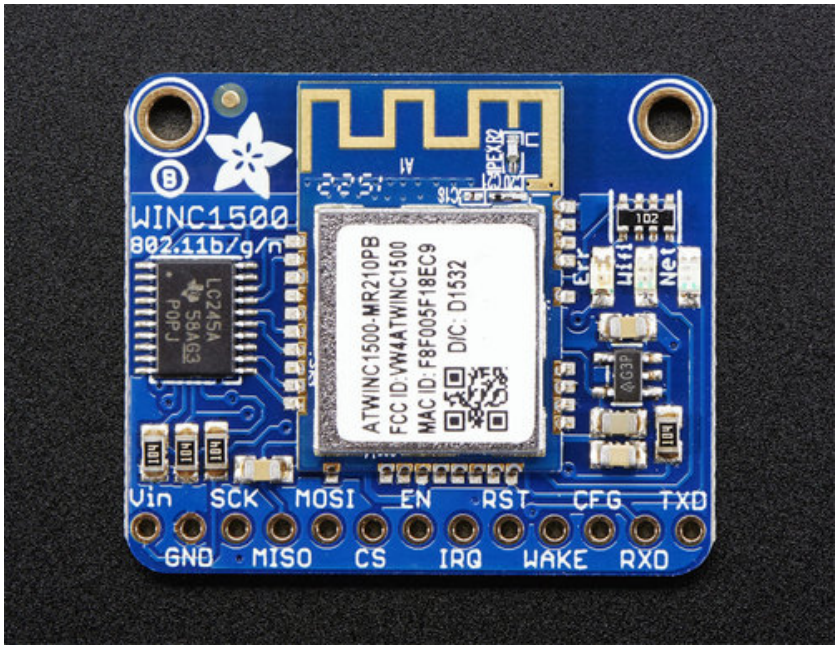
adafruit learning system

Adafruit ATWINC1500 WiFi Breakout

Created by lady ada

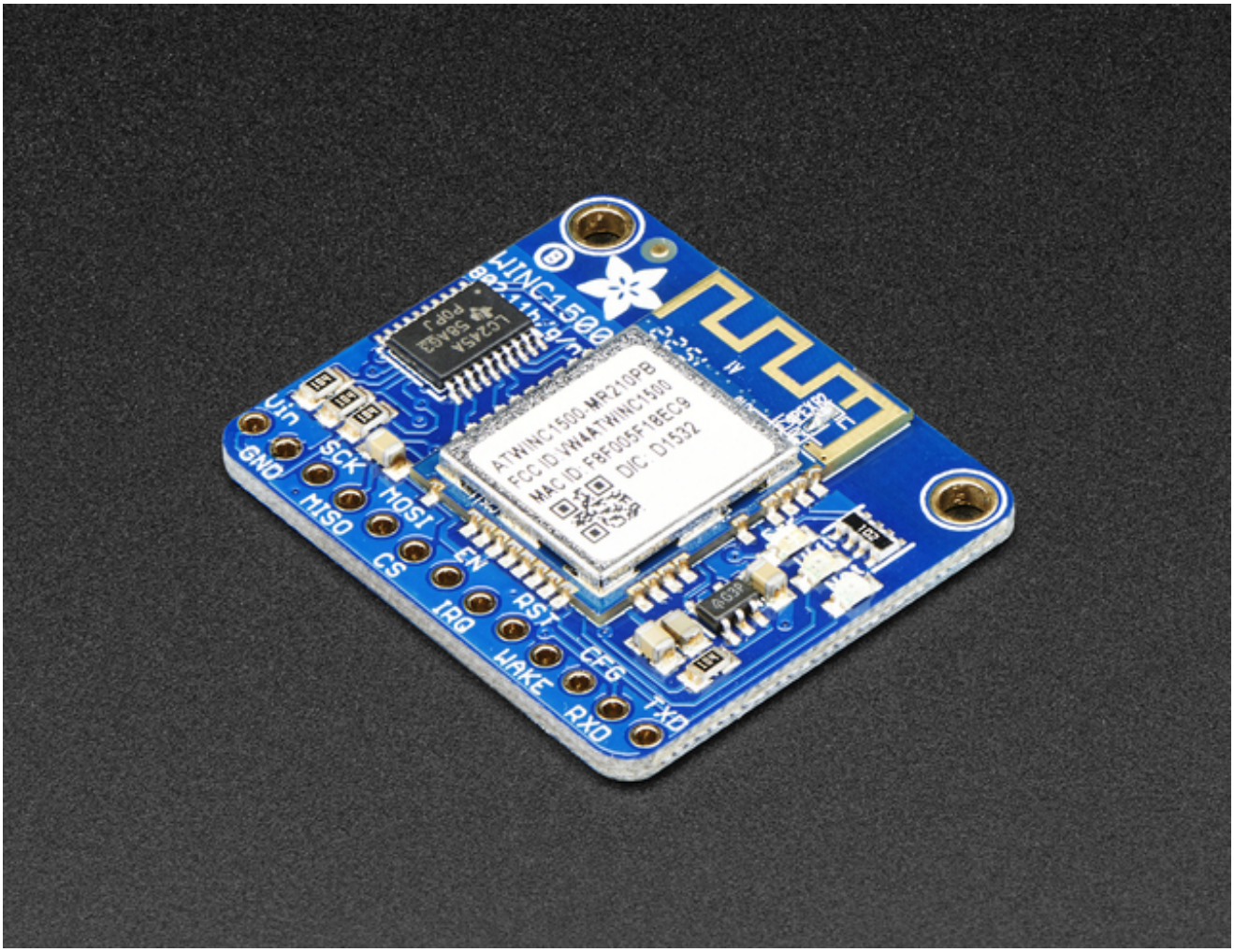


Last updated on 2016-03-09 12:29:56 PM EST

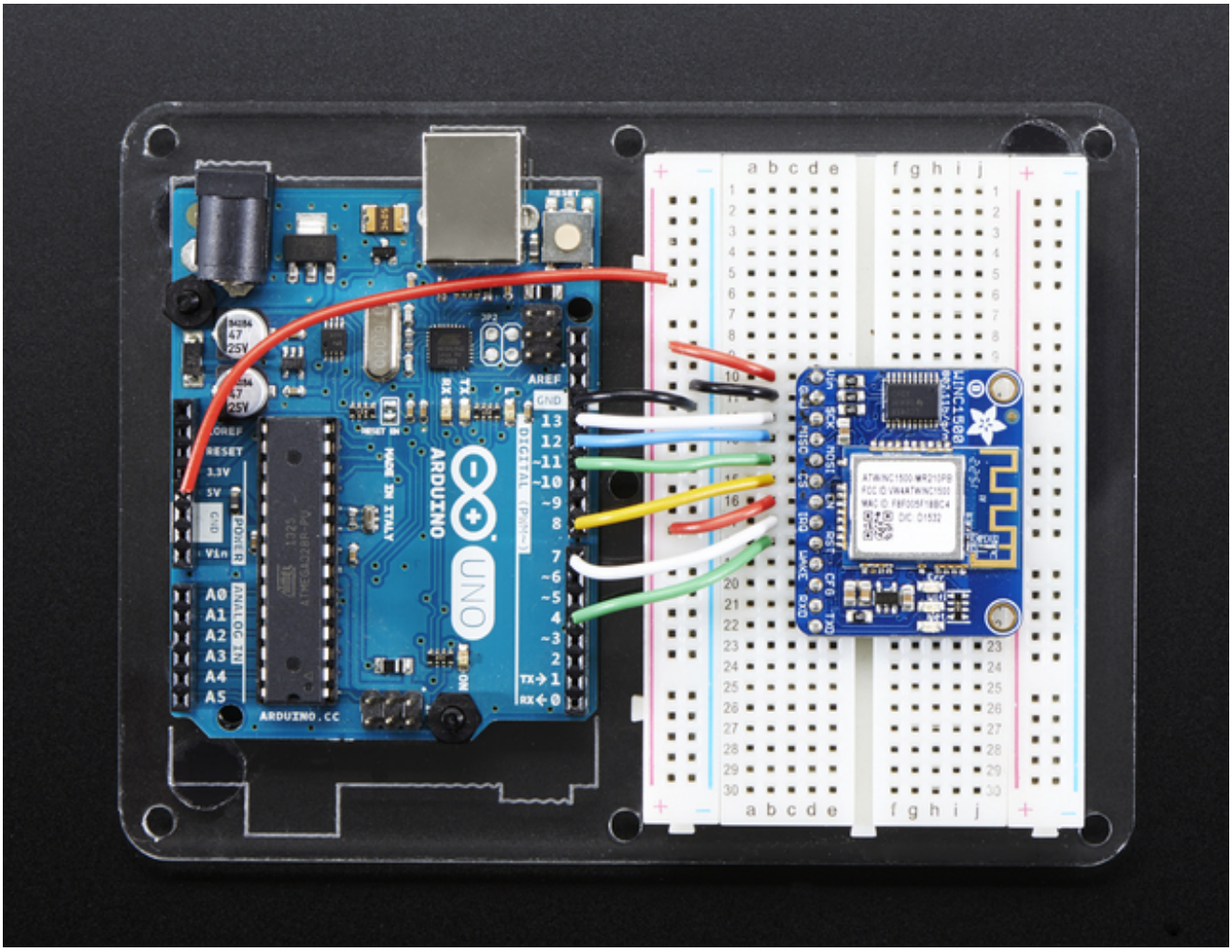
Guide Contents

Guide Contents 2
Overview 3
Pinouts 7
Power Pins 7
SPI Pins 7
Other SPI Interface Pins 8
Assembly 9
Prepare the header strip: 9
Add the breakout board: 9
And Solder! 10
Wiring & Test 12
Download the Adafruit Library 13
Check Connections & Version 13
Scanning WiFi 15
Connect & Read Webpage 15
Updating SSL Certificates 18
Command Line Usage 21
Windows 21
Mac OS X (Command Line) Usage 22
GUI Usage 22
Manually Adding Certificates 24
Certicate Format 29
Downloads 32
Datasheets and More 32
Schematic 32
Fabrication Print 32

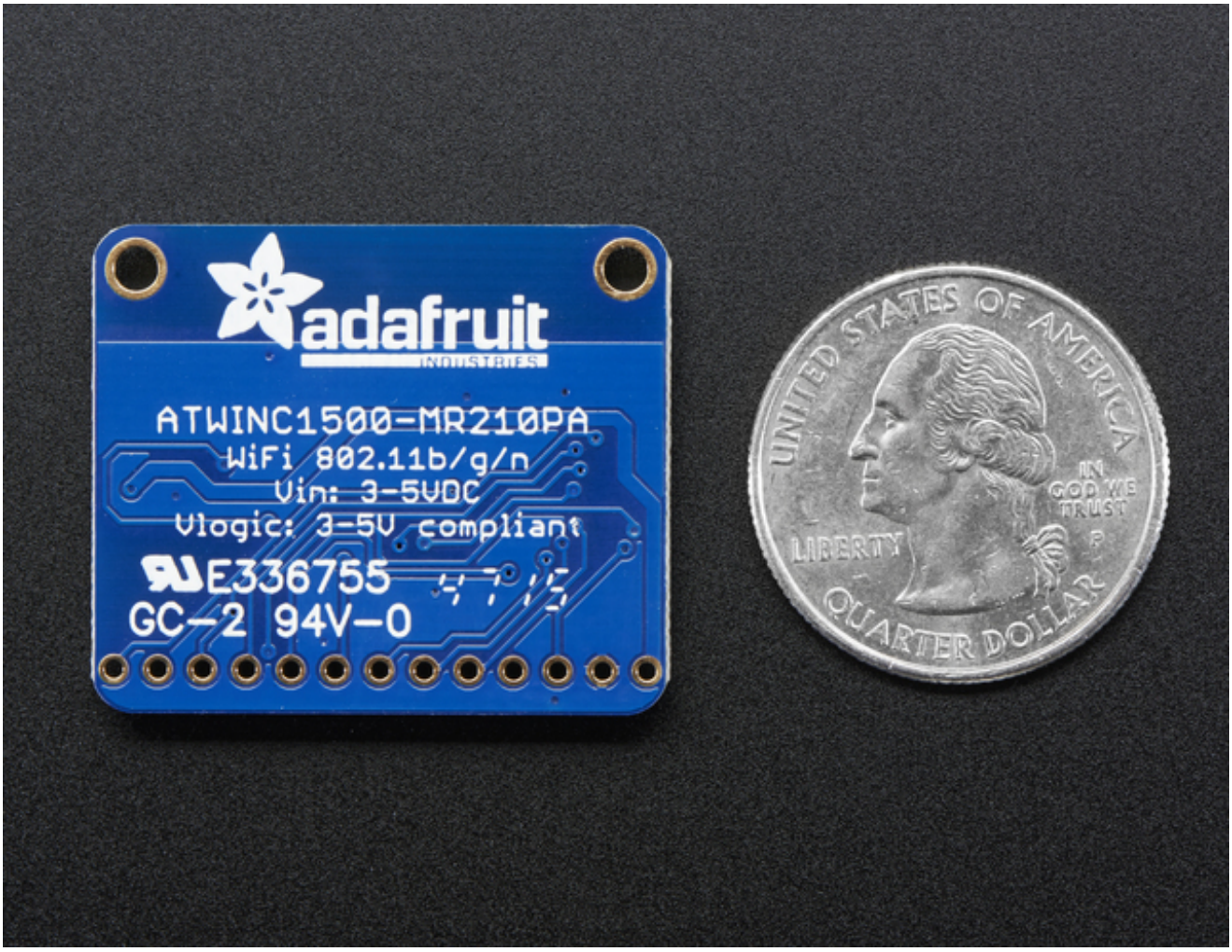
Overview



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.

