protocol WiFi module we've decided to make a little breakout for it. The breakout comes with level shifting on all the input pins so you can use it with 3V or 5V logic. 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.



Comes with a stick of header you can solder on, to plug into a breadboard and a set of tutorials & code so you can follow along!

Pinouts



The ATWINC1500 module does have a bunch of pins, but they're pretty easy to understand. Lets check it out!

Power Pins

- **Vin** - this is the power-in pin. Connect to 3.3 - 5.5VDC
  The wifi module can draw up to 300mA during transmit (for small blips of time) So make sure that your power supply can supply it.
- **GND** - ground for signal and power

SPI Pins

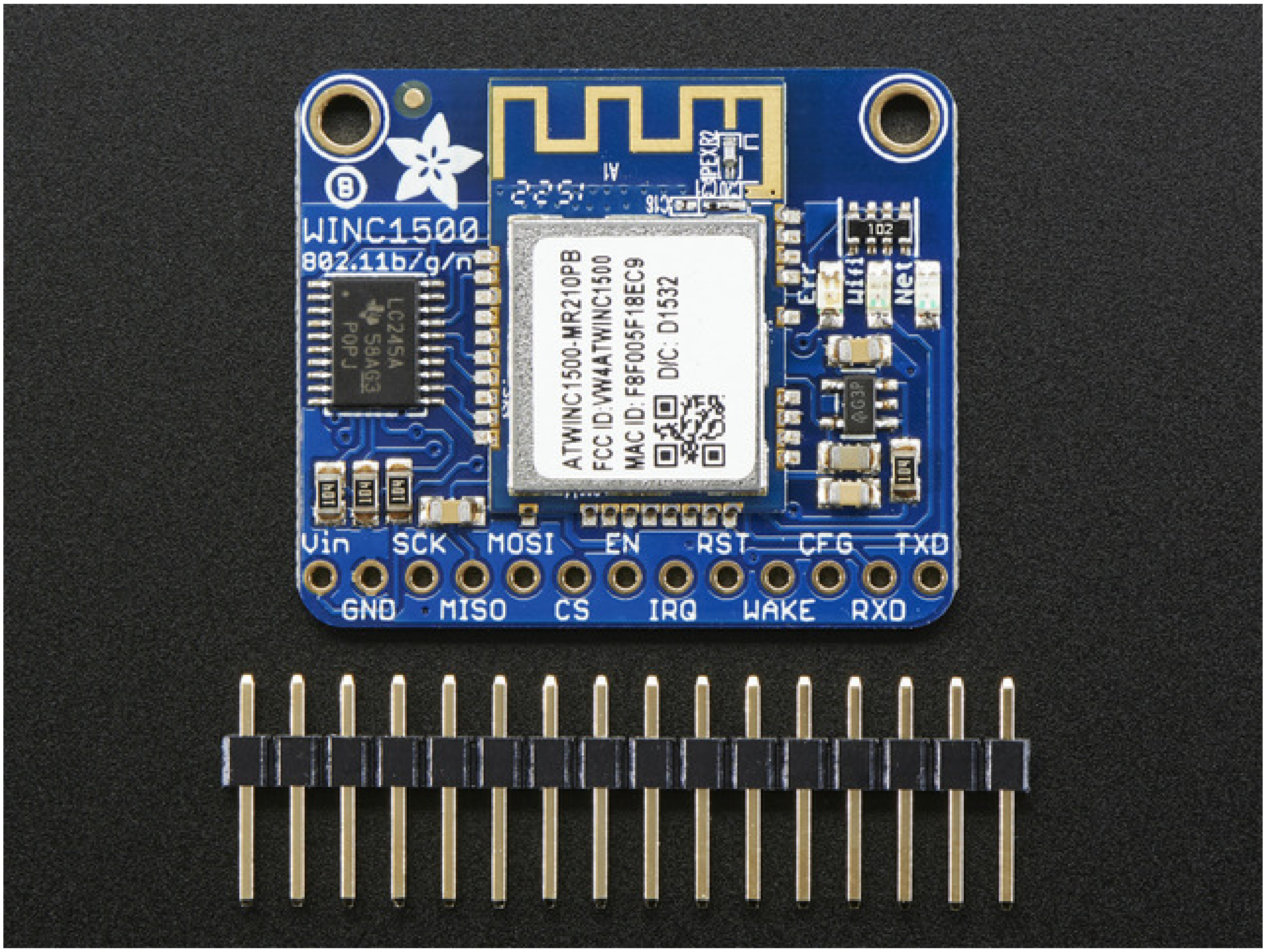
This is how you send and receive data from the module

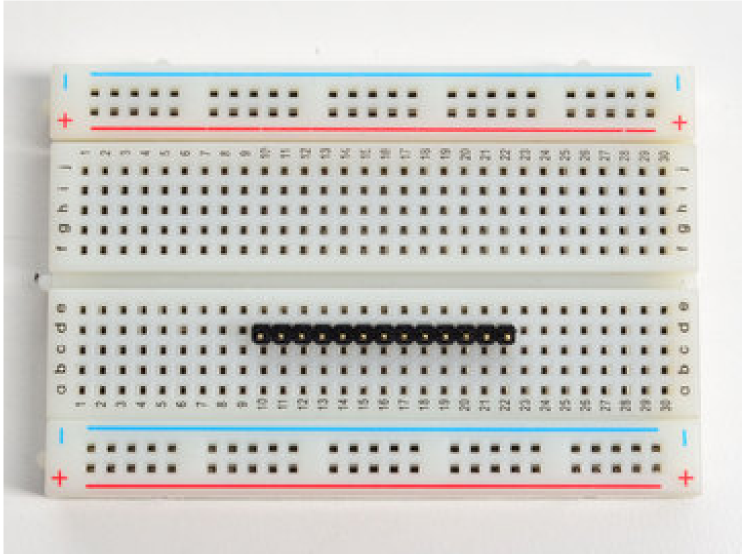
- **SCK** - SPI clock input, 3V or 5V compliant
- **MISO** - SPI data out from module, 3.3V line level
- **MOSI** - SPI data into module, 3V or 5V compliant
- **CS** - SPI chip select, pull down when transmitting to/from the ATWINC. By default this is pulled to Vin with a 100K resistor

Other SPI Interface Pins

- **EN** - Enables the entire module, by default tied low with a 100K resistor. Tie to 3-5V to keep the module on all the time, connect to a ground signal to disable the module
- **IRQ** - Interrupts from the module, connect to your microcontroller's INT input line. 3.3V logic level
- **RST** - Module reset, by default tied low with a 100K resistor. Pull high to bring out of reset.
- **Wake** - wake input signal, used to wake up the module (not used in existing code, but available if you can figure it out!) 3-5V logic in
- **CFG** - allows you to select between SPI (default) or UART data transport. Since we don't have any UART code, keep disconnected
- **RXD**/**TXD** - UART data transport pins. Since we don't have any UART code, keep disconnected

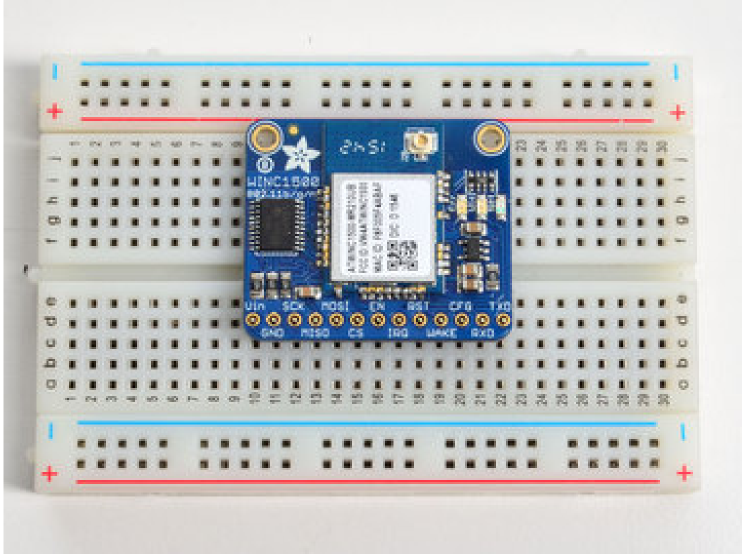
Assembly



- 

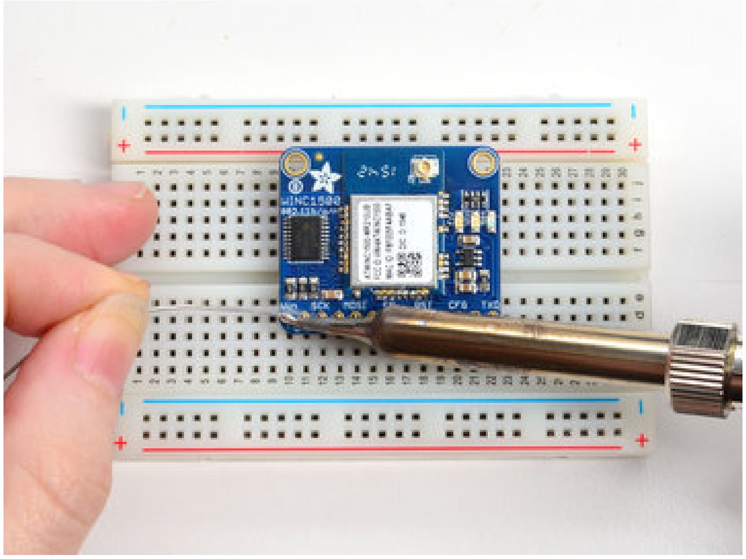
Prepare the header strip:

