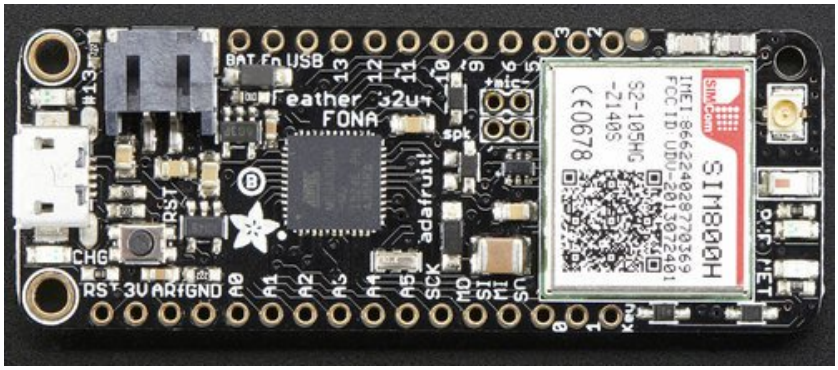
adafruit learning system

Adafruit Feather 32u4 FONA

Created by lady ada



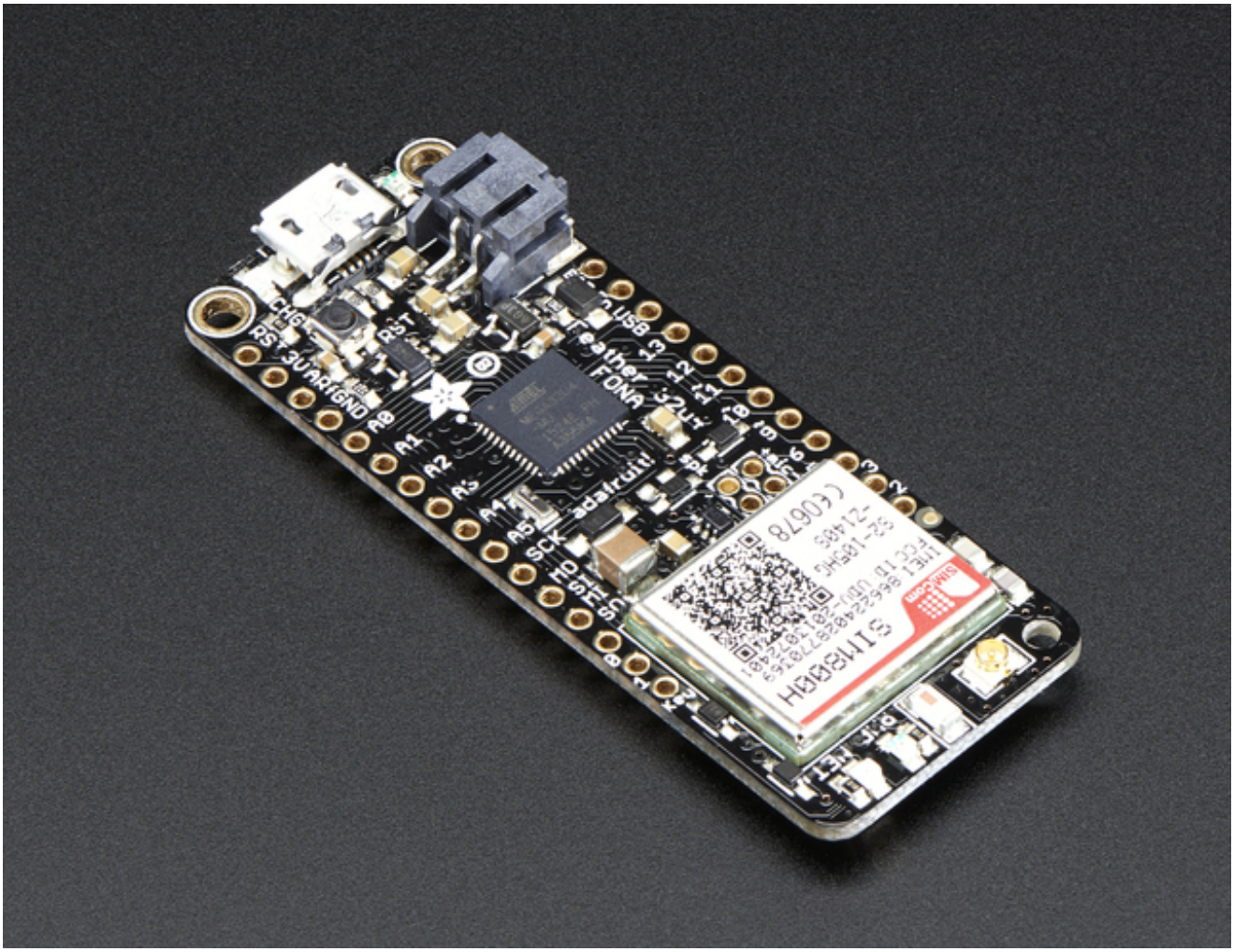
Last updated on 2016-04-27 02:03:47 PM EDT