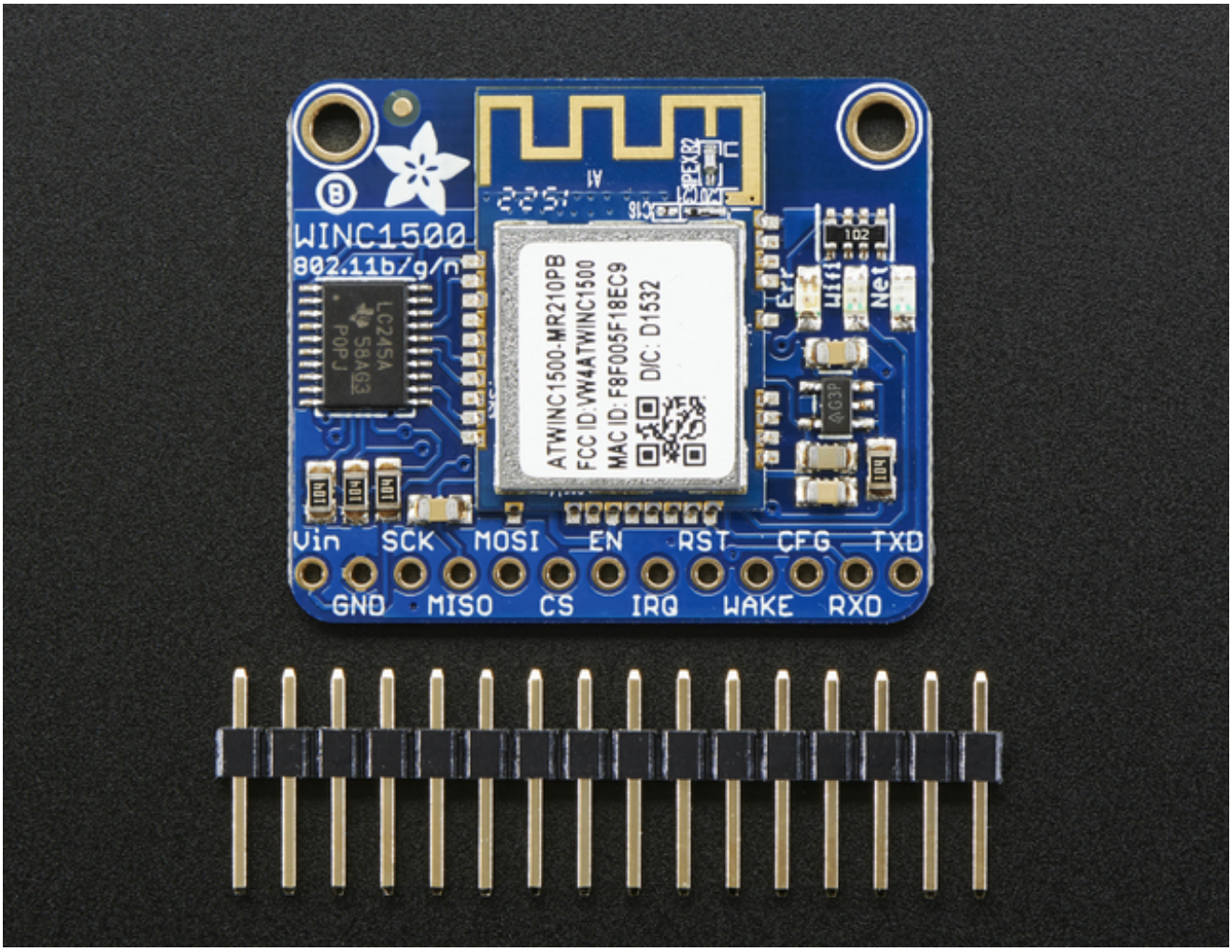


The ATWINC1500 uses SPI to communicate, so with about 5 or 6 wires, you can get your wired up and ready to go. Right now the Atmel-supplied library works great with Arduino Uno or Zero, but may not work on other Arduinos. You can clock it as fast as 12MHz for speedy, reliable packet streaming. Communication 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.



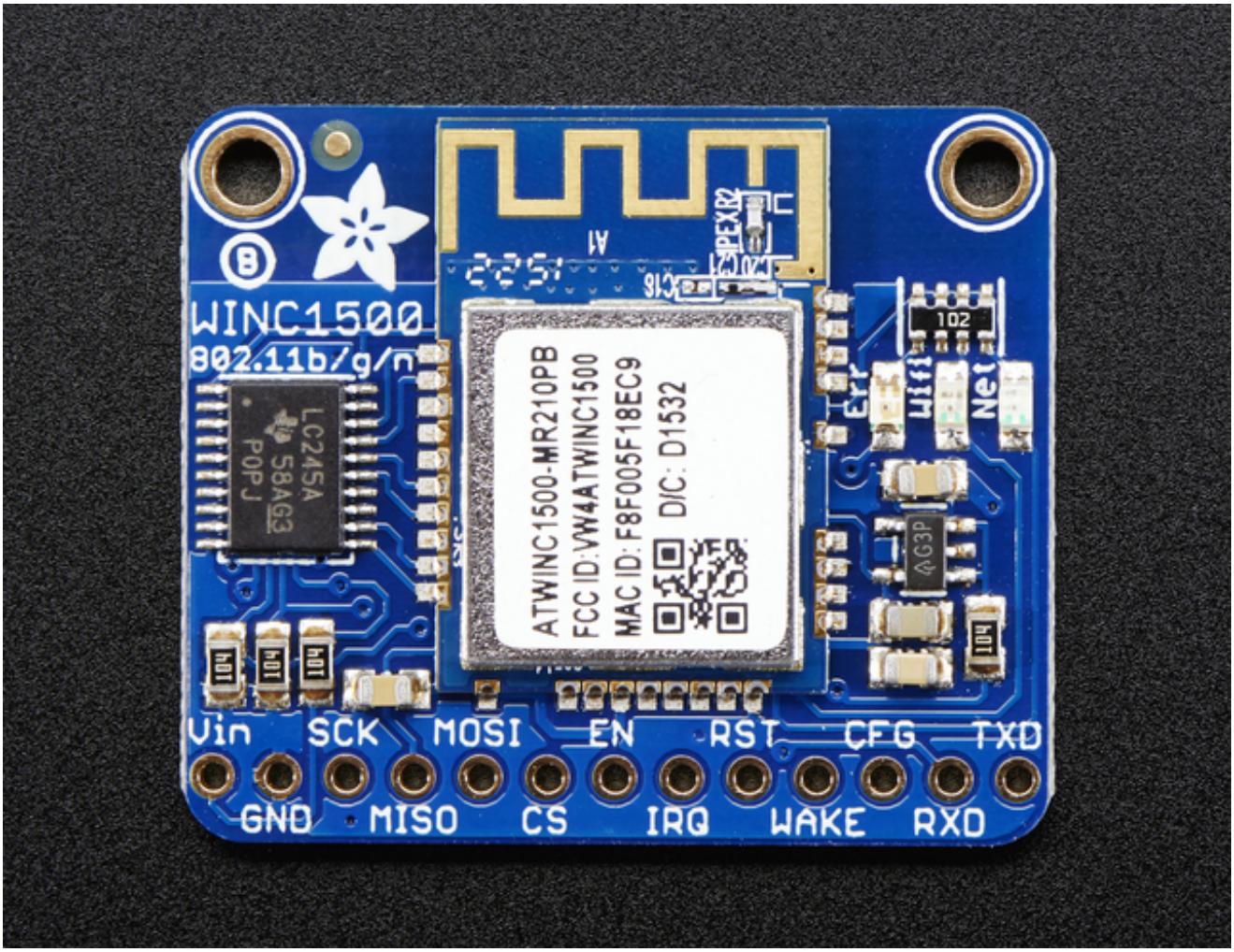
This module works with 802.11b, g, or n networks & supports WEP, WPA and WPA2 encryption. The datasheet says it can do Soft-AP mode but we don't have any code to actually use that.

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

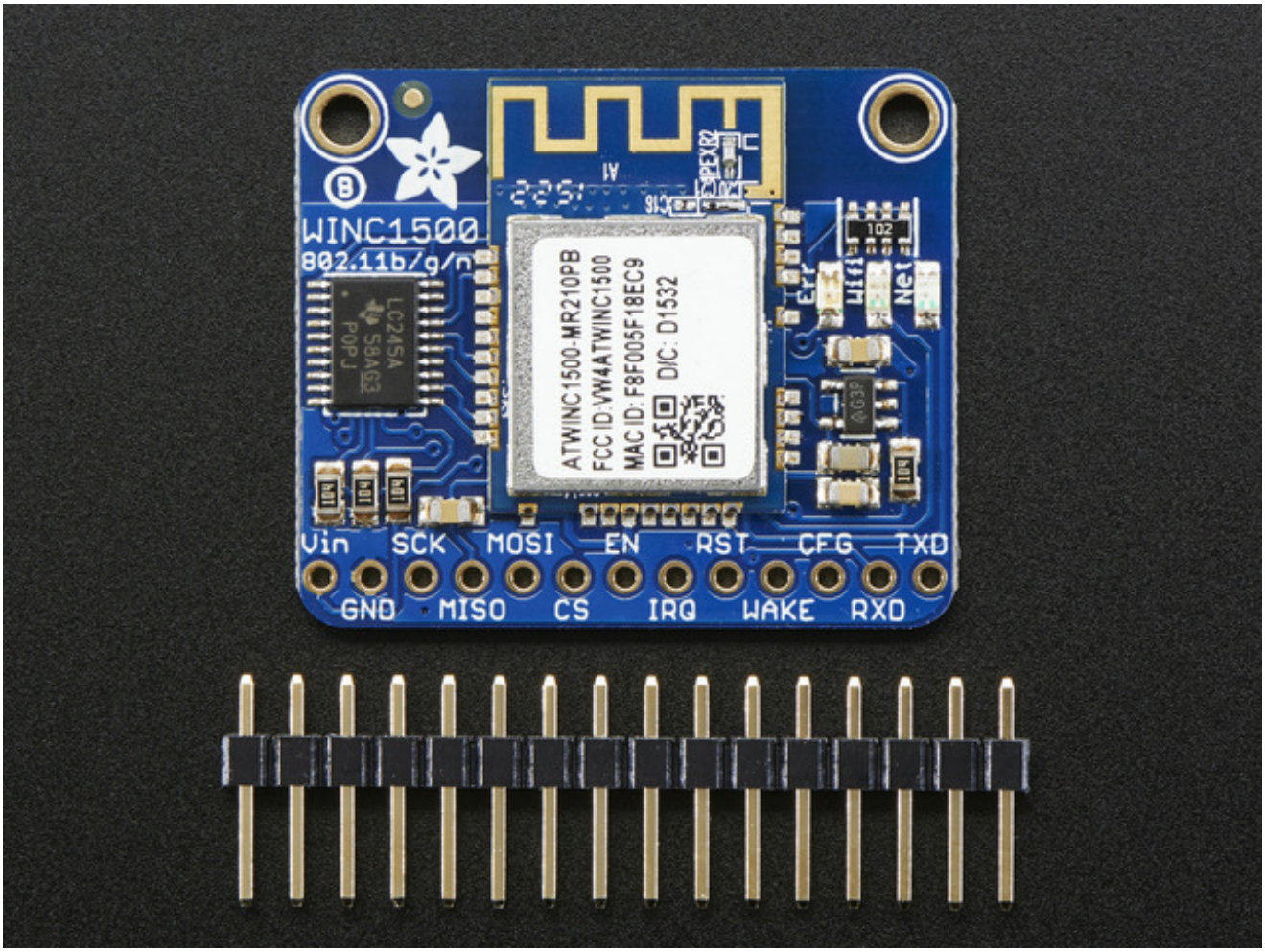
- **SCK** - SPI clock input, 3V or 5V compliant

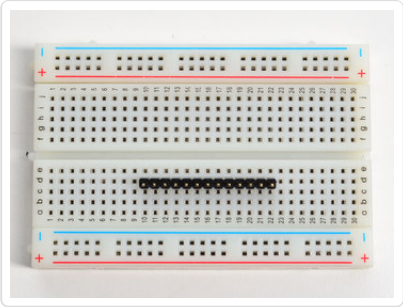
- **MISO** - SPI data out from module, 3.3V line level
- **MOSI** - SPI data into module, 3V or 5V compliant
- **CS** - SPI chip select, pull down when transmitting to/from the ATWINC. By default this is pulled to Vin with a 100K resistor

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

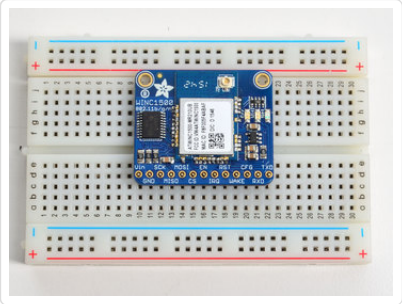
Assembly



- 

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**

- 

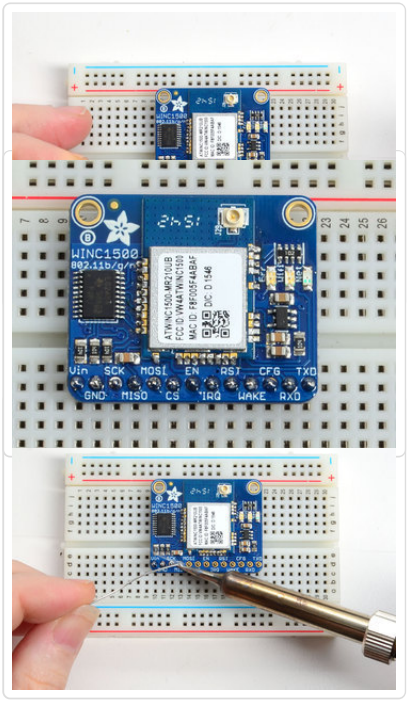
Add the breakout board:

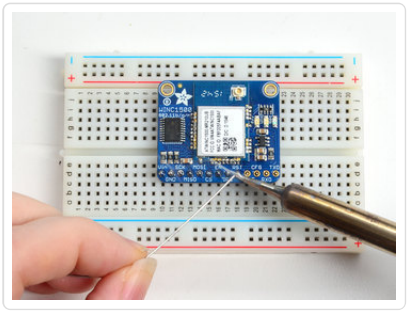
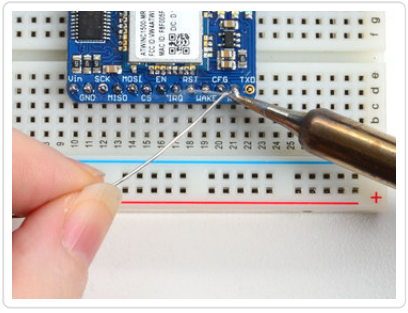
Place the breakout board over the pins so that the short pins poke through the breakout pads

And Solder!