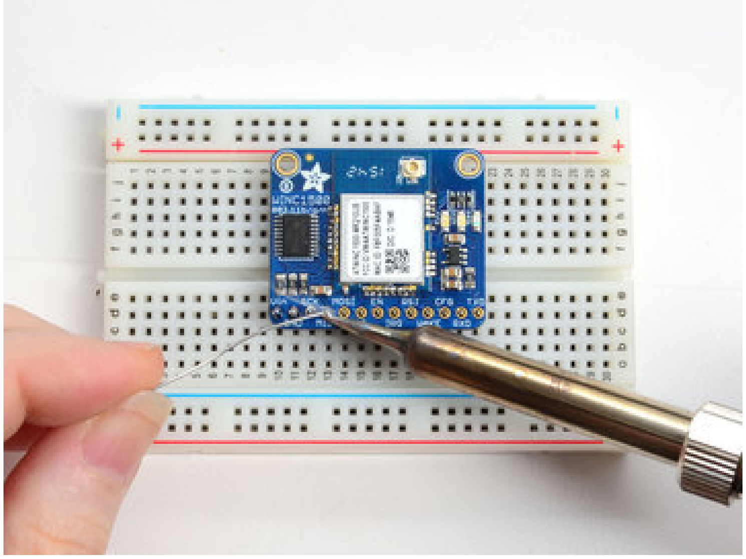
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**

- 

Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads

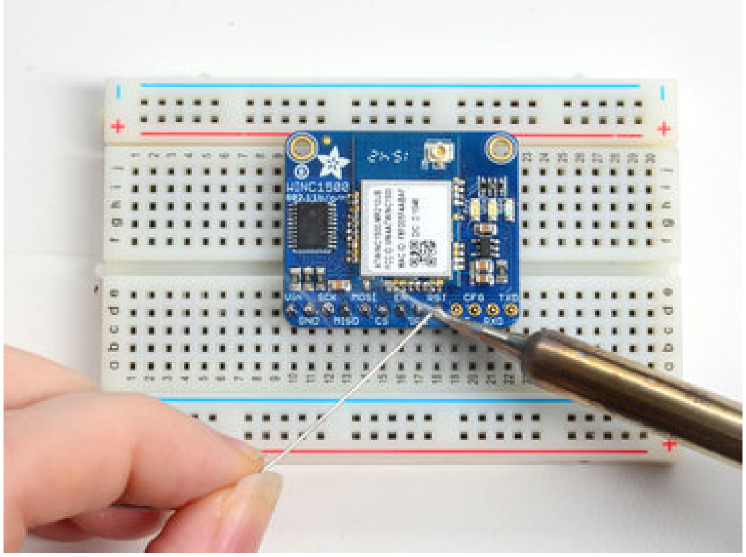
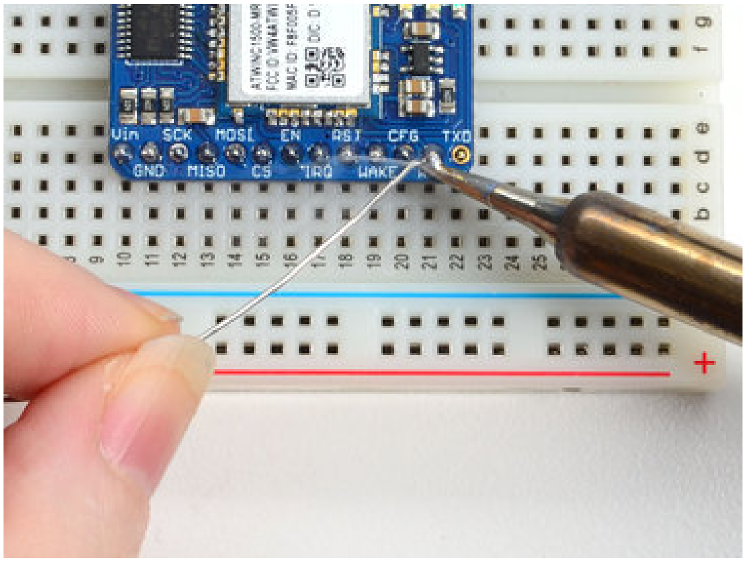
- 

- 

And Solder!

Be sure to solder all pins for reliable electrical contact.

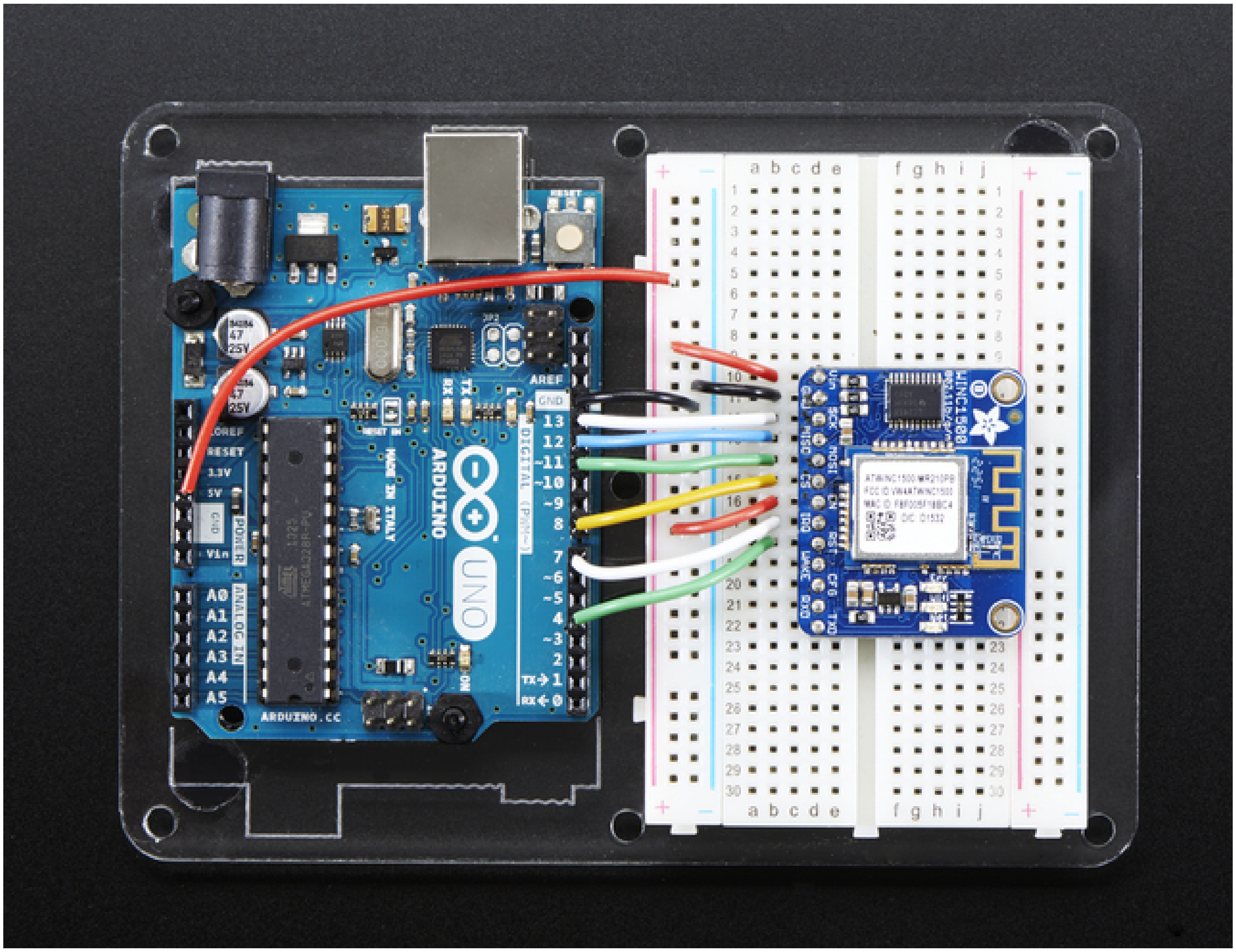
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](http://adafru.it/aTk) (http://adafru.it/aTk)).