Guide Contents

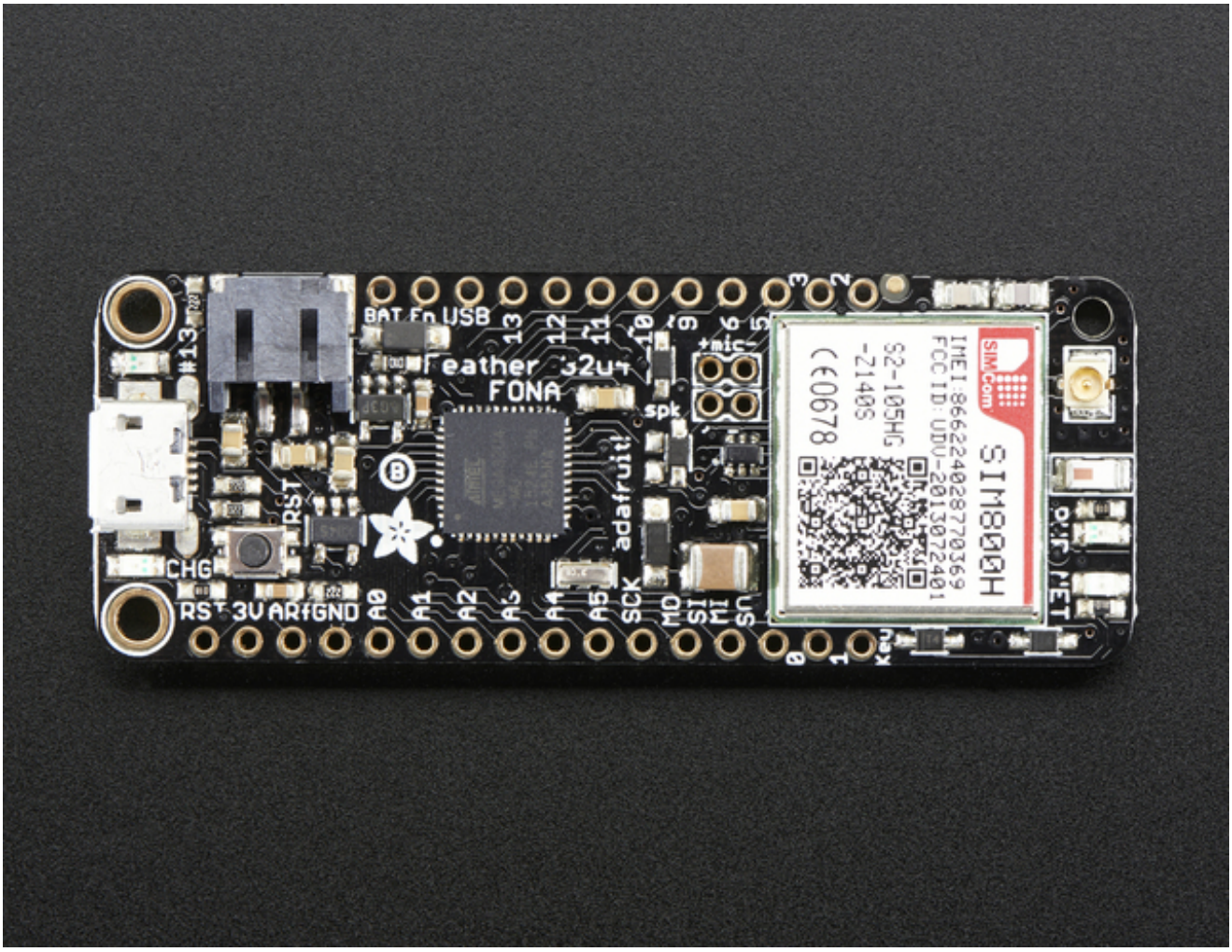
Overview

Feather is the new development board from Adafruit, and like its namesake it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller cores.