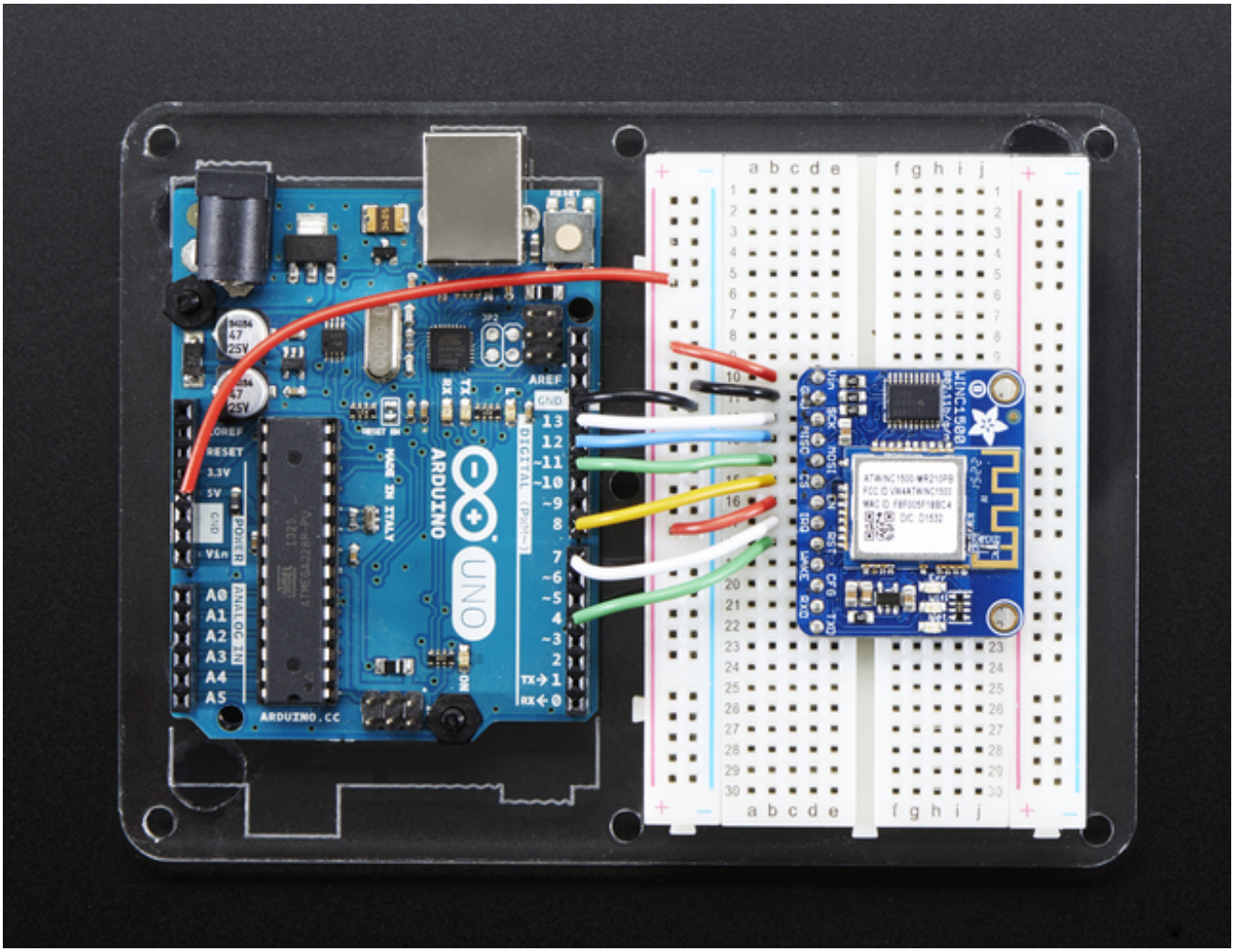
Be sure to solder all pins for reliable electrical contact.

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

- 

- 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

Download the Adafruit Library

We will start by downloading the Adafruit_ATWINC1500 Library, available from our GitHub repository (http://adafru.it/kUE).

This library is a light fork of the official Arduino Wifi101 library, (http://adafru.it/kUF)the only real changes are to allow changes to the default pin usage.

You can download the latest ZIP file by clicking the button below.

Download Adafruit_WINC1500 library

http://adafru.it/kVa

Rename the uncompressed folder **Adafruit_WINC1500**. Check that the **Adafruit_WINC1500** folder contains a folder named **src** and **examples** and a file named **library.properties**

Place the **Adafruit_WINC1500** library folder your ***sketchbookfolder*/libraries/** folder. You may need to create the libraries subfolder if its your first library. Restart the IDE. You can figure out your ***sketchbookfolder*** by opening up the Preferences tab in the Arduino IDE.

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

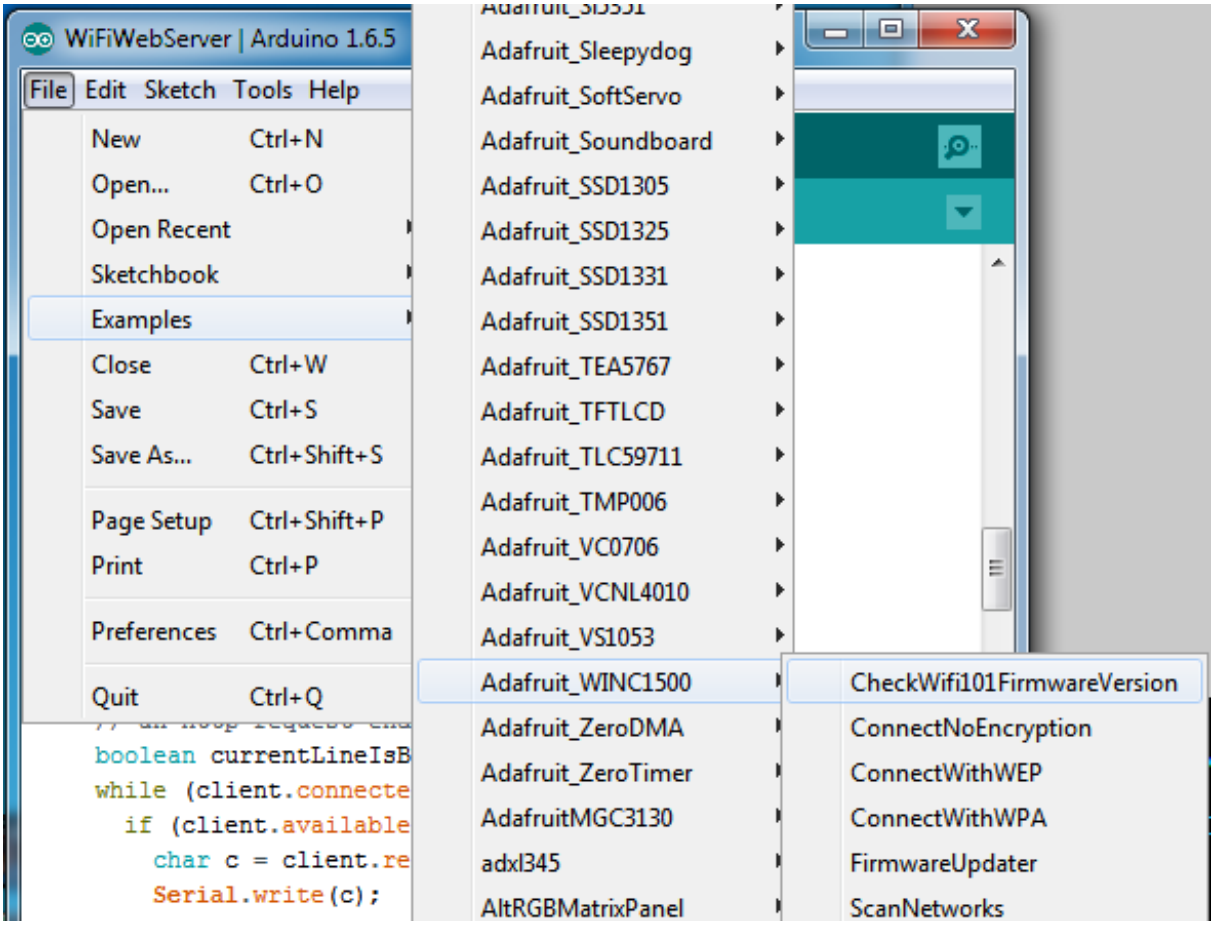
Restart the Arduino IDE.

You may need to use Arduino 1.6.5 or later

Check Connections & Version

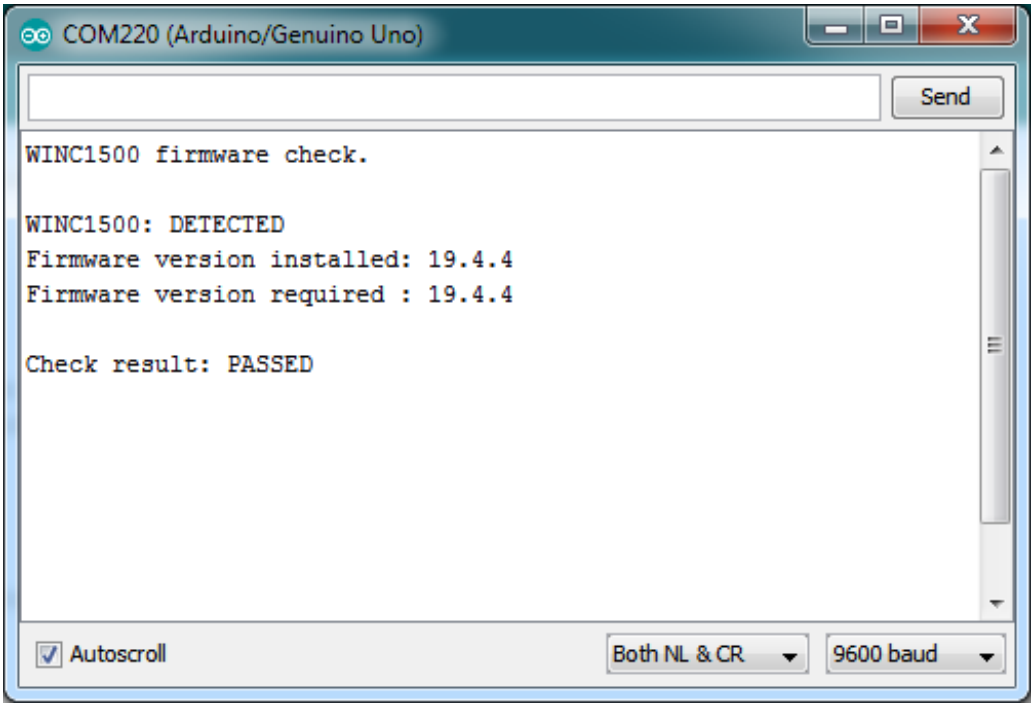
Before we start, its important to verify you have the right wiring & firmware version.

Load up the **Adafruit_WINC1500->CheckWifi101Firmware** sketch



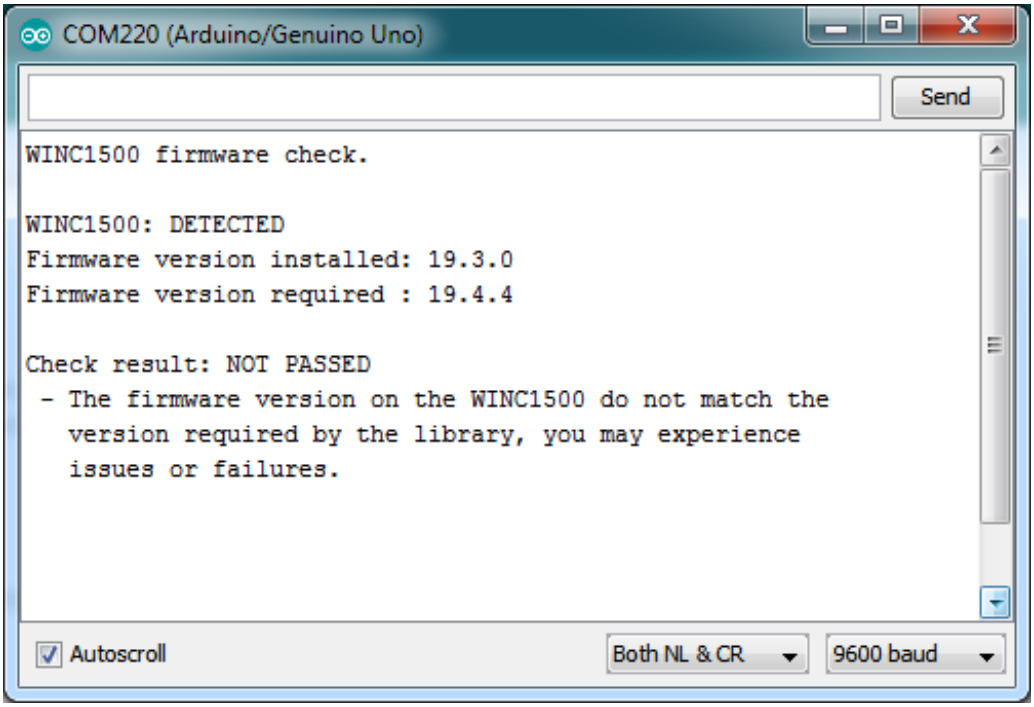
Upload to your Arduino and open up the Serial Console at 9600 baud:

You should see that the firmware is 19.4.4



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

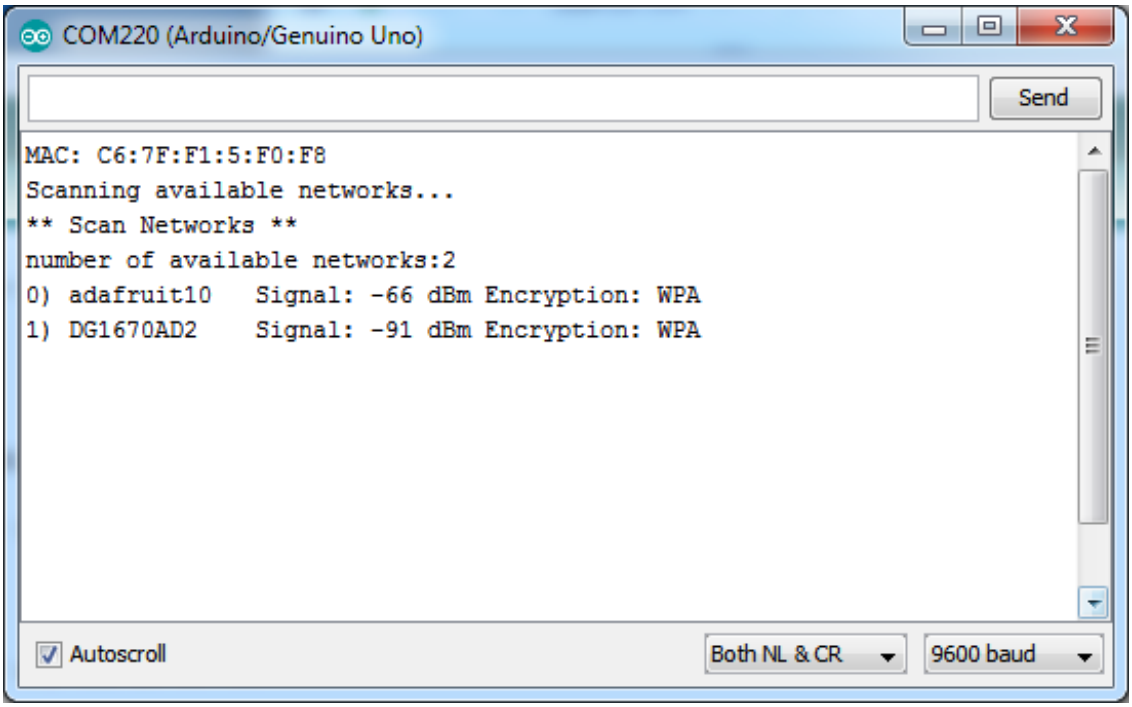
If you get not response, the firmware is either waaay to old, or something is amiss with your wiring!