- 
- 

-

  You're done! Check your solder joints visually and continue onto the next steps

Wiring & Test



For this initial demo we'll be using an Arduino UNO to connect. You can also use an Arduino Zero, but you'll have to use the ICSP 6-pin header not pins 11,12,13

- **Vin** - connect this to 3.3V or 5V, whichever is the logic voltage of the microcontroller you're using. For UNO this will be 5V, for Zero its 3.3V
- **GND** - connect to common ground
- **SCK** - Connect to SPI clock. On UNO this is pin **#13** on Zero its on the 6-pin ISP header (http://adafru.it/iCE)
- **MISO** - Connect to SPI MISO. On UNO this is pin **#12** on Zero its on the 6-pin ISP header (http://adafru.it/iCE)
- **MOSI** - Connect to SPI MOSI. On UNO this is pin **#11** on Zero its on the 6-pin ISP header (http://adafru.it/iCE)

For the remaining pins, you can be a little flexible:

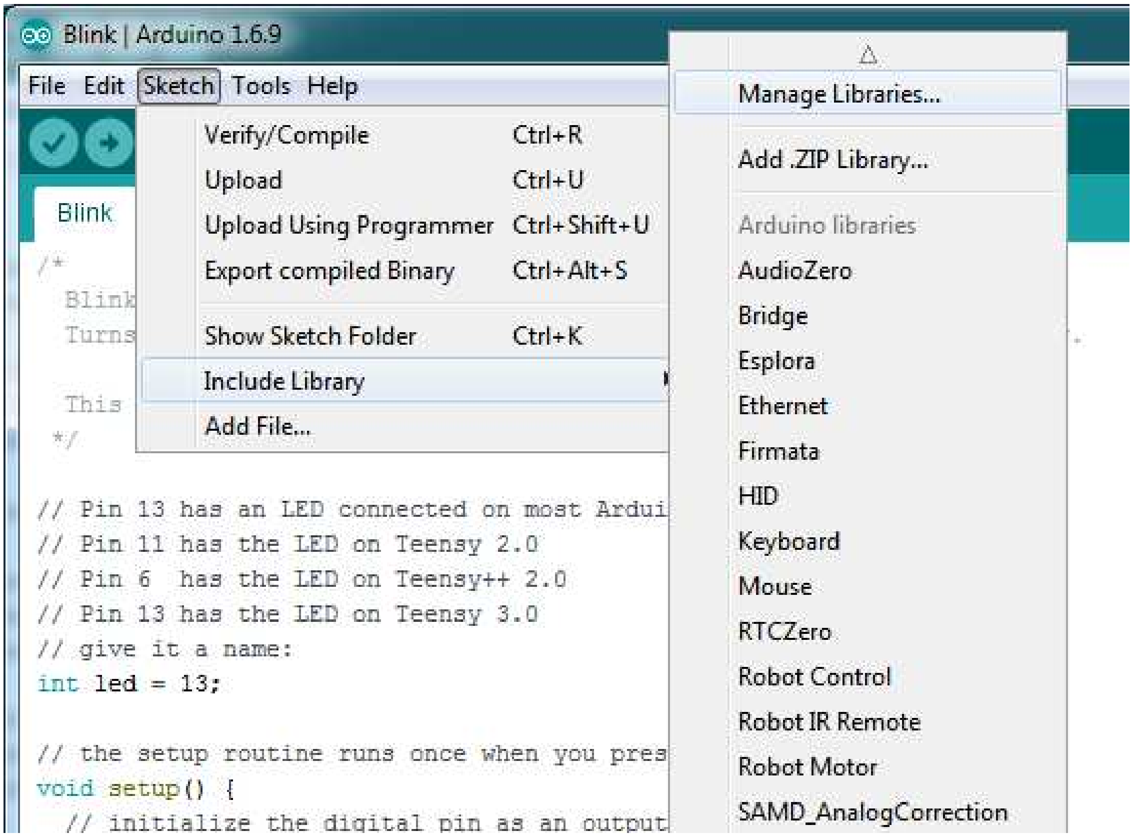
- **CS** - Connect to any digital I/O pin, we use **#8** by default
- **EN** - connect this to 3.3V or 5V, whichever is the logic voltage of the microcontroller you're using. For UNO this will be 5V, for Zero its 3.3V; later on if you want to enable/disable the module, connect it to a digial I/O pin
- **IRQ** - Connect to any digital I/O pin, preferrably one with an interrupt capability. We use **#7** by default
- **RST** - Connect to any digital I/O pin. We use **#4** by default

You can change these pins later but for now use them, so you can verify your setup!

Install the Library

We will start by installing the official Arduino WiFi101 library (http://adafru.it/kUF).

We want the latest version so visit the Library Manager



```
/*
  Blink
  Turns

  This
 */

// Pin 13 has an LED connected on most Ardui
// Pin 11 has the LED on Teensy 2.0
// Pin 6  has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

// the setup routine runs once when you pres
void setup() {
  // initialize the digital pin as an output
```

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 (http://adafru.it/aYM)!

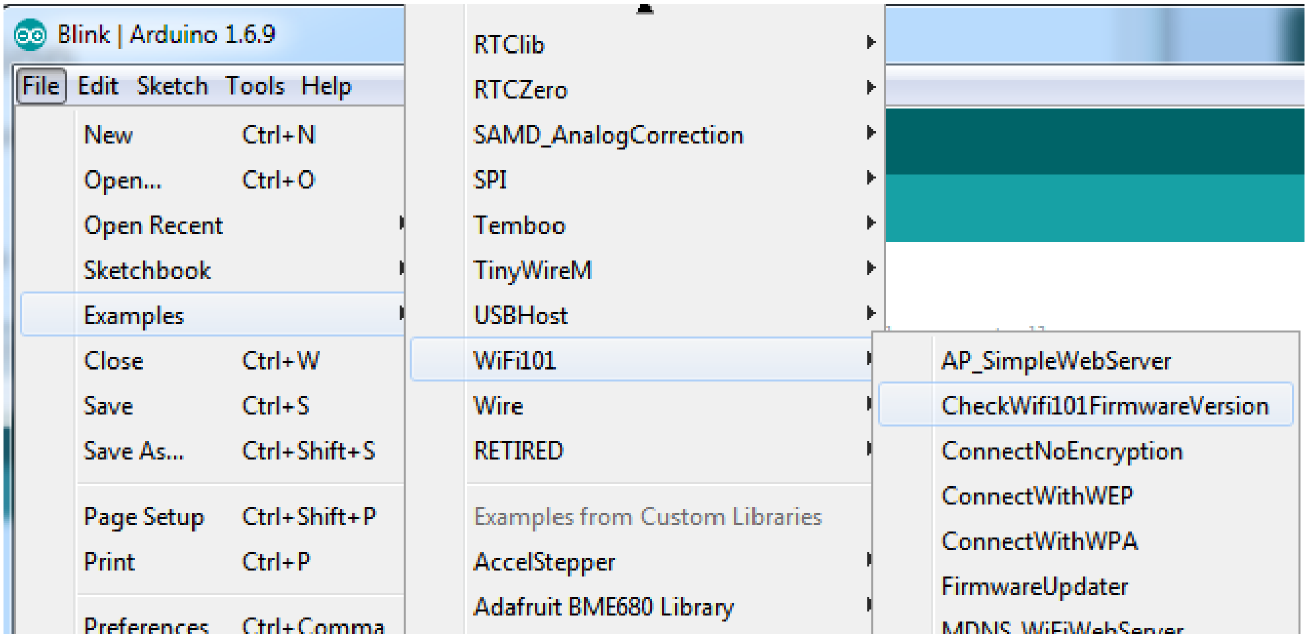
Restart the Arduino IDE.

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.