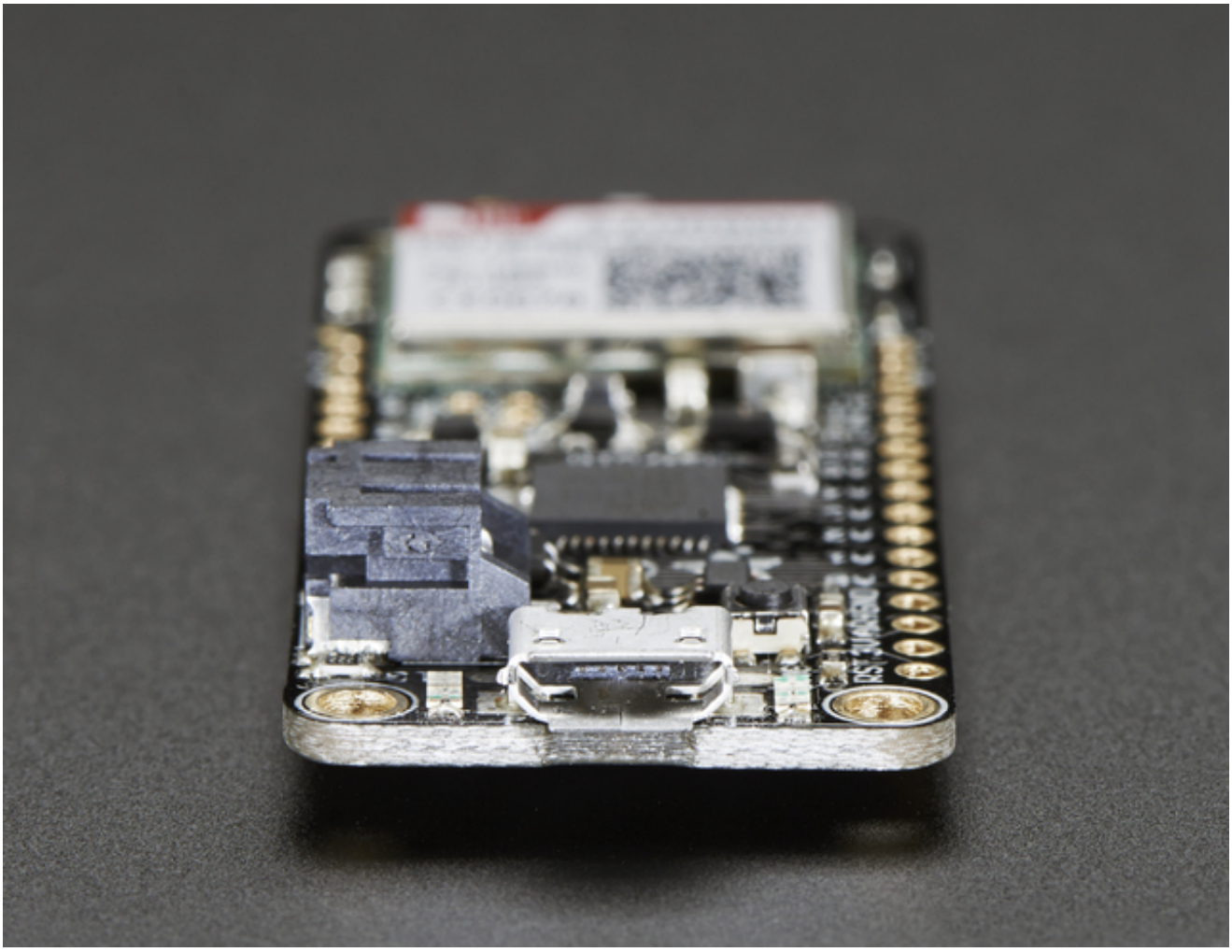
This is the **Adafruit Feather 32u4 FONA** - our take on an 'all-in-one' Arduino-compatible + audio/sms/data capable cellular with built in USB and battery charging. Its an Adafruit Feather 32u4 with a FONA800 module (http://adafru.it/1946), ready to rock! We have other boards and accessories in the Feather family, check'em out here (http://adafru.it/l7B).



At the Feather 32u4's heart is at ATmega32u4 clocked at 8 MHz and at 3.3V logic, a chip setup we've had tons of experience with as it's the same as the Flora (http://adafru.it/dVl). This chip has 32K of flash and 2K of RAM, with built in USB so not only does it have a USB-to-Serial program & debug capability built in with no need for an FTDI-like chip, it can also act like a mouse, keyboard, USB MIDI device, etc.