Scanning WiFi

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

Run the **Adafruit_WINC1500->ScanNetworks** example to see a list of available visible networks



Connect & Read Webpage

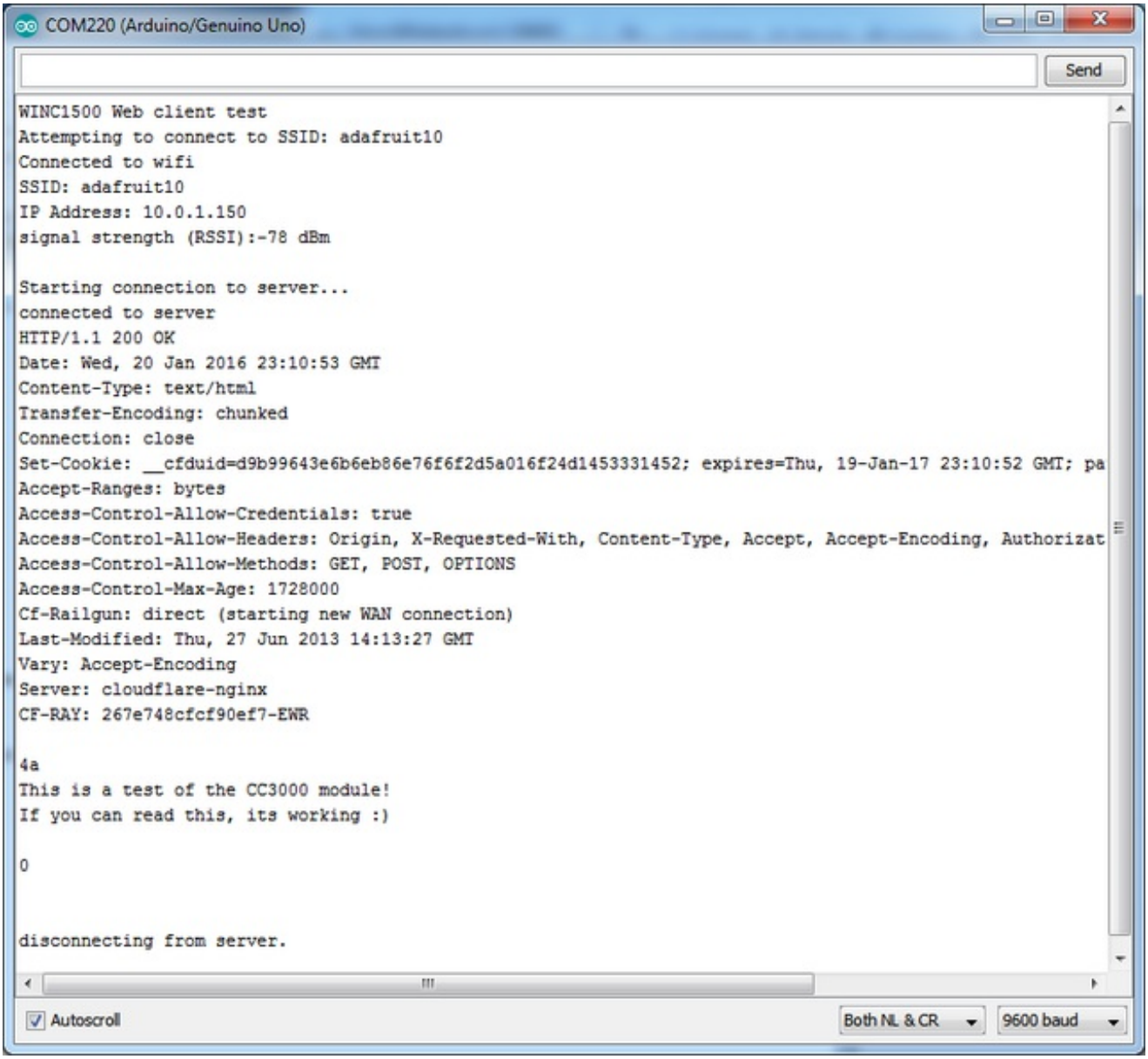
Connect & Read Webpage

OK finally you get to connect and read some data!

Open up the **Adafruit_WINC1500->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

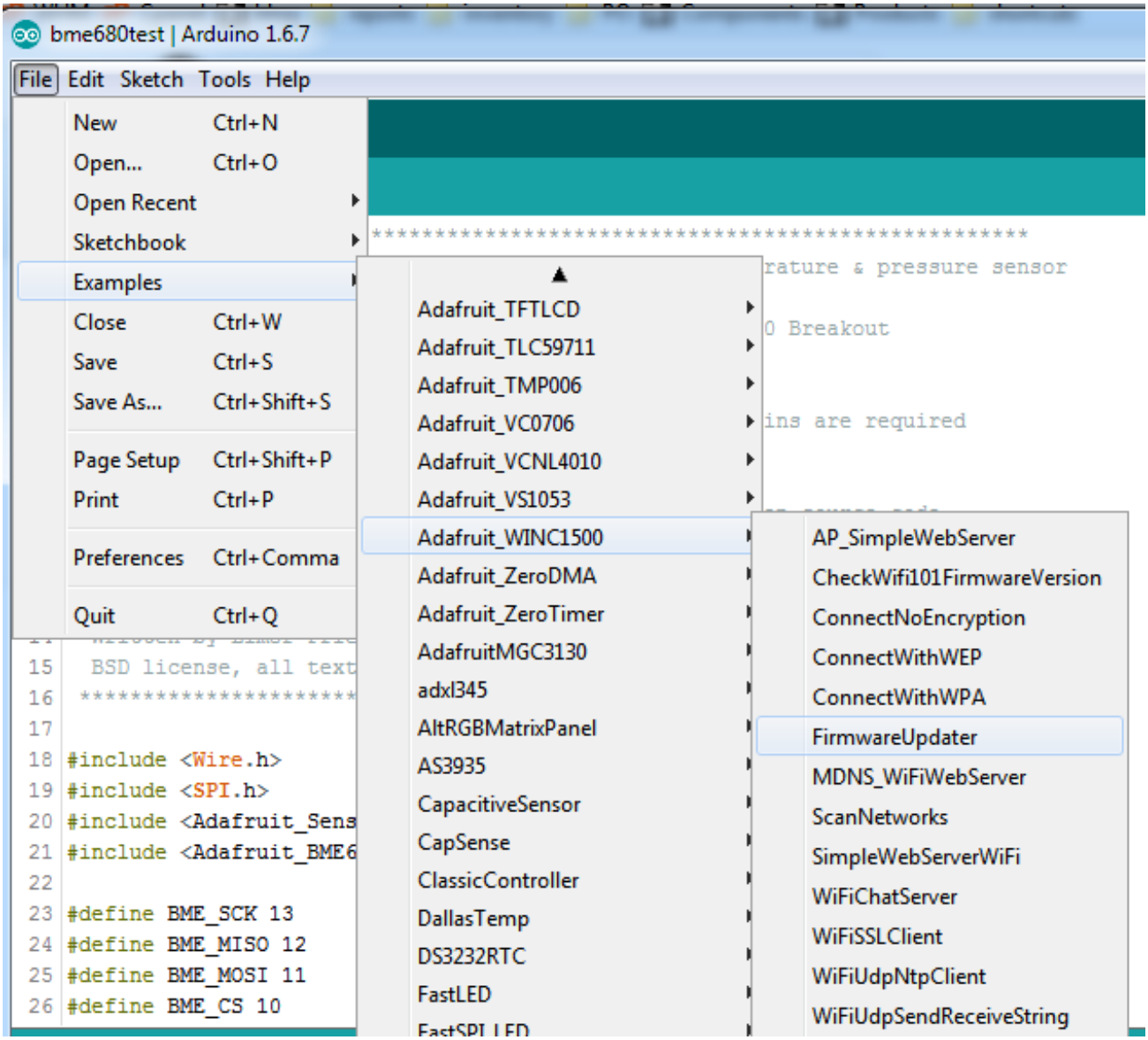
Updating SSL Certificates

> Do not use the updater to update the WINC1500 firmware, you could brick it. Only use it for updating SSL certs

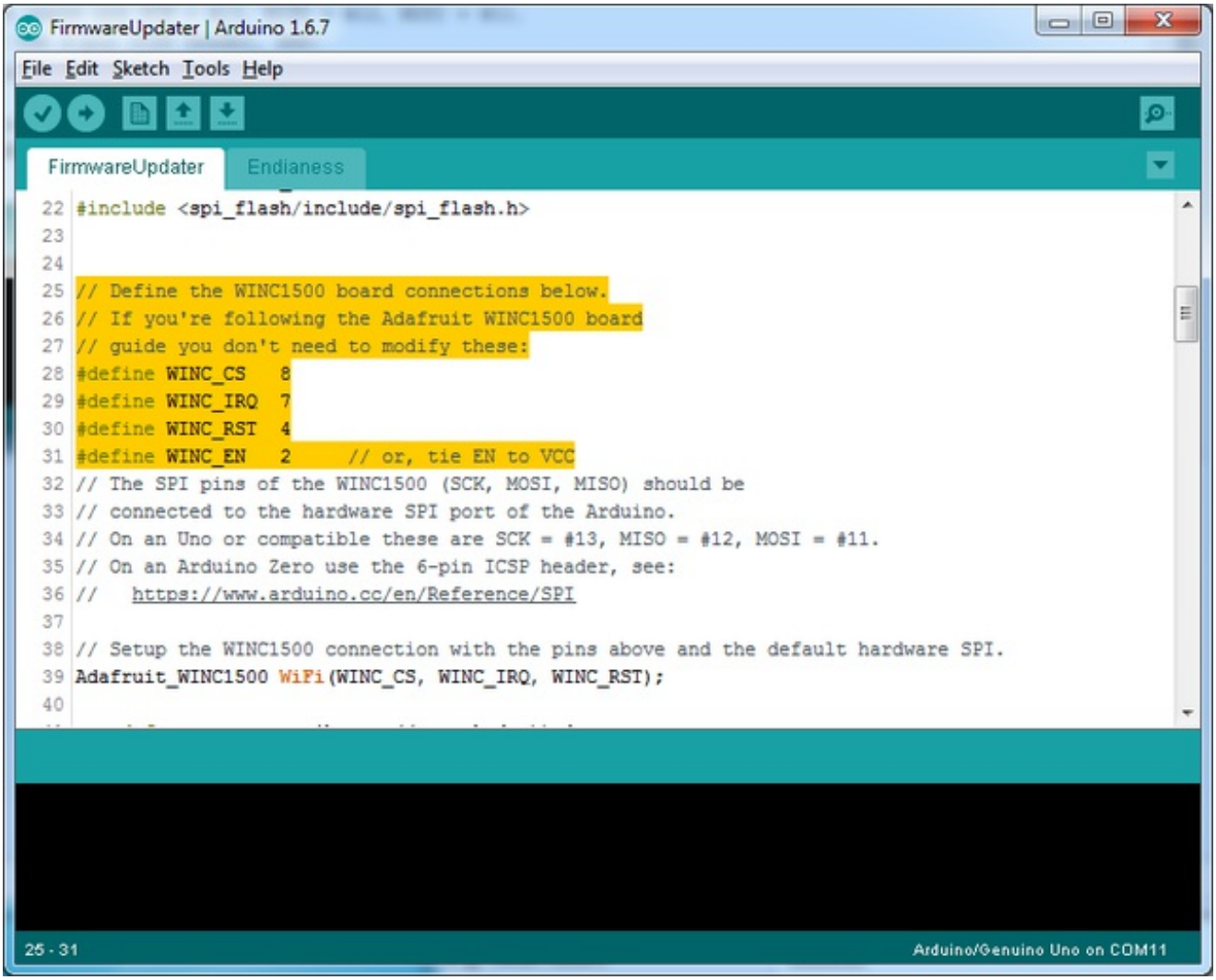
If you're trying to connect to a computer or service via SSL and the connection is failing, you may need to update the certificates built into the WINC1500. By default it comes with many of the most popular SSL certificates but you may bump into a site that requires one that isnt included.