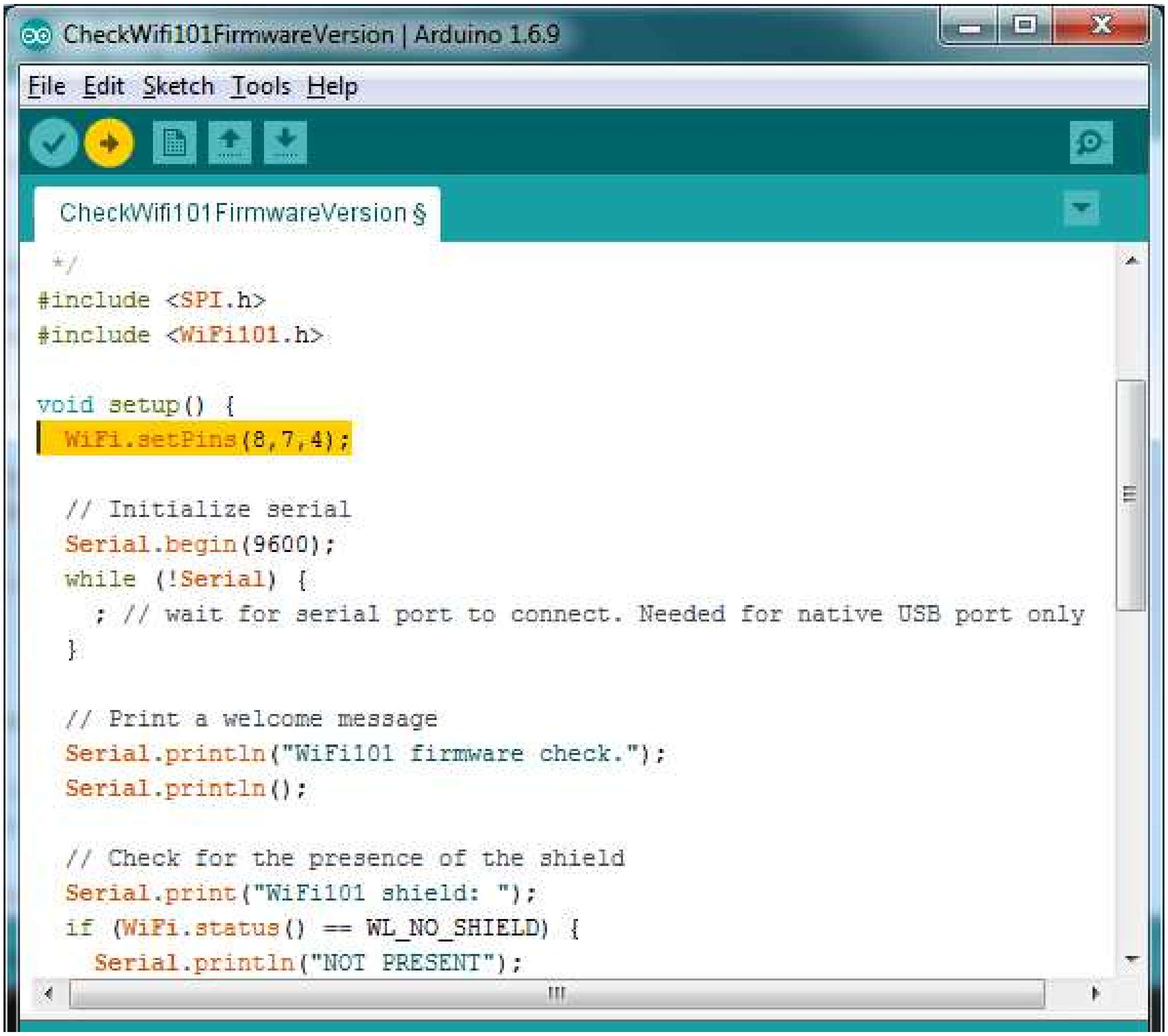
Load up the **WiFi101->CheckWifi101Firmware** sketch



Note that to use the official Arduino WiFi101 Library, we must configure it to use the pins specific to the ATWINC1500 Breakout. With each example sketch, you'll need to add WiFi.setPins(8,7,4); to the top of the setup function, before WiFi.status() is called.

```
//Configure pins for Adafruit ATWINC1500 Breakout
WiFi.setPins(8,7,4);
```

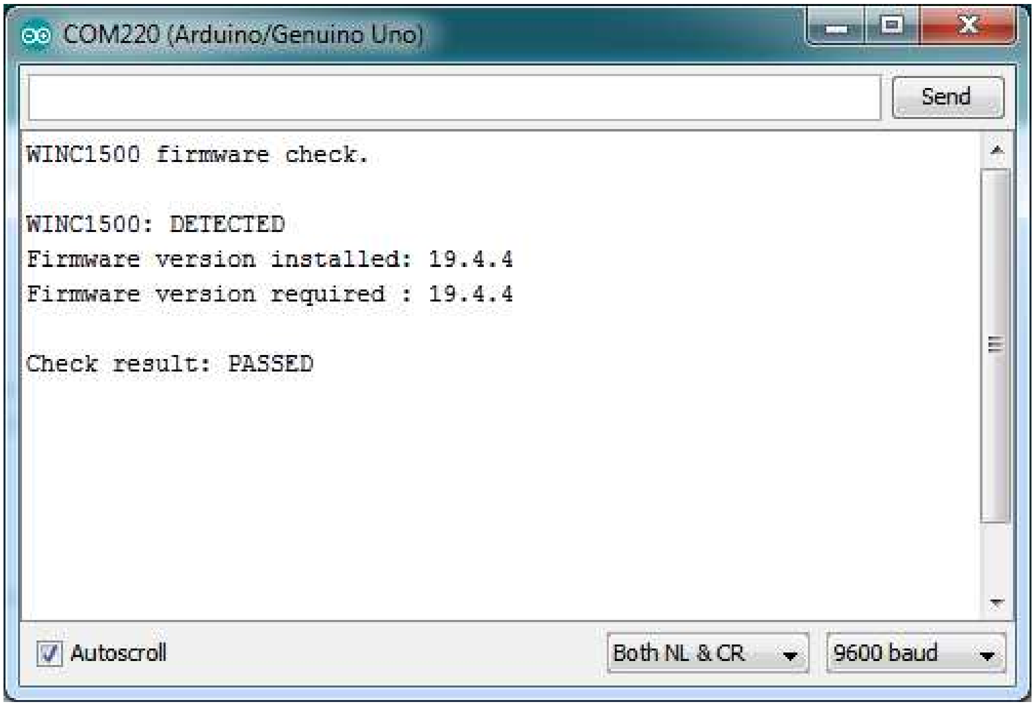
Like so:



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");
```

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

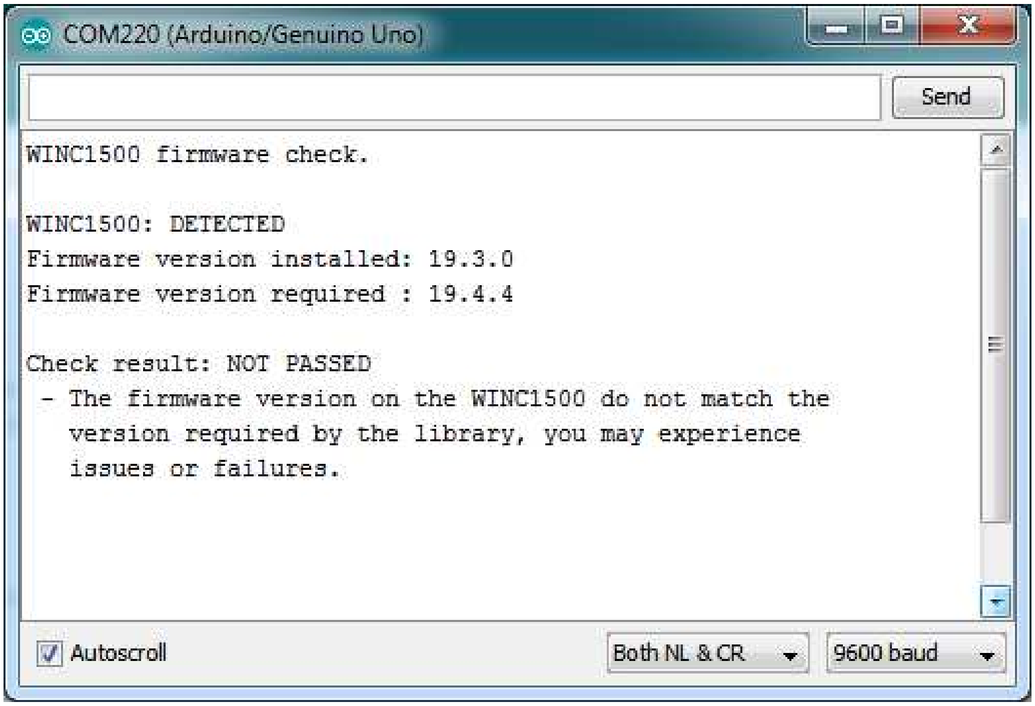


```
WINC1500 firmware check.