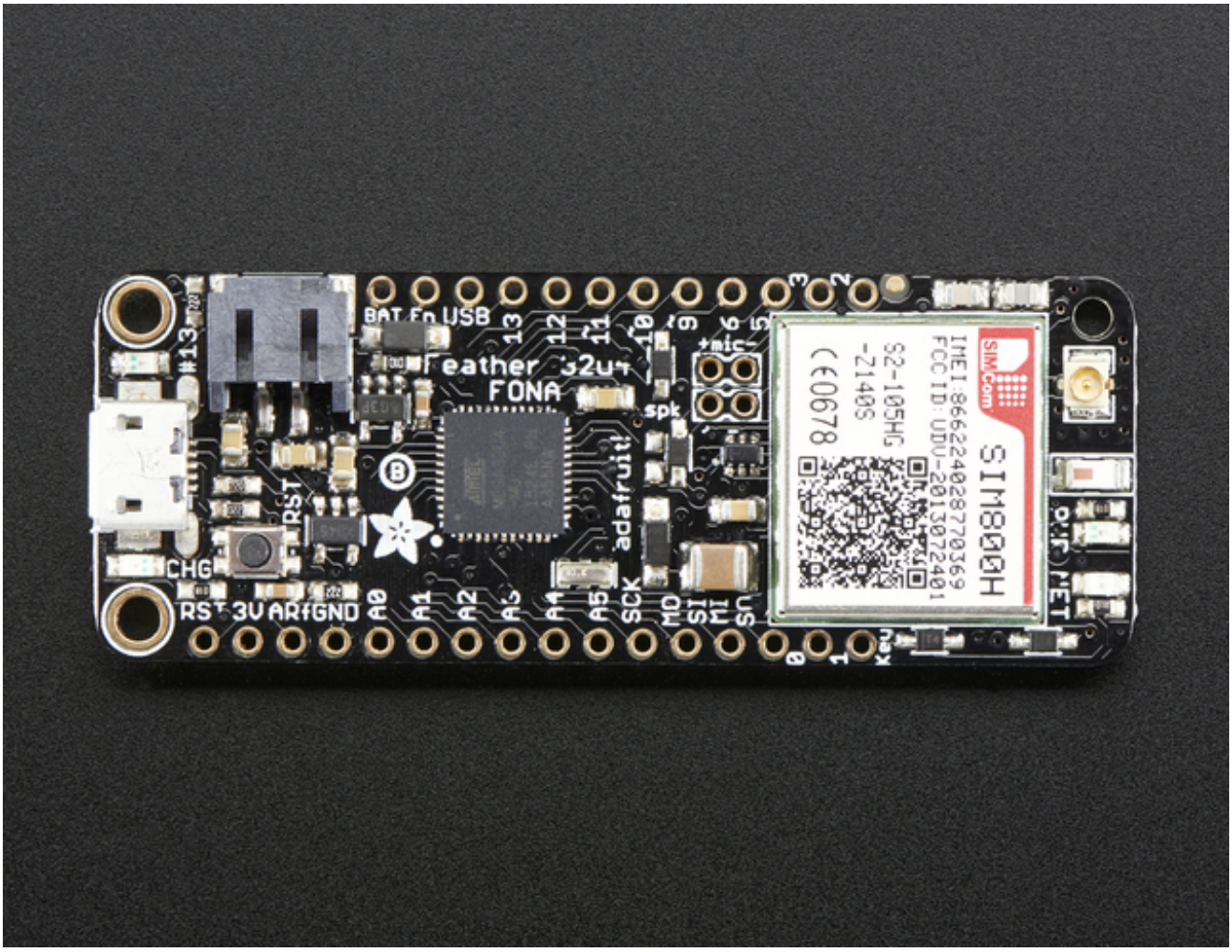


Since you'll be taking this on the road, we added a connector for any of our 3.7V Lithium polymer batteries and built in battery charging. **A 500mAh+ Lipoly battery is required for use**, it keeps the cellular module happy during the high current spikes. Plug the Feather into microUSB to charge at 500mA.



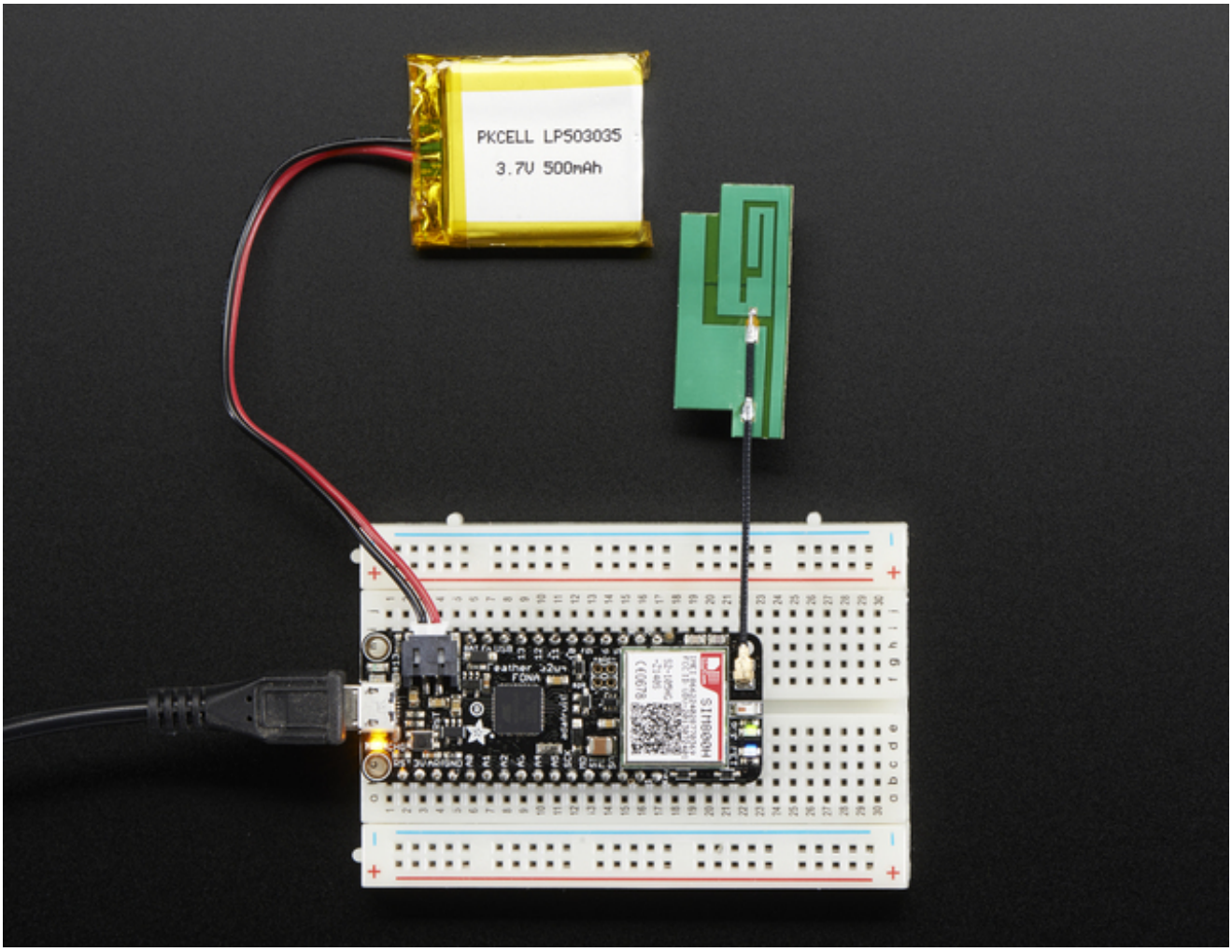
Here's some handy specs! Like all Feather 32u4's you get:

- Measures 2.4" x 0.9" x 0.28" (51mm x 23mm x 8mm) without headers soldered in
- Light as a (large?) feather - 8.2 grams
- ATmega32u4 @ 8MHz with 3.3V logic/power
- 3.3V regulator with 500mA peak current output
- USB native support, comes with USB bootloader and serial port debugging
- You also get tons of pins - 20 GPIO pins
- Hardware Serial, hardware I2C, hardware SPI support
- 8 x PWM pins
- 10 x analog inputs
- Built in 500mA lipoly charger with charging status indicator LED
- Pin #13 red LED for general purpose blinking
- Power/enable pin for the 3.3V regulator
- 4 mounting holes
- Reset button



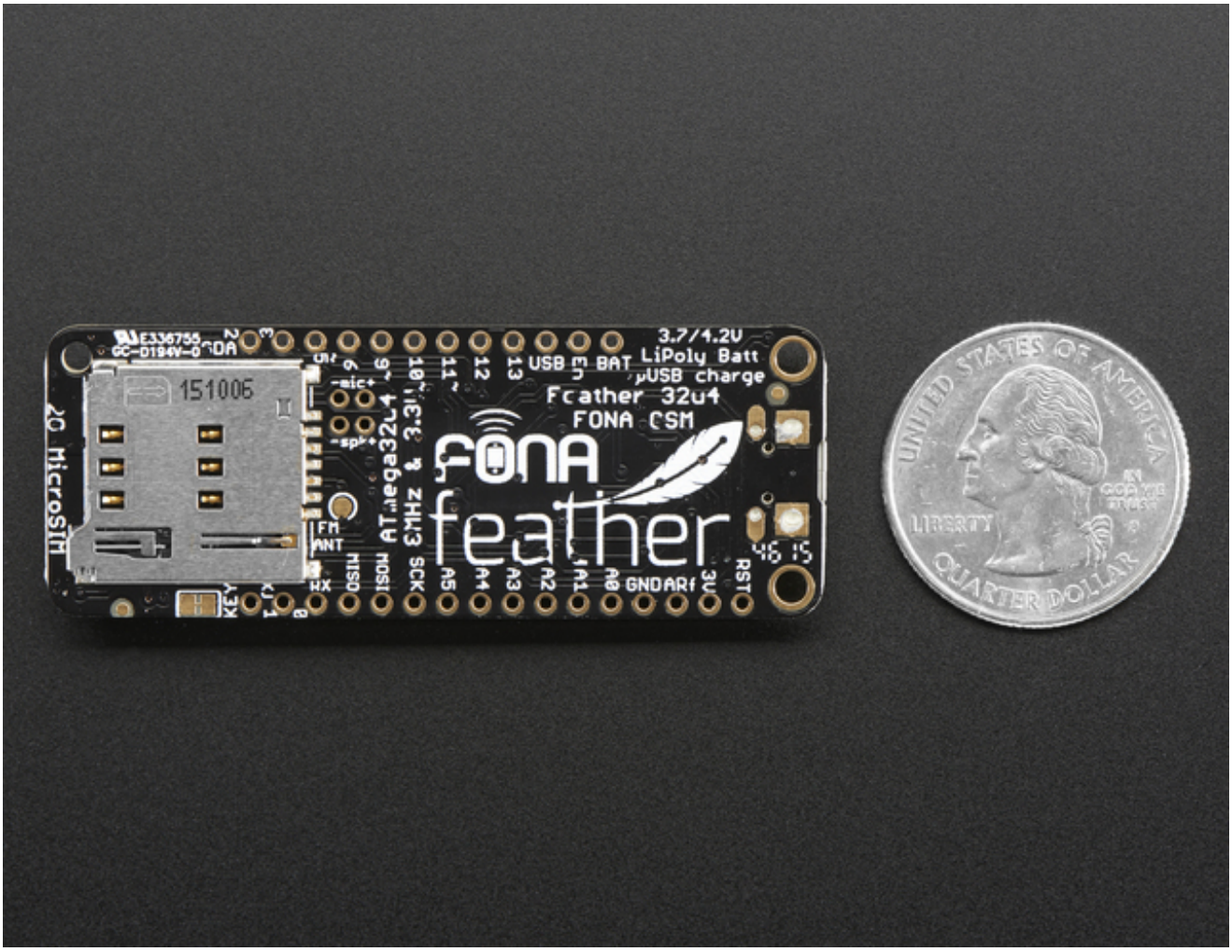
Connect your Feather to the Internet or make phone calls with our trusted-and-tested FONA module. At the heart is a GSM cellular module (we use the latest SIM800) the size of a postage stamp. This module can do just about everything.