Its quite easy to update the certificates, you'll need to upload some code and run the uploaders but it only has to happen once

Start out by uploading the **FirmwareUpdater** sketch from **Adafruit_WINC1500**

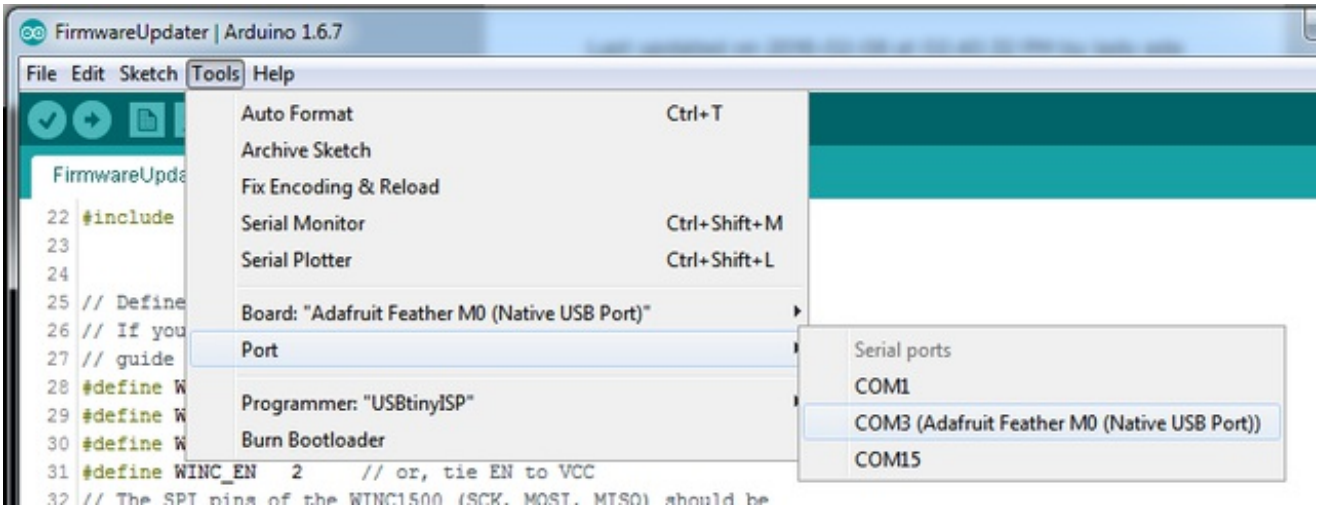


update the pins as necessary, we have the default for use with the Feather M0 WINC1500



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



Now download or clone the WiFi101 Firmware Updater repository (http://adafru.it/leT) from github, you can just click here to grab the latest Zip

Download WiFi101 Firmware Updater

http://adafru.it/leU

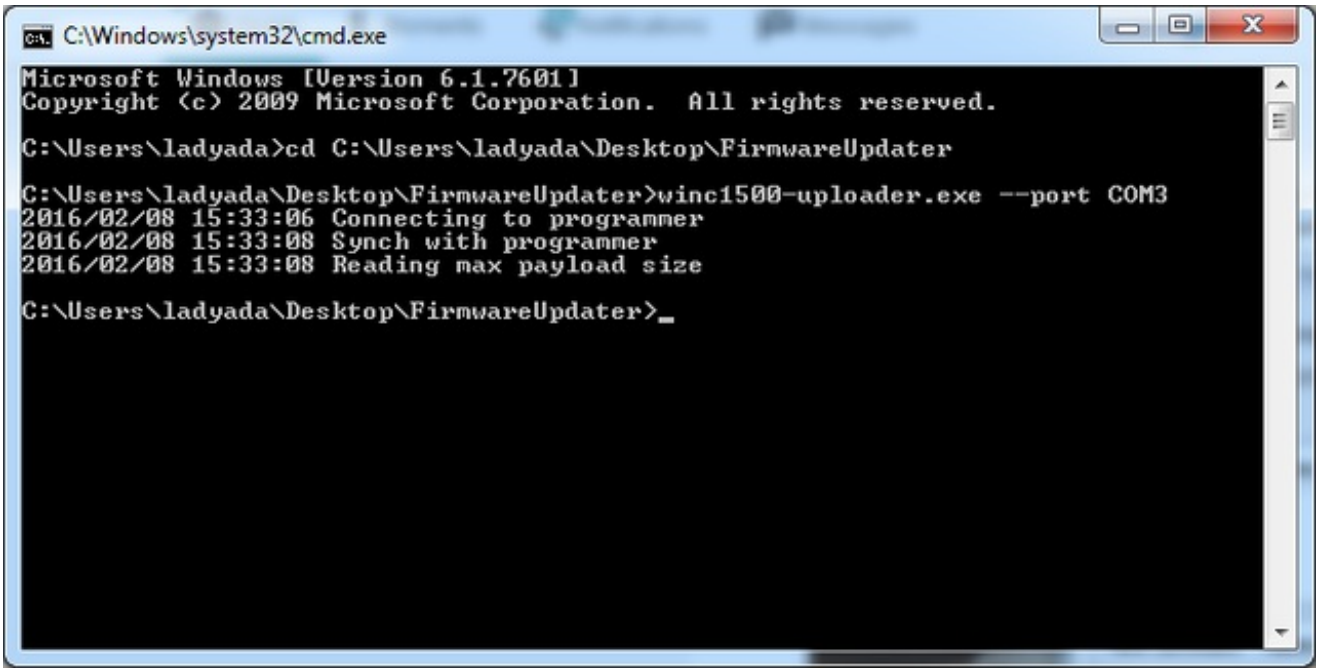
Command Line Usage

Windows

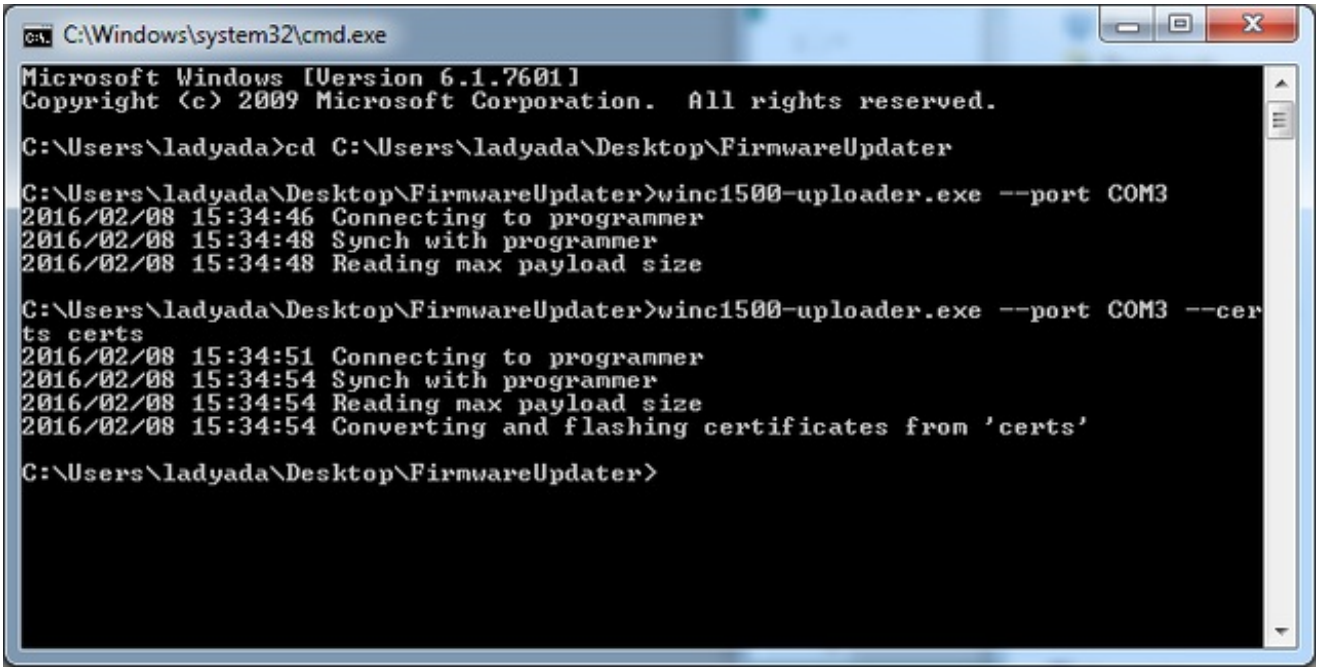
Uncompress it on your desktop. Now use powershell, command or terminal to **cd** to the uncompressed directory and run

winc1500-uploader --port *serialport*

for example, on windows, **winc1500-uploader --port COM3**

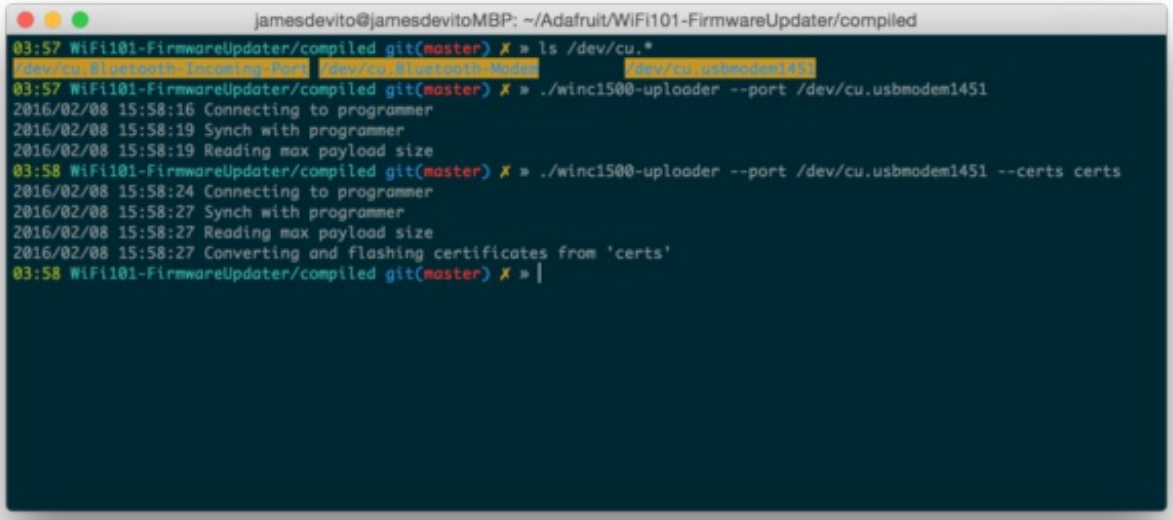


You should see that it was able to reda the max payload size. Next up just run the same command but add **--certs certs** to upload all the certificates in the certs directory