WINC1500: DETECTED
Firmware version installed: 19.4.4
Firmware version required : 19.4.4

Check result: PASSED
```

You should see the firmware version. If your version has notPASSED follow this page to update your firmware! (http://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 waaay to old, or something is amiss with your wiring!



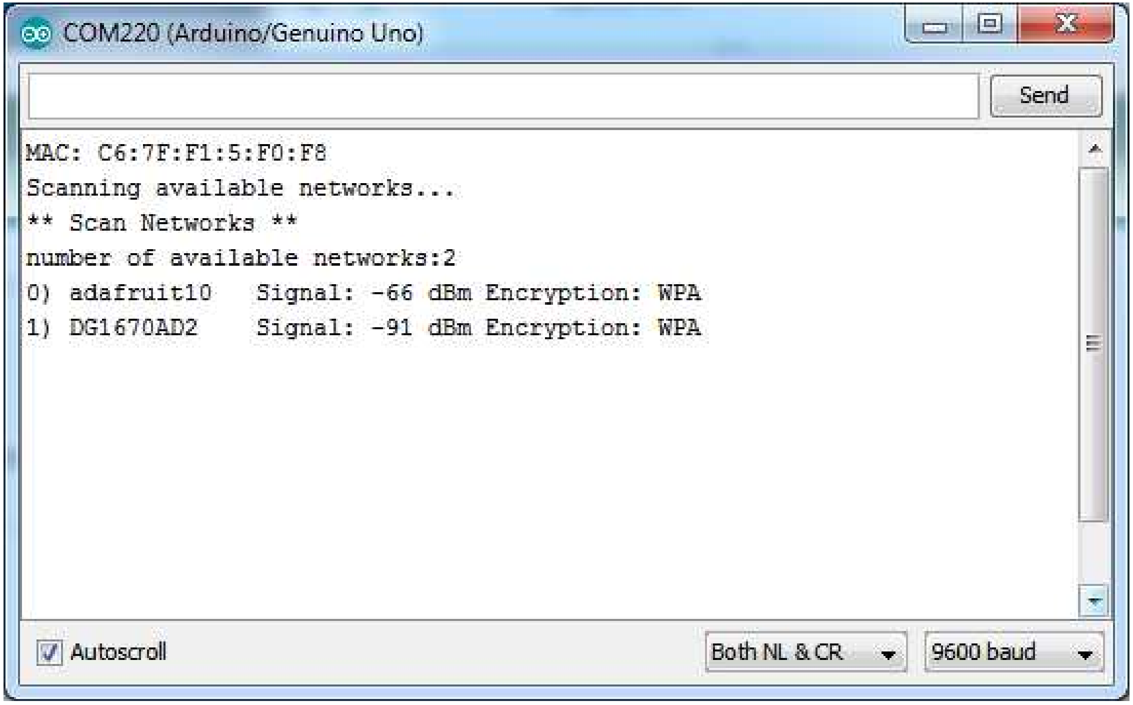
```
WINC1500 firmware check.

WINC1500: DETECTED
Firmware version installed: 19.3.0
Firmware version required : 19.4.4

Check result: NOT PASSED
 - The firmware version on the WINC1500 do not match the
   version required by the library, you may experience
   issues or failures.
```

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

Don't forget to add WiFi.setPins(8,7,4) at the top of setup()

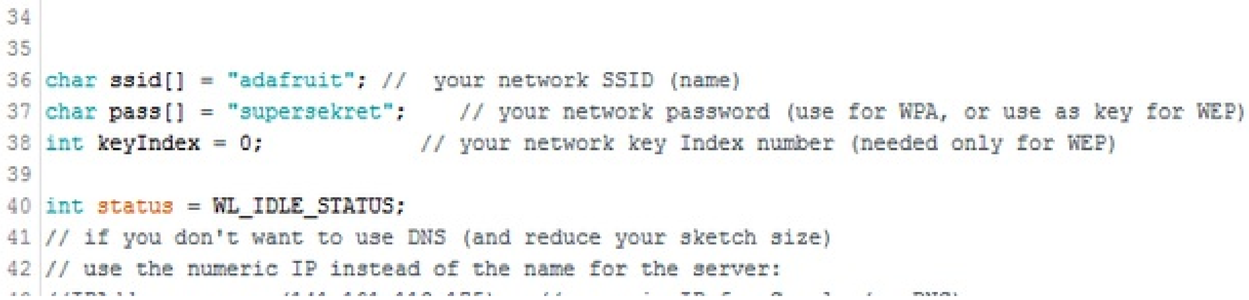


```
MAC: C6:7F:F1:5:F0:F8
Scanning available networks...
** Scan Networks **
number of available networks:2
0) adafruit10   Signal: -66 dBm Encryption: WPA
1) DG1670AD2    Signal: -91 dBm Encryption: WPA
```

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

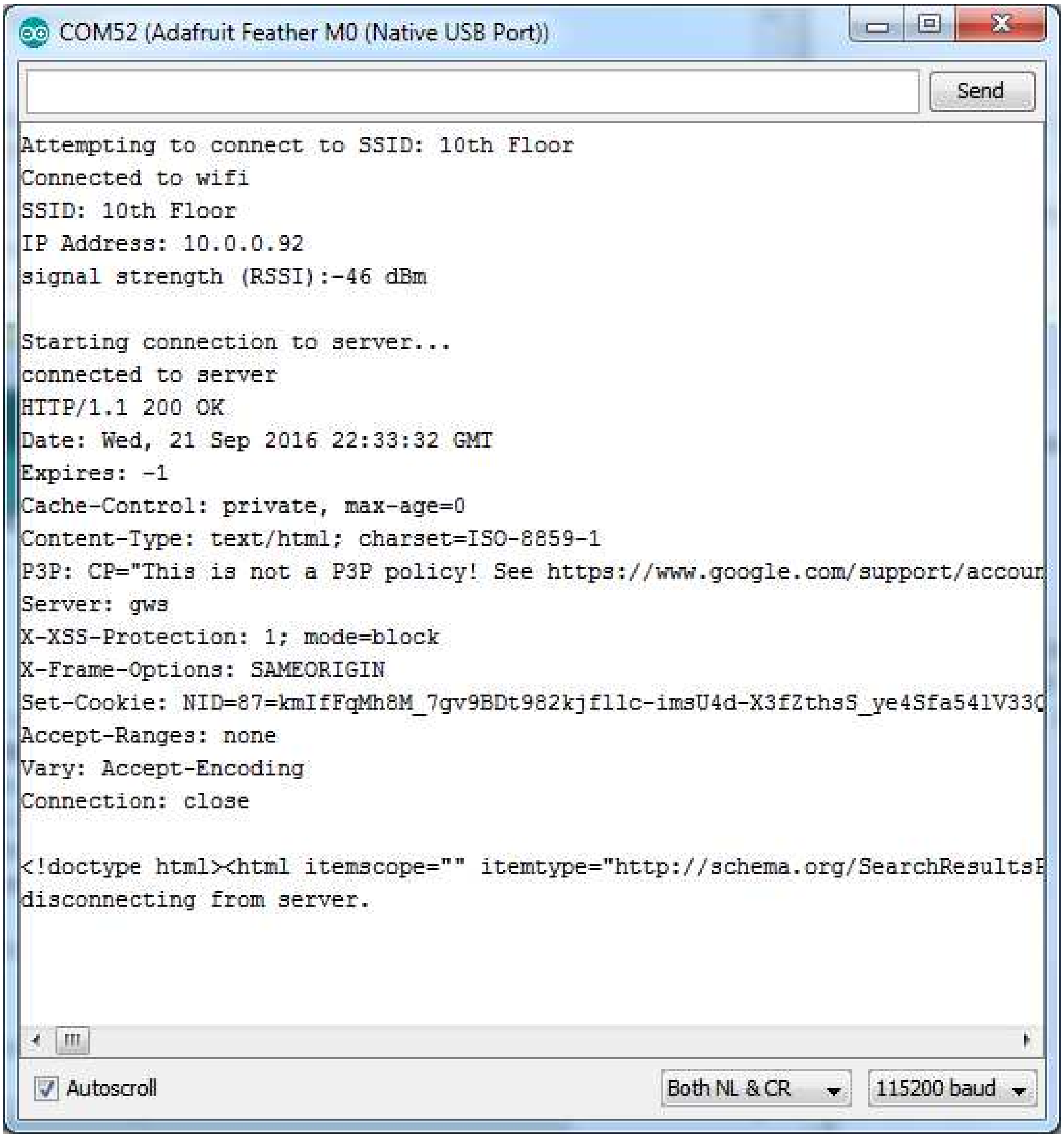


```


char ssid[] = "adafruit"; //  your network SSID (name)
char pass[] = "supersekret";    // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;             // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
```

Add the following lines at the top of setup()