- Quad-band 850/900/1800/1900MHz - connect onto any global GSM network with any 2G SIM (in the USA, T-Mobile is suggested)
- Make and receive voice calls using an external 8Ω speaker + electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts (yeah, we don't exactly know why this was included but it works really well)
- AT command interface with "auto baud" detection
- Pair-able Bluetooth client interface with SPP (for controlling the module) as well as audio.



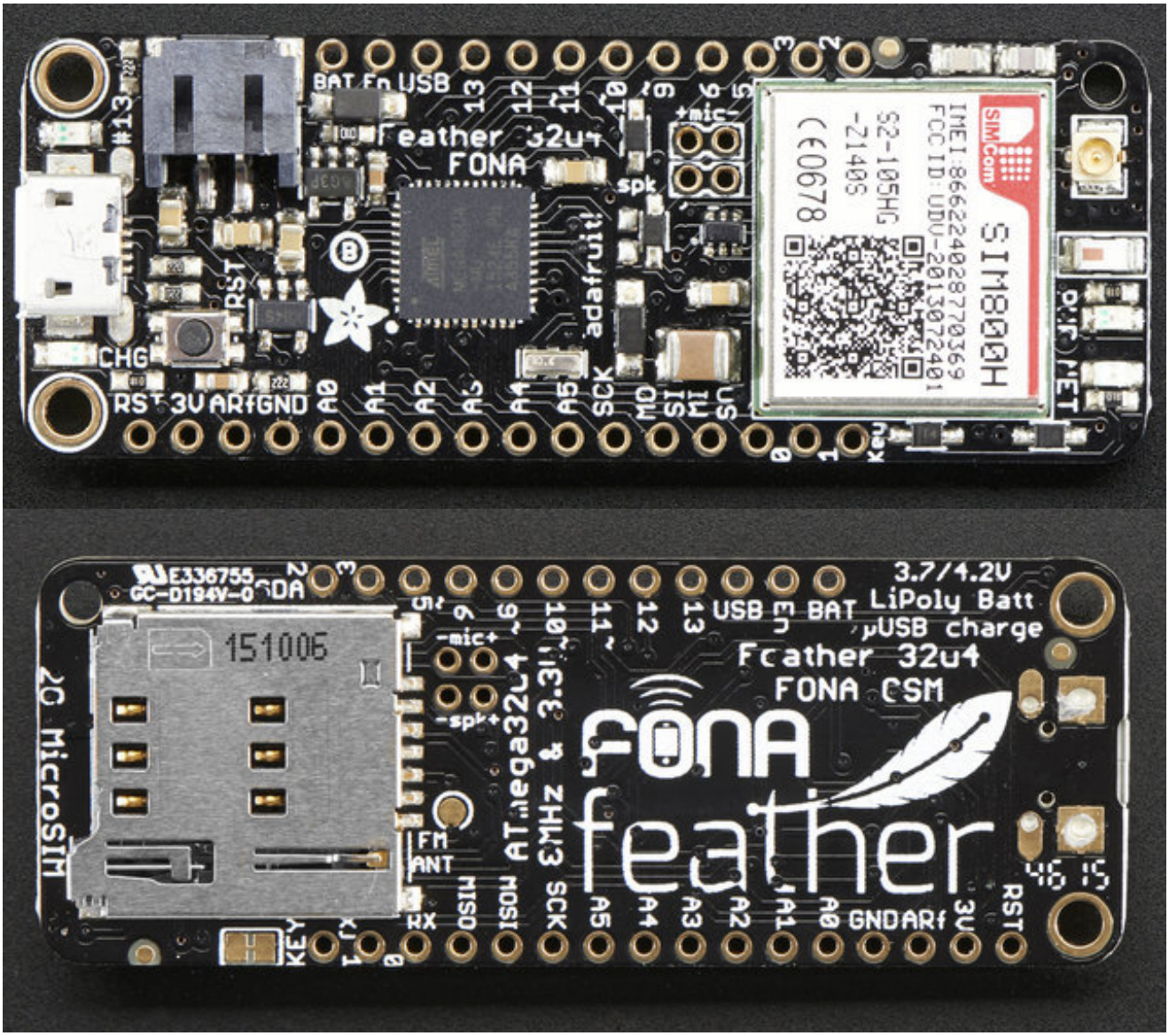
You will also need some required accessories to make Feather FONA work. These are not included!