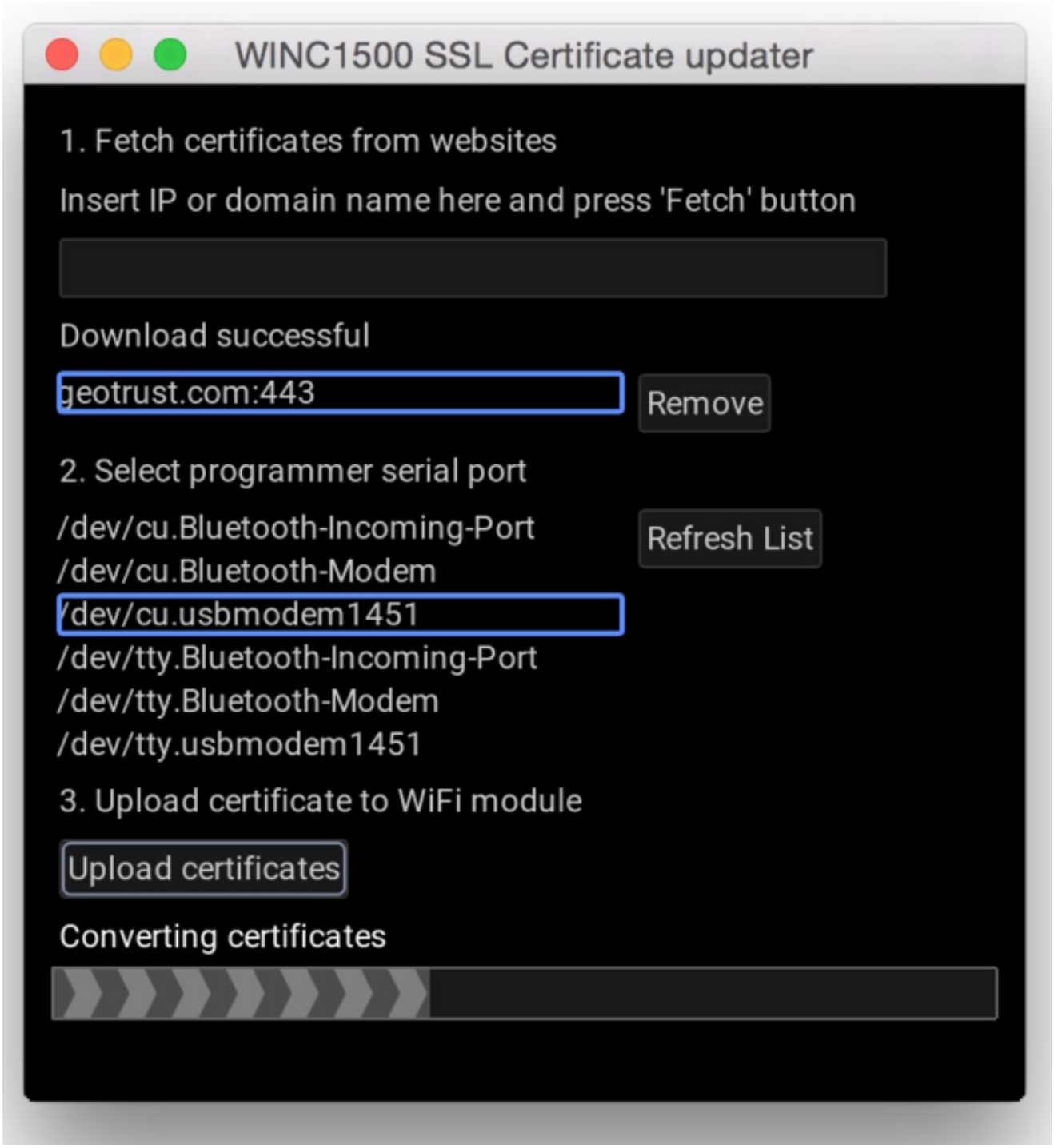
Mac OS X (Command Line) Usage

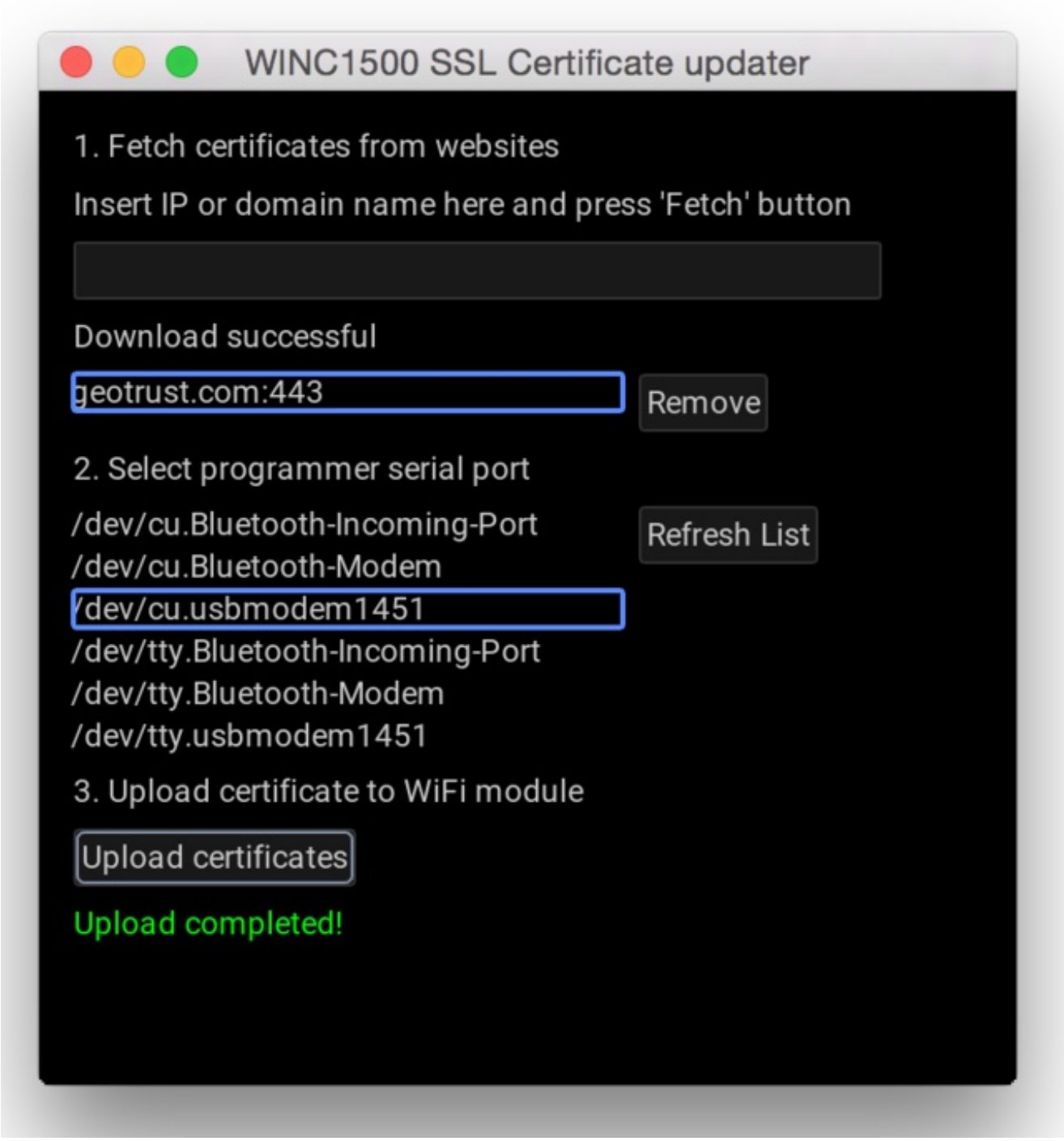
With Mac, you can use command line, its essentially the same as above except the serial port will be something like **/dev/cu.usbserialnnnn** You can figure it out by running **ls /dev/cu.*** to list all serial devices, ignore the Bluetooth modem interfaces



GUI Usage

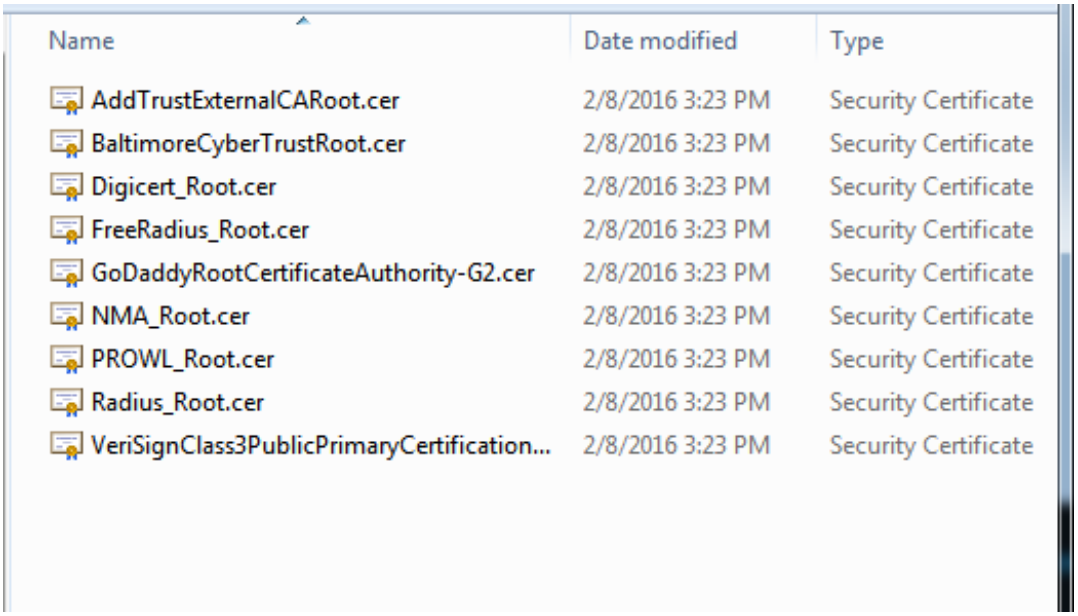
You can also use the GUI which is nice and will also let you fetch the certificate and upload it directly. If you don't need any particular site's cert just put in www.google.com


WINC1500 SSL Certificate updater
1. Fetch certificates from websites
Insert IP or domain name here and press 'Fetch' button
Download successful
geotrust.com:443
Remove
2. Select programmer serial port
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.Bluetooth-Modem
/dev/cu.usbmodem1451
/dev/tty.Bluetooth-Incoming-Port
/dev/tty.Bluetooth-Modem
/dev/tty.usbmodem1451
Refresh List
3. Upload certificate to WiFi module
Upload certificates
Converting certificates

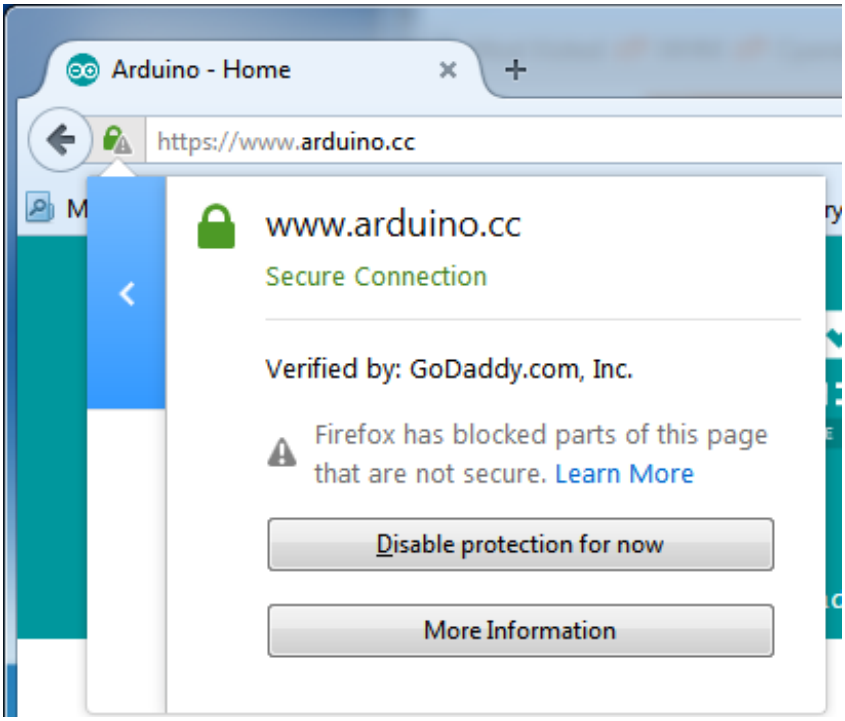


Manually Adding Certificates

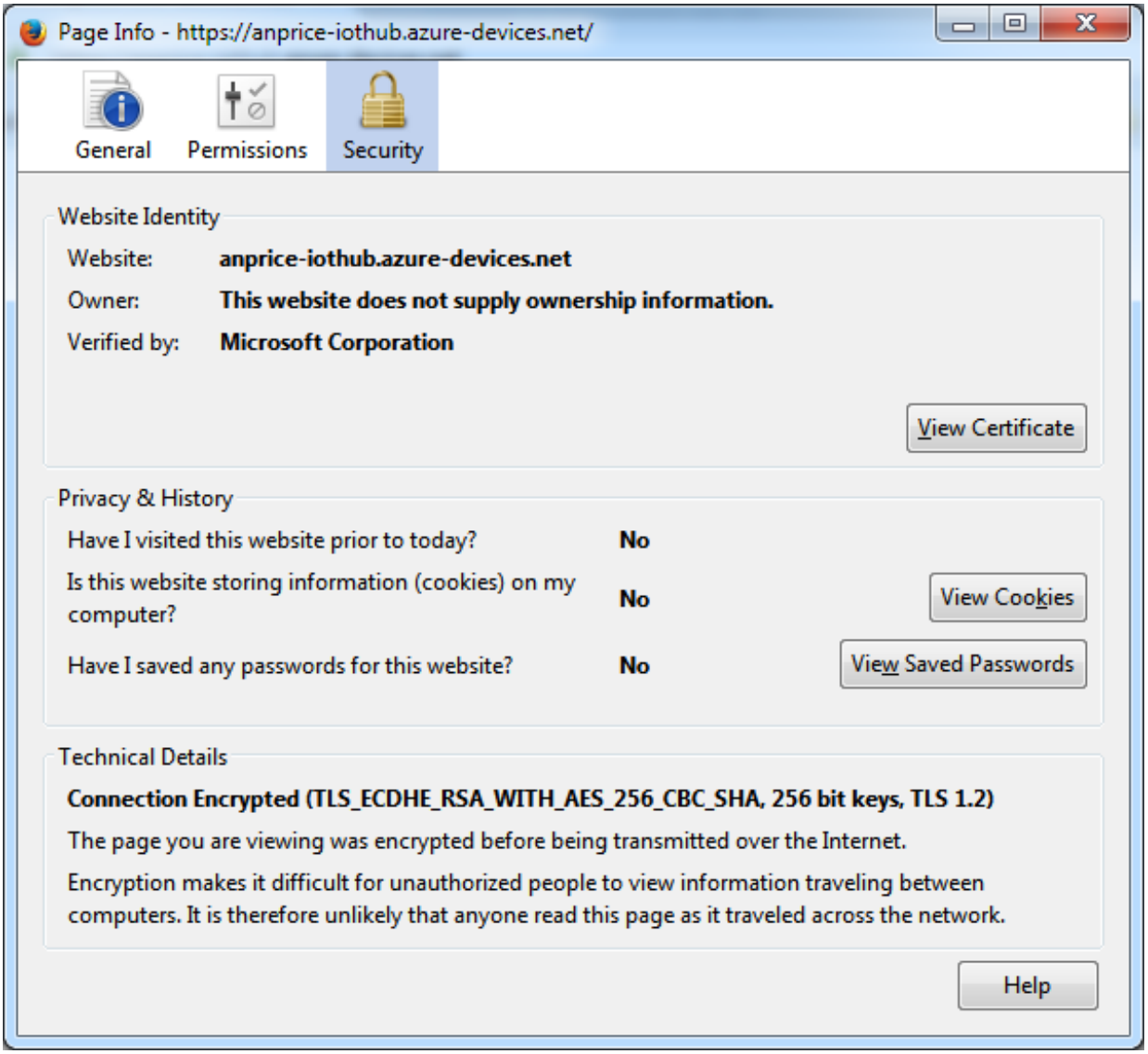
You can upload other certificates, make sure they are in DER format (http://adafru.it/leV) (binary, not ascii!) and end the name with .cer



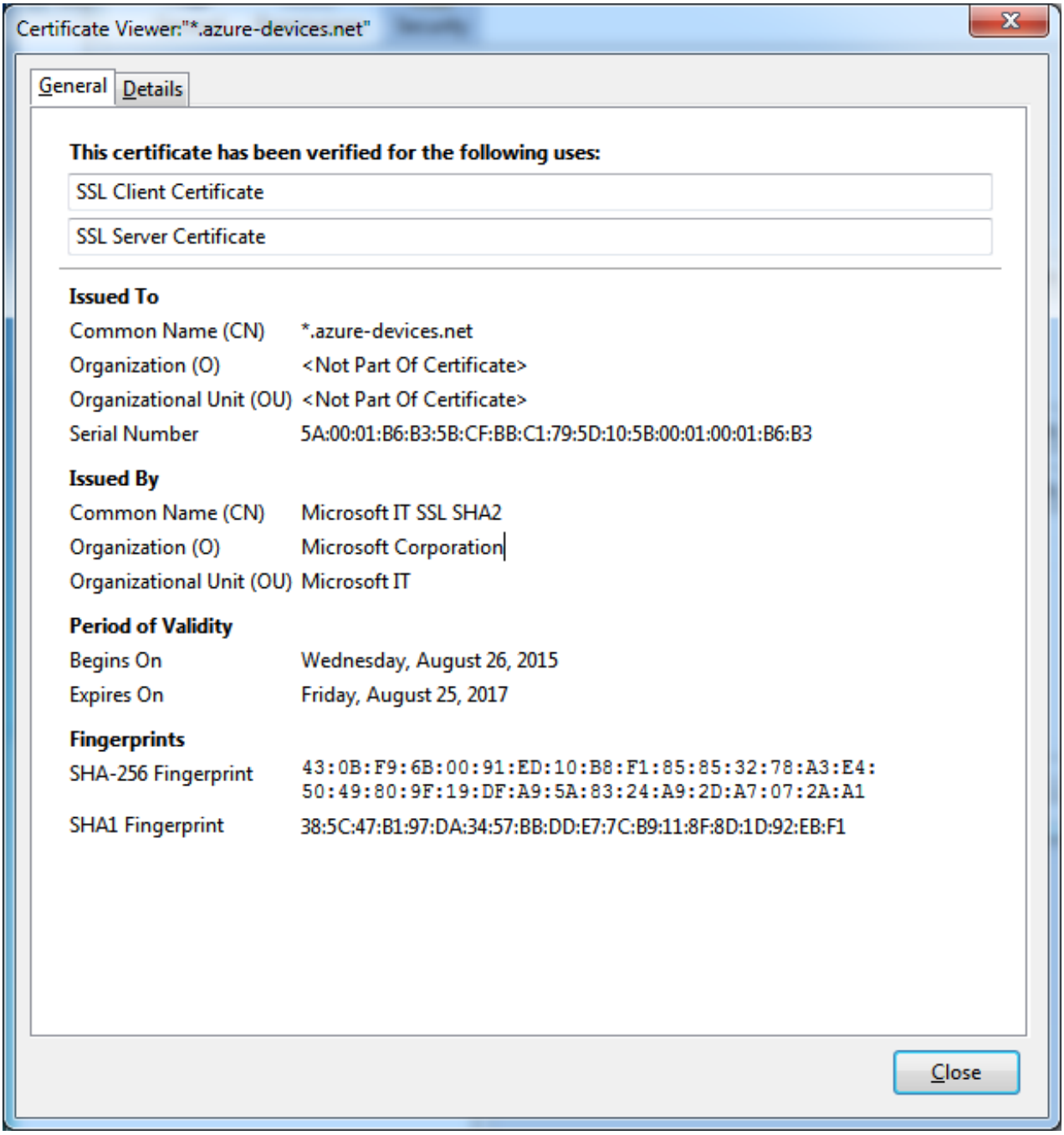
To figure out what certificate you need, go to the page you're trying to connect to, using your browser. Then click on the lock (it may be in a different location) to make sure you're using **https** and its secured. Then click **More Information**