```
//Configure pins for Adafruit ATWINC1500 Feather
WiFi.setPins(8,7,4);
```

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

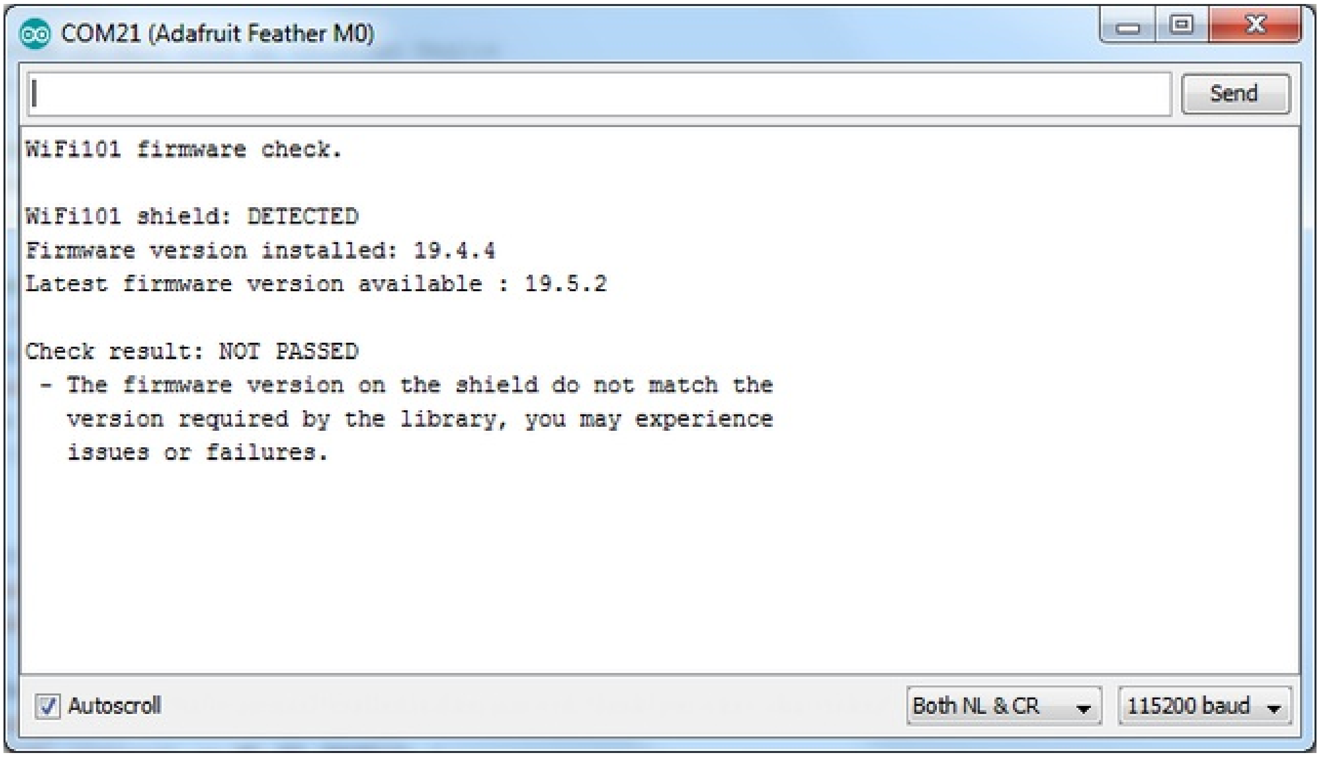


```
Attempting to connect to SSID: 10th Floor
Connected to wifi
SSID: 10th Floor
IP Address: 10.0.0.92
signal strength (RSSI):-46 dBm

Starting connection to server...
connected to server
HTTP/1.1 200 OK
Date: Wed, 21 Sep 2016 22:33:32 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See https://www.google.com/support/accoun
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: NID=87=kmIfFqMh8M_7gv9BDt982kjfllc-imsU4d-X3fZthsS_ye4Sfa541V33
Accept-Ranges: none
Vary: Accept-Encoding
Connection: close

<!doctype html><html itemscope="" itemtype="http://schema.org/SearchResults
disconnecting from server.
```

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

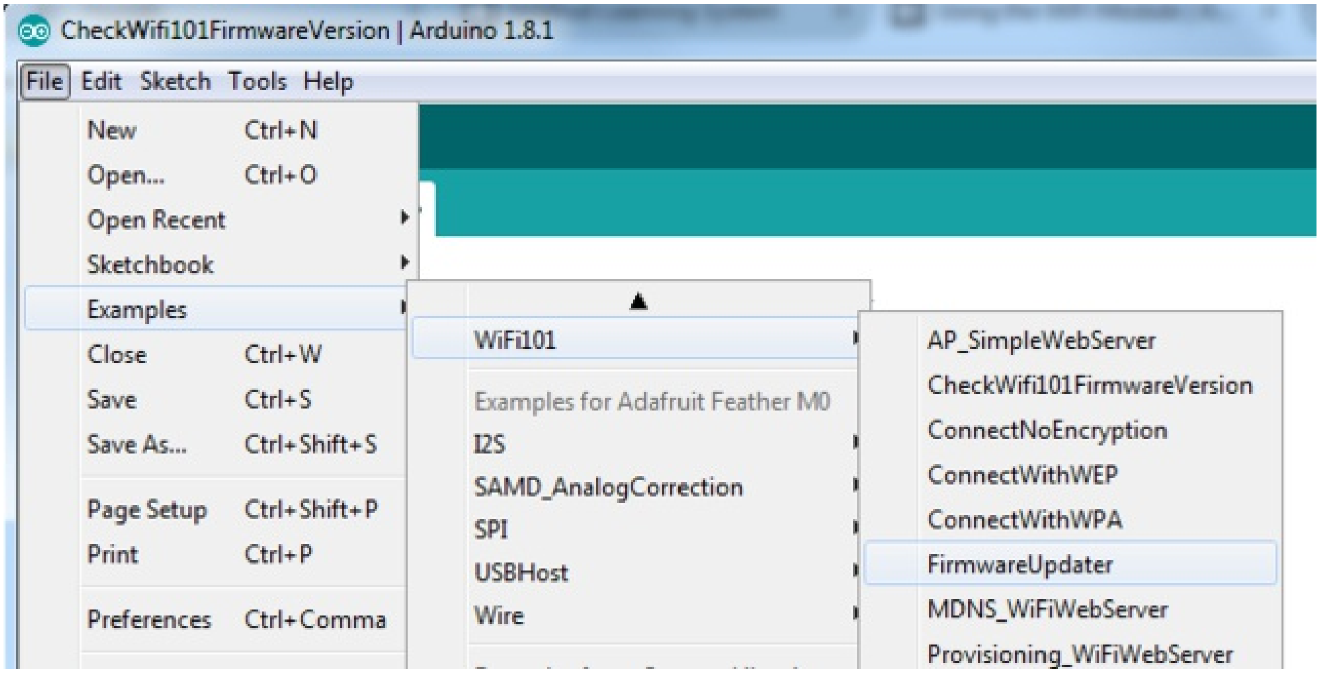
Updating Firmware

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:

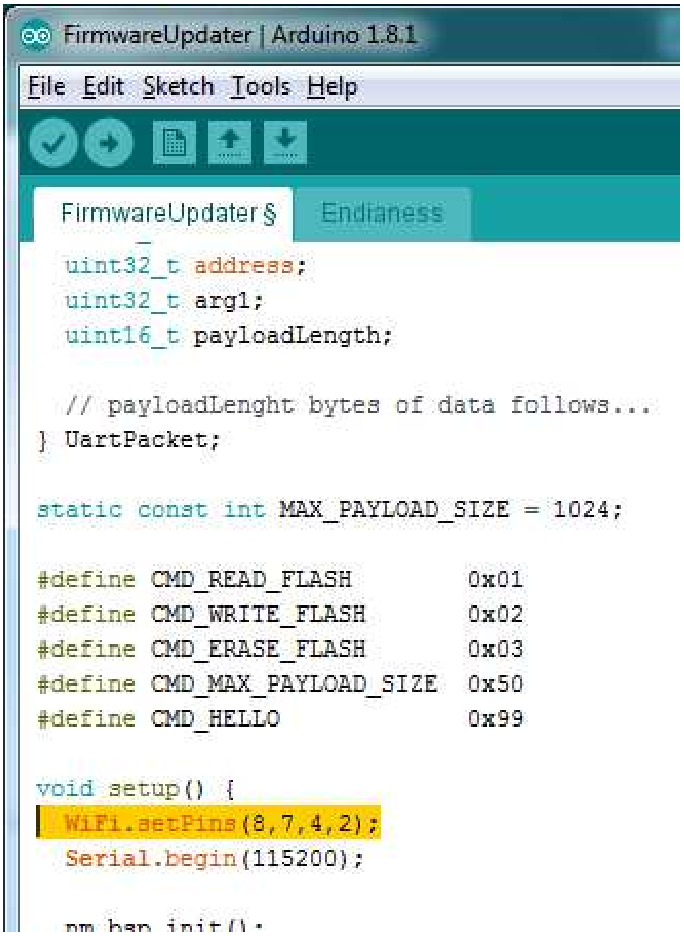


```
WiFi101 firmware check.