- **SIM Card!** A 2G Mini SIM card is required to do anything on the cellular network. US AT&T no longer sells 2G SIMs and will shut off their 2G network, so for American customers we recommend any T-Mobile or reseller (TING, SIMPLE mobile, etc) that uses the T-Mobile network. (http://adafru.it/2505)
- **Lipoly Battery** - 500mAh or larger! This 500mAh (http://adafru.it/drL)battery, or this 1200mAh (http://adafru.it/258) will work great.
- **MicroUSB cable (http://adafru.it/592)** for charging the battery.
- **External Antenna** - We like this slim sticker-type (http://adafru.it/1991), which plugs right in. Alternatively, this straight SMA one (http://adafru.it/1859) or this right-angle SMA one will work (http://adafru.it/1858) but you'll also need a uFL to SMA adapter cable (http://adafru.it/851) so you can connect to your SMA antenna
- **External Mic & Speaker** - If you want to make phone calls, you'll also need this electret mic (http://adafru.it/dDa) and mini 8 ohm speaker (http://adafru.it/dDb)

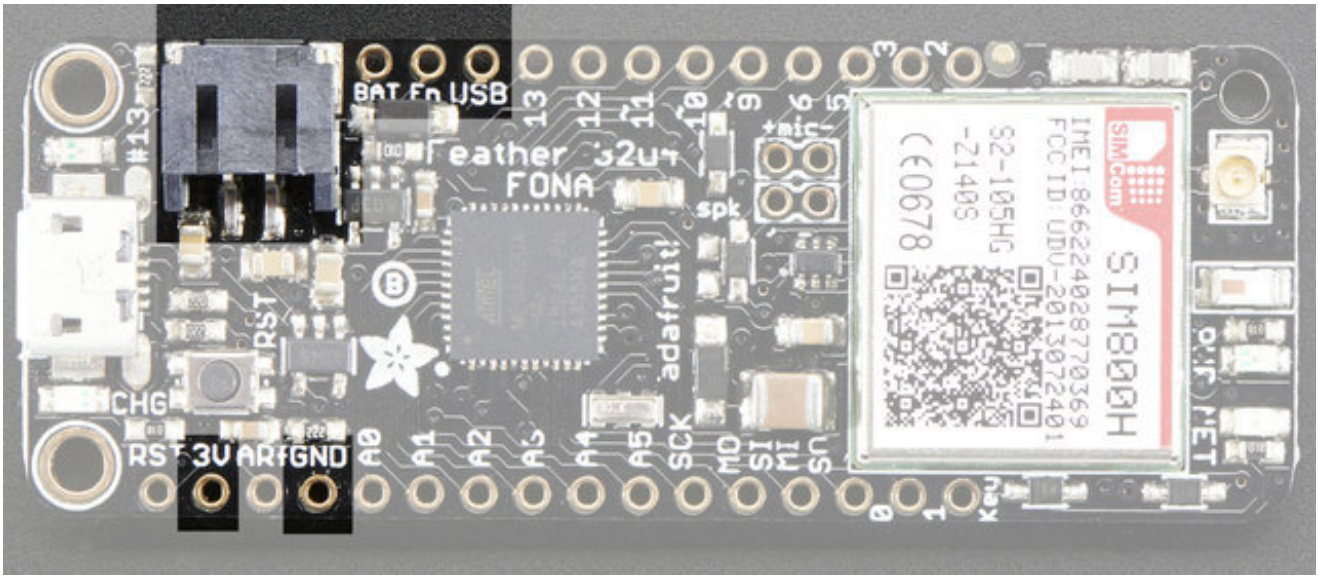

3.7/4.2V
LiPoly Batt
µUSB charge
Feather 32u4
FONA CSM
FONA
feather
151006
UNITED STATES OF AMERICA
LIBERTY
QUARTER DOLLAR

Pinouts

The Feather 32u4 FONA is chock-full of microcontroller goodness. There's also a lot of pins and ports. We'll take you a tour of them now!



Power Pins



- **GND** - this is the common ground for all power and logic
- **BAT** - this is the positive voltage to/from the JST jack for the optional Lipoly battery
- **USB** - this is the positive voltage to/from the micro USB jack if connected
- **EN** - this is the 3.3V regulator's enable pin. It's pulled up, so connect to ground to disable the 3.3V regulator. Note that the cellular module is powed by VBAT so this will only disable the microcontroller
- **3V** - this is the output from the 3.3V regulator, it can supply 500mA peak

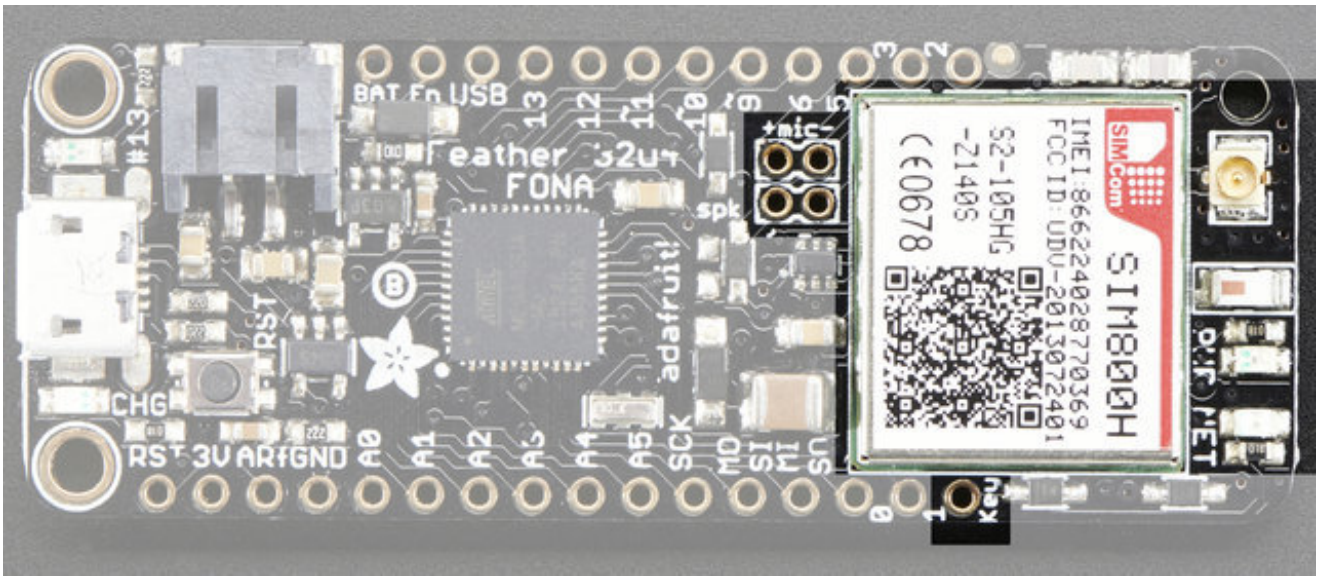
Logic pins

This is the general purpose I/O pin set for the microcontroller. All logic is 3.3V

- **#0** / **RX** - GPIO #0, also receive (input) pin for **Serial1** and Interrupt #2
- **#1** / **TX** - GPIO #1, also transmit (output) pin for **Serial1** and Interrupt #3
- **#2** / **SDA** - GPIO #2, also the I2C (Wire) data pin. There's no pull up on this pin by default so when using with I2C, you may need a 2.2K-10K pullup. Also Interrupt #1
- **#3** / **SCL** - GPIO #3, also the I2C (Wire) clock pin. There's no pull up on this pin by default so when using with I2C, you may need a 2.2K-10K pullup. Can also do PWM output and act as Interrupt #0.
- **#5** - GPIO #5, can also do PWM output. Also connected to the FONA's DTR pin if you want to use it for powersaving functionality, which is not enabled by default
- **#6** - GPIO #6, can also do PWM output and analog input **A7**. Also connected to FONA RTS in case you want to use flow control, which is not enabled by default
- **#9** - GPIO #9, **connected to FONA RXD.**
- **#10** - GPIO #10, also analog input **A10** and can do PWM output.
- **#11** - GPIO #11, can do PWM output.
- **#12** - GPIO #12, also analog input **A11** and can do PWM output.
- **#13** - GPIO #13, can do PWM output and is connected to the **red LED** next to the USB jack
- **A0 thru A5** - These are each analog input as well as digital I/O pins.

- **SCK/MOSI/MISO** - These are the hardware SPI pins. Also used to reprogram the chip with an AVR programmer if you need.

Cellular Module