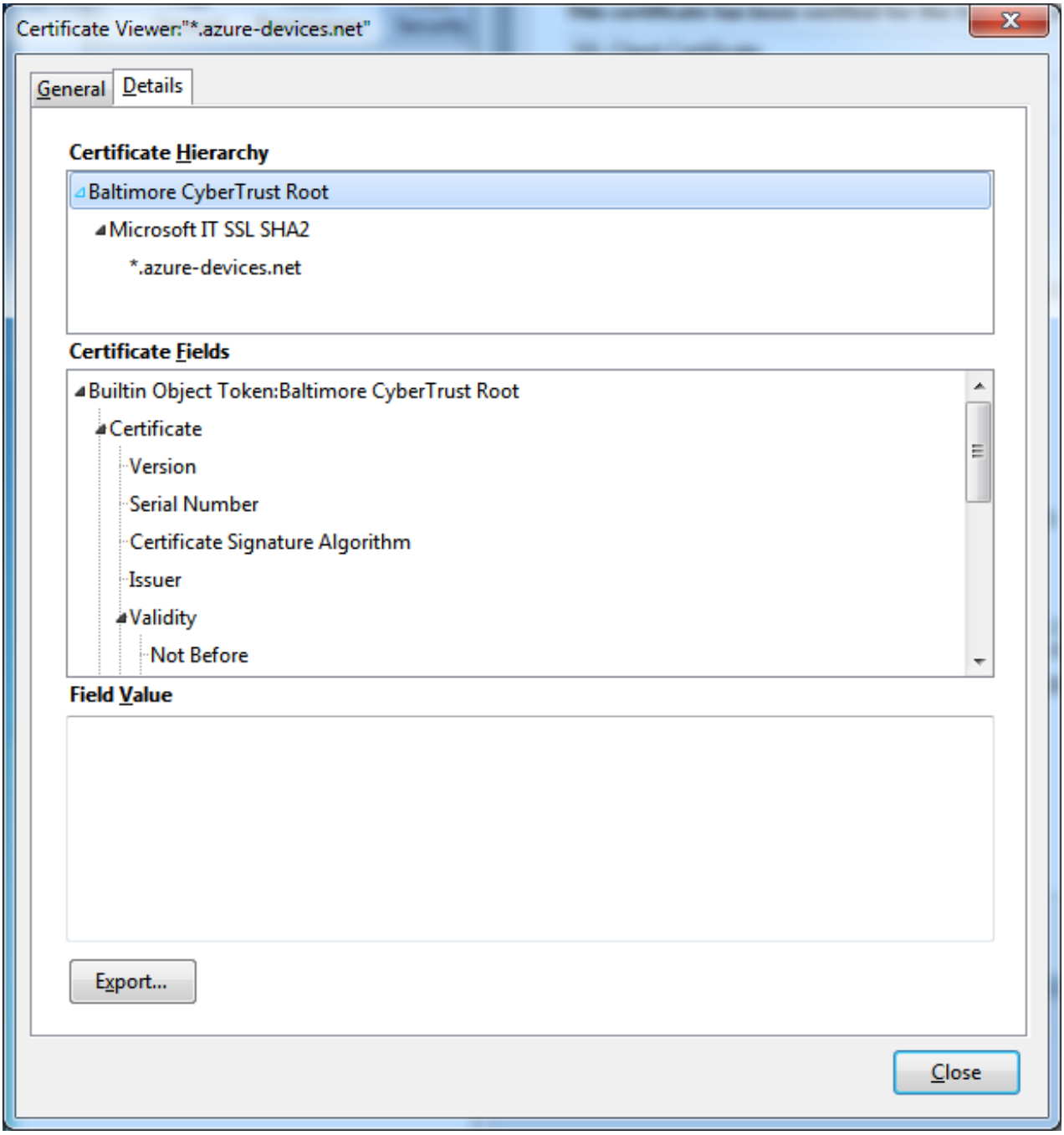
When you get the details popup, click on **View Certificate**



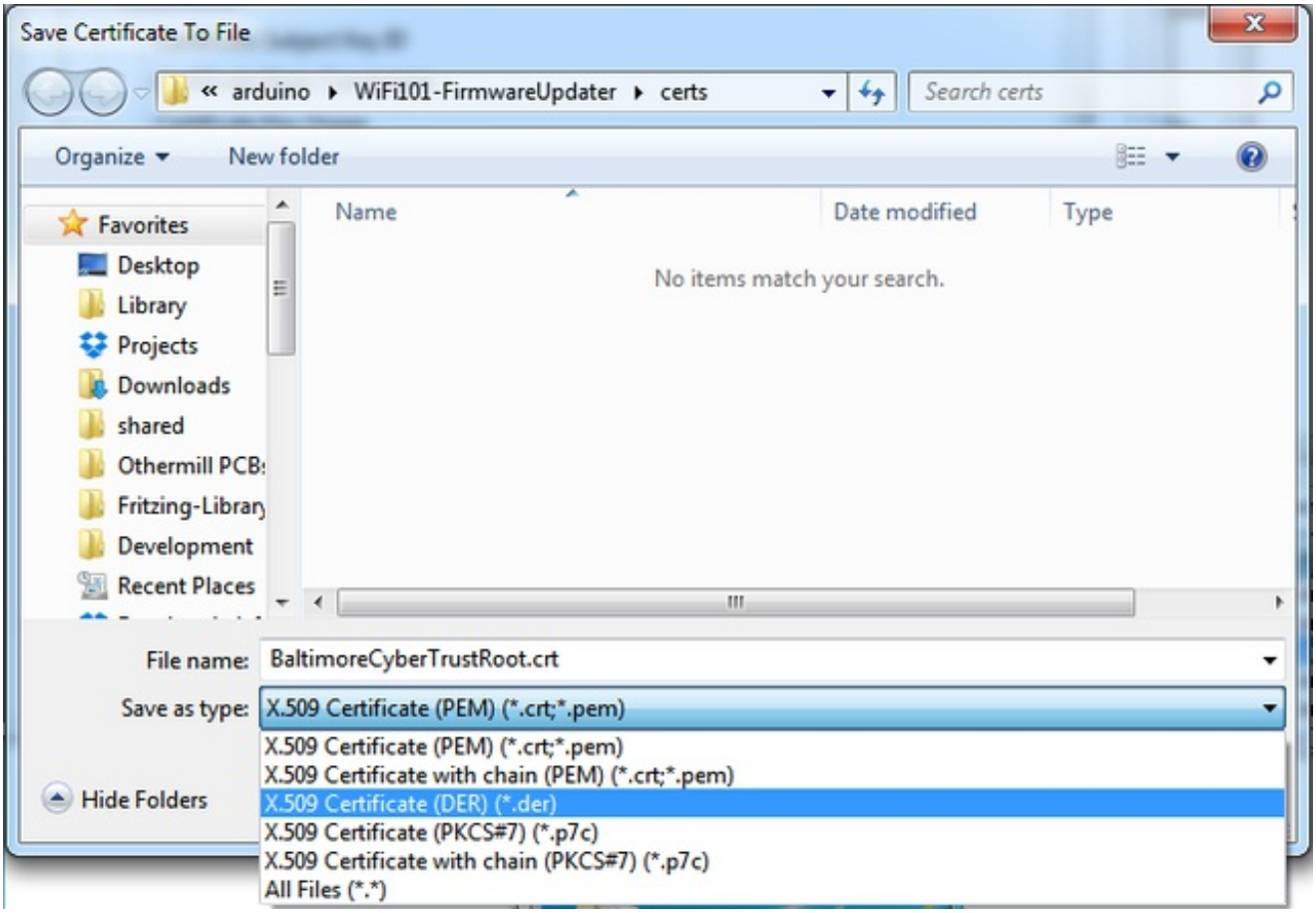
Almost done, once you see the view of the certificate, click on **Details**



Finally, you can see the Root Certificate, its at the top of **Certificate Hierarchy** now click **Export**

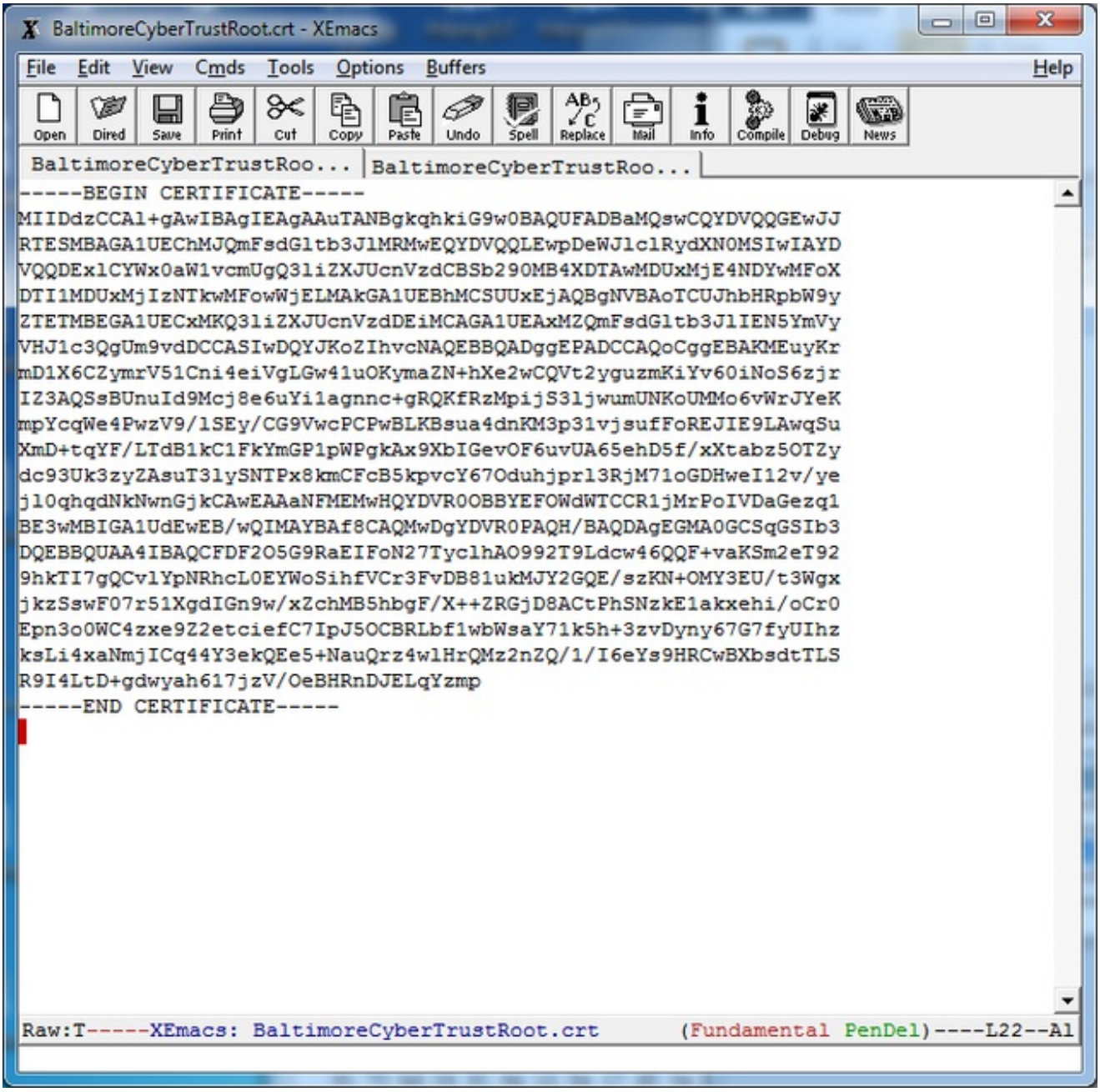


and export/save it to the **certs** directory, in **DER** format

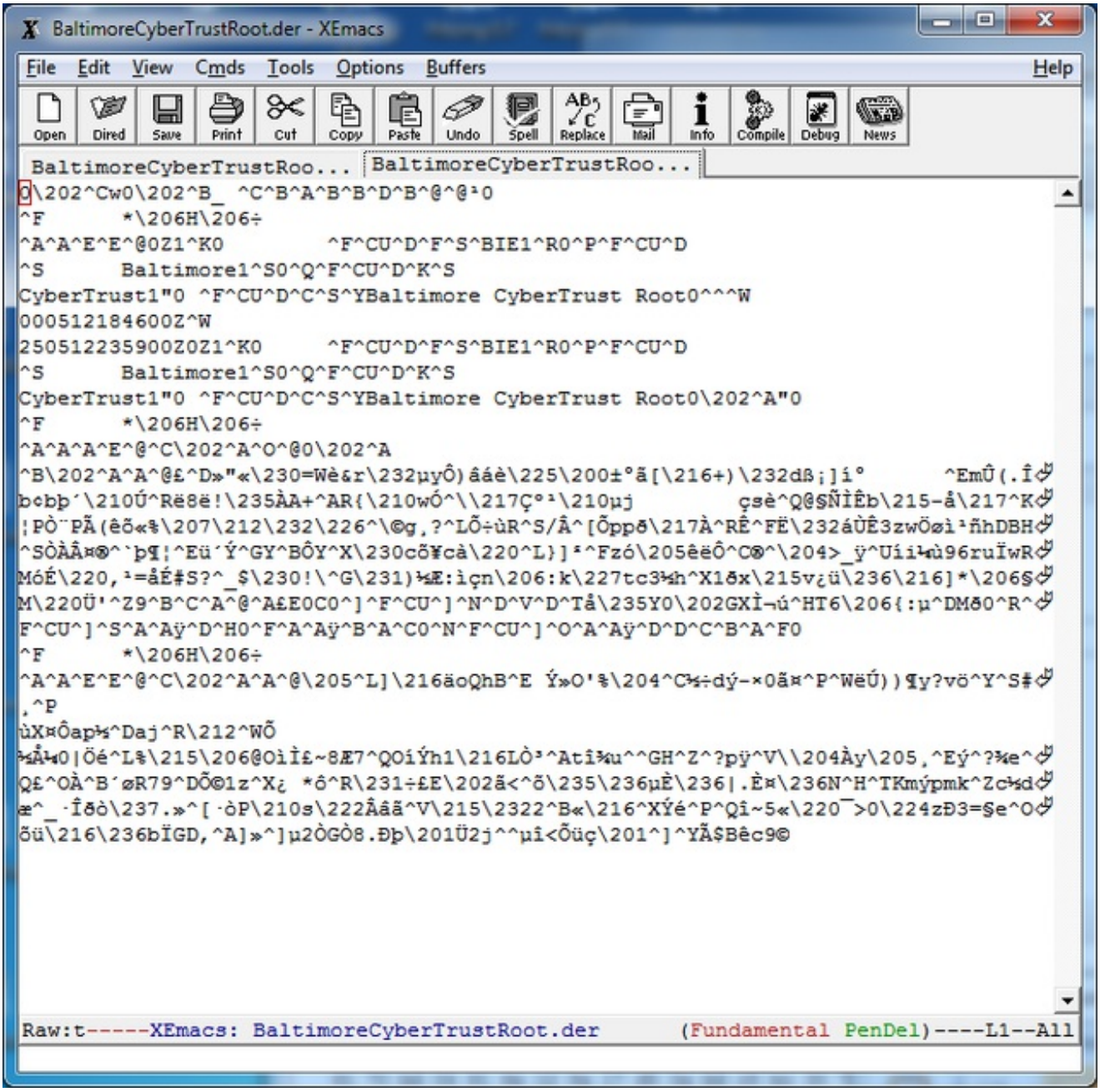


Certicate Format

Open up the certificate in a text editor, if you see this you have an ascii certificate **which is not what you want!**



Instead, make sure its in binary format like this (it should be a jumble of characters)


BaltimoreCyberTrustRoot.der - XEmacs
File Edit View Cmds Tools Options Buffers Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
BaltimoreCyberTrustRoo...
BaltimoreCyberTrustRoo...
0\202^Cw0\202^B_ ^C^B^A^B^B^D^B^@^@³0
^F *\206H\206÷
^A^A^E^E^@0Z1^K0 ^F^CU^D^F^S^BIE1^R0^P^F^CU^D
^S Baltimore1^S0^Q^F^CU^D^K^S
CyberTrust1"0 ^F^CU^D^C^S^YBaltimore CyberTrust Root0^^^W
000512184600Z^W
250512235900Z0Z1^K0 ^F^CU^D^F^S^BIE1^R0^P^F^CU^D
^S Baltimore1^S0^Q^F^CU^D^K^S
CyberTrust1"0 ^F^CU^D^C^S^YBaltimore CyberTrust Root0\202^A"0
^F *\206H\206÷
^A^A^A^E^@^C\202^A^O^@0\202^A
^B\202^A^A^@£^D»"«\230=Wè&r\232µyÔ)âáè\225\200±°ã[\216+)\232dß¡]í° ^EmÛ(.Î
b<bþ´\210Ú^Rë8ë!\235ÀA+^AR{\210wÓ^\\217Ç°³\210µj çsè^Q@§ÑÌÊb\215-å\217^K
¦PÒ¨PÃ(êõ«%\207\212\232\226^\©g¸?^LÕ÷ùR^S/Â^[Õppð\217À^RÊ^FË\232áÙÊ3zwÖøì¹ñhDBH
^SÒÀÂ¤®^`þ¶¦^Eü´Ý^GY^BÔY^X\230cõ¥cà\220^L}]²^Fzó\205êëÔ^C®^\204>_ÿ^Uíi↳ù96ruÏwR
MóÉ\220,³=åÉ#S?^_$\230!\^G\231)½Æ:içn\206:k\227tc3¾h^X1ðx\215v¿ü\236\216]*\206§
M\220Ü'^Z9^B^C^A^@^A£E0C0^]^F^CU^]^N^D^V^D^Tå\235Y0\202GXÌ¬ú^HT6\206{:µ^DMð0^R^
F^CU^]^S^A^Aÿ^D^H0^F^A^Aÿ^B^A^C0^N^F^CU^]^O^A^Aÿ^D^D^C^B^A^F0
^F *\206H\206÷
^A^A^E^E^@^C\202^A^A^@\205^L]\216äoQhB^E Ý»O'%\204^C¾÷dý-×0ã¤^P^WëÚ))¶y?vö^Y^S#
.^P
ùX¤Ôap½^Daj^R\212^WÕ
¾Å↳0|Öé^L%\215\206@OìÌ£~8Æ7^QOíÝh1\216LÒ³^Atî¾u^^GH^Z^?pÿ^V\\204Ày\205¸^Eý^?¾e^
Q£^OÀ^B´øR79^DÕ©1z^X¿ *ô^R\231÷£E\202ã<^õ\235\236µÈ\236|.È¤\236N^H^TKmýpmk^Zc½d
æ^_·Îðò\237.»^[·òP\210s\222Ââã^V\215\2322^B«\216^XÝé^P^Qî~5«\220¯>0\224zÐ3=§e^O
õü\216\236bÏGD,^A]»^]µ2ÒGÒ8.Ðþ\201Ü2j^^µî<Õüç\201^]^YÃ$Bêc9©
Raw:t-----XEmacs: BaltimoreCyberTrustRoot.der (Fundamental PenDel)----L1--All

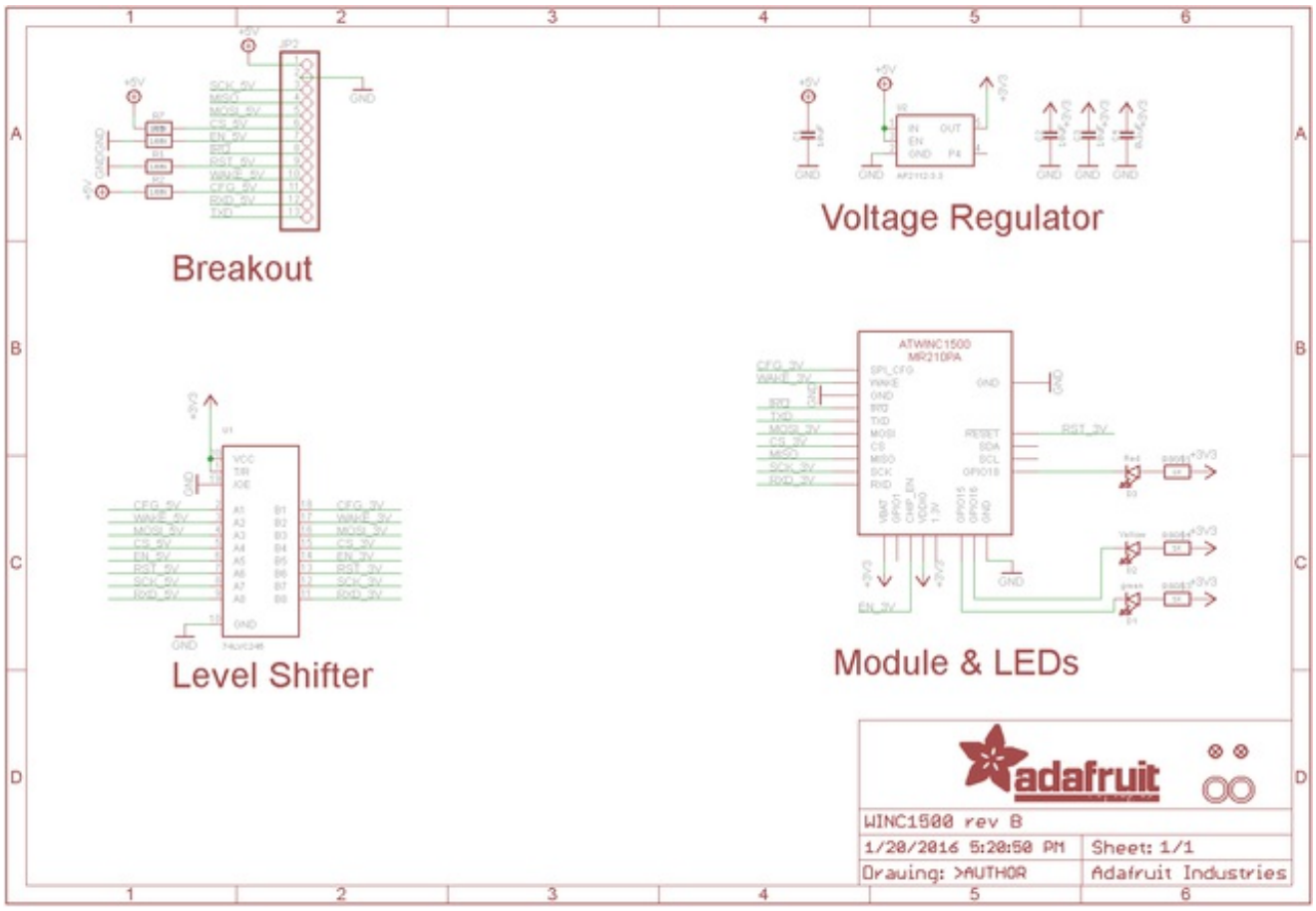
Downloads

Datasheets and More

- All the details you could ever want are over at the Atmel website (http://adafru.it/kUD)

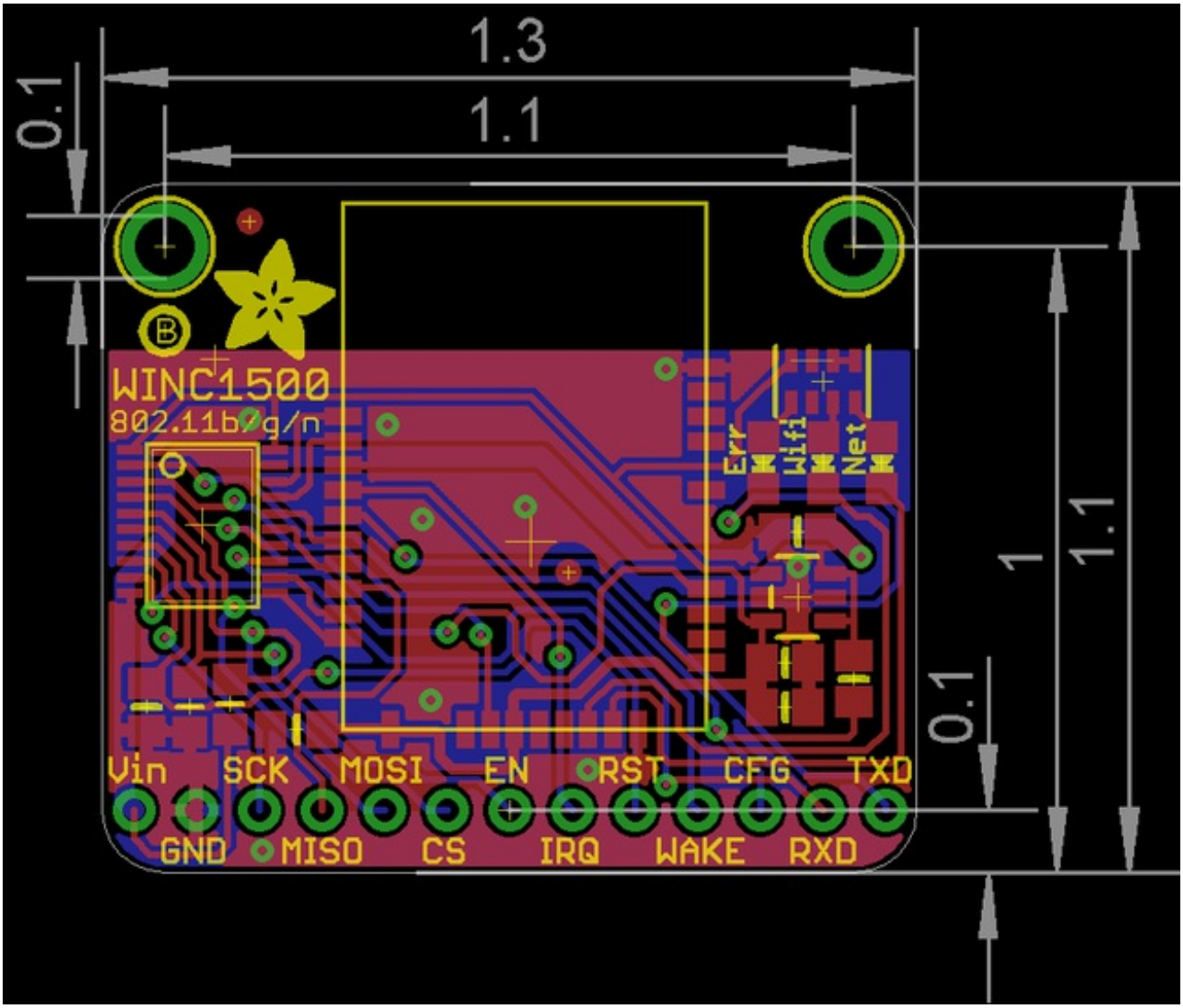
Schematic

Click to embiggen. Please note we ship with the 'PB rev module, not 'PA!



Fabrication Print

Dimensions in inches


1.3
1.1
0.1
WINC1500
802.11b/g/n
Err
Wifi
Net
1
1.1
0.1
Vin
SCK
MOSI
EN
RST
CFG
TXD
GND
MISO
CS
IRQ
WAKE
RXD