WiFi101 shield: DETECTED
Firmware version installed: 19.4.4
Latest firmware version available : 19.5.2

Check result: NOT PASSED
 - The firmware version on the shield do not match the
   version required by the library, you may experience
   issues or failures.
```

No problem - you can update the firmware through your Feather or Zero! Start by loading up the **FirmwareUpdater** sketch



update the pins as necessary, we have the default for use with the Feather M0 or WINC1500 breakout

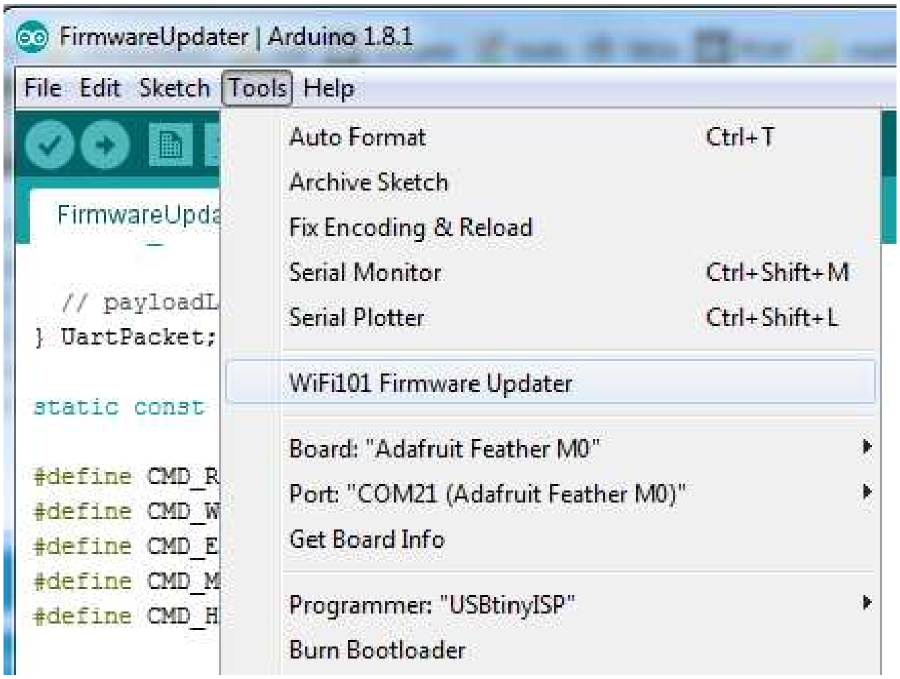


```
  uint32_t address;
  uint32_t arg1;
  uint16_t payloadLength;

  // payloadLenght bytes of data follows...
} UartPacket;

static const int MAX_PAYLOAD_SIZE = 1024;

#define CMD_READ_FLASH        0x01
#define CMD_WRITE_FLASH       0x02
#define CMD_ERASE_FLASH       0x03
#define CMD_MAX_PAYLOAD_SIZE  0x50
#define CMD_HELLO             0x99

void setup() {
  WiFi.setPins(8,7,4,2);
  Serial.begin(115200);

  nm_bsp_init();
```

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

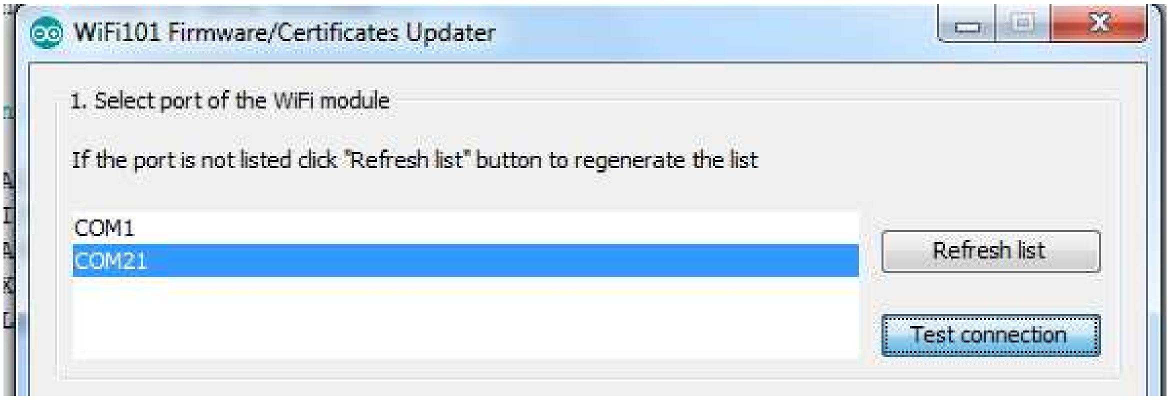
Upload it to your board. Make sure the Serial console is not open before or after uploading.

Then select the Updater tool built into the IDE


FirmwareUpdater | Arduino 1.8.1
File Edit Sketch Tools Help
FirmwareUpda
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Adafruit Feather M0"
Port: "COM21 (Adafruit Feather M0)"
Get Board Info
Programmer: "USBtinyISP"
Burn Bootloader

```
  // payloadL
} UartPacket;

static const

#define CMD_R
#define CMD_W
#define CMD_E
#define CMD_M
#define CMD_H
```

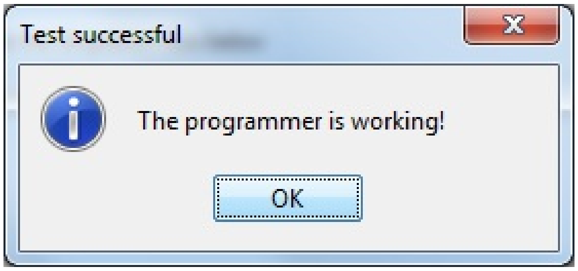
Select the right COM port, and click Test Connection



1. Select port of the WiFi module

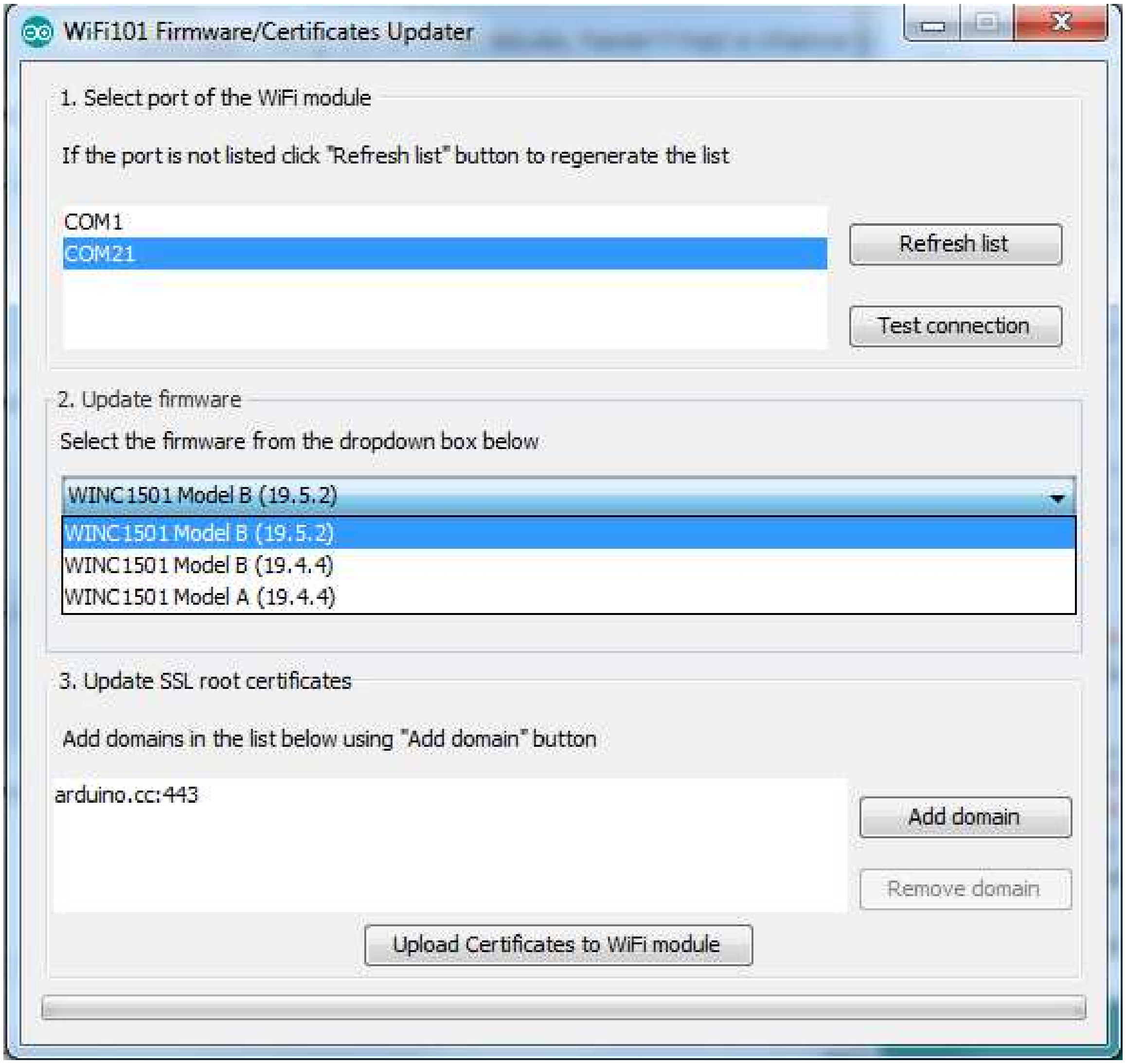
If the port is not listed click "Refresh list" button to regenerate the list

If all is good you'll get a confirmation



The programmer is working!

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



WiFi101 Firmware/Certificates Updater

1. Select port of the WiFi module

If the port is not listed click "Refresh list" button to regenerate the list

COM1
COM21

2. Update firmware

Select the firmware from the dropdown box below

WINC1501 Model B (19.5.2)

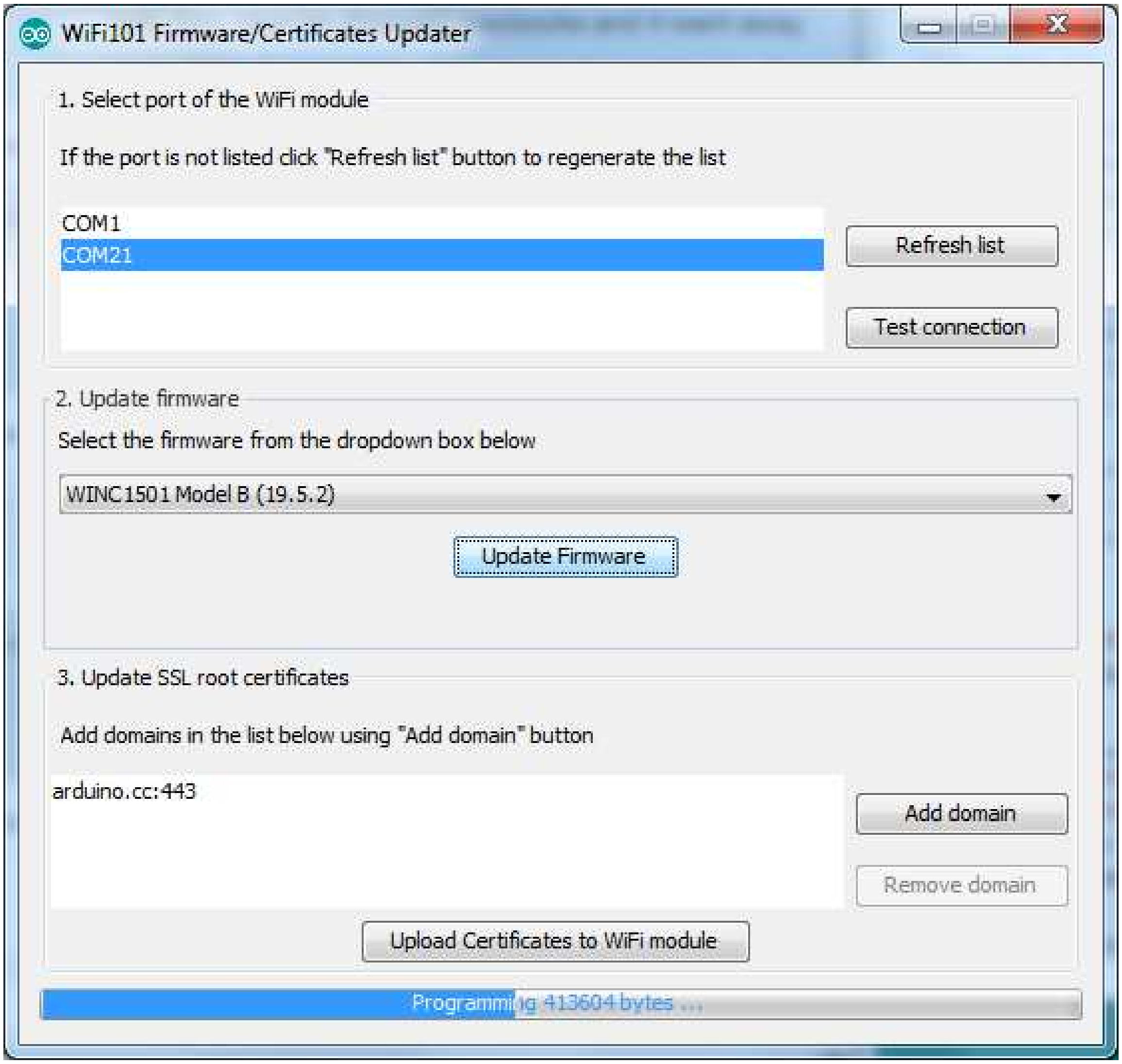
WINC1501 Model B (19.5.2)
WINC1501 Model B (19.4.4)
WINC1501 Model A (19.4.4)

3. Update SSL root certificates

Add domains in the list below using "Add domain" button

arduino.cc:443

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



WiFi101 Firmware/Certificates Updater

1. Select port of the WiFi module

If the port is not listed click "Refresh list" button to regenerate the list

COM1
COM21

2. Update firmware

Select the firmware from the dropdown box below

WINC1501 Model B (19.5.2)

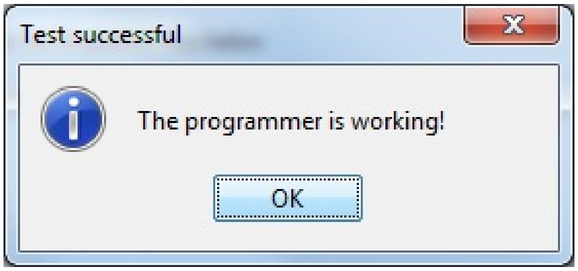
3. Update SSL root certificates

Add domains in the list below using "Add domain" button

arduino.cc:443

Programming 413604 bytes ...

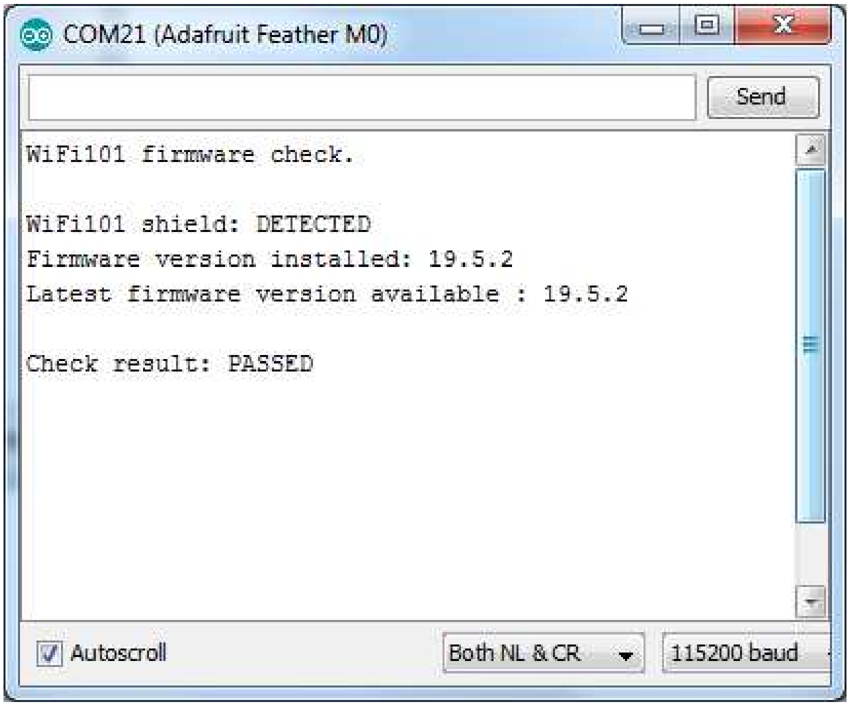
And a minute or two later...



Test successful

The programmer is working!

Now you're ready to rock!



```
WiFi101 firmware check.

WiFi101 shield: DETECTED
Firmware version installed: 19.5.2
Latest firmware version available : 19.5.2

Check result: PASSED
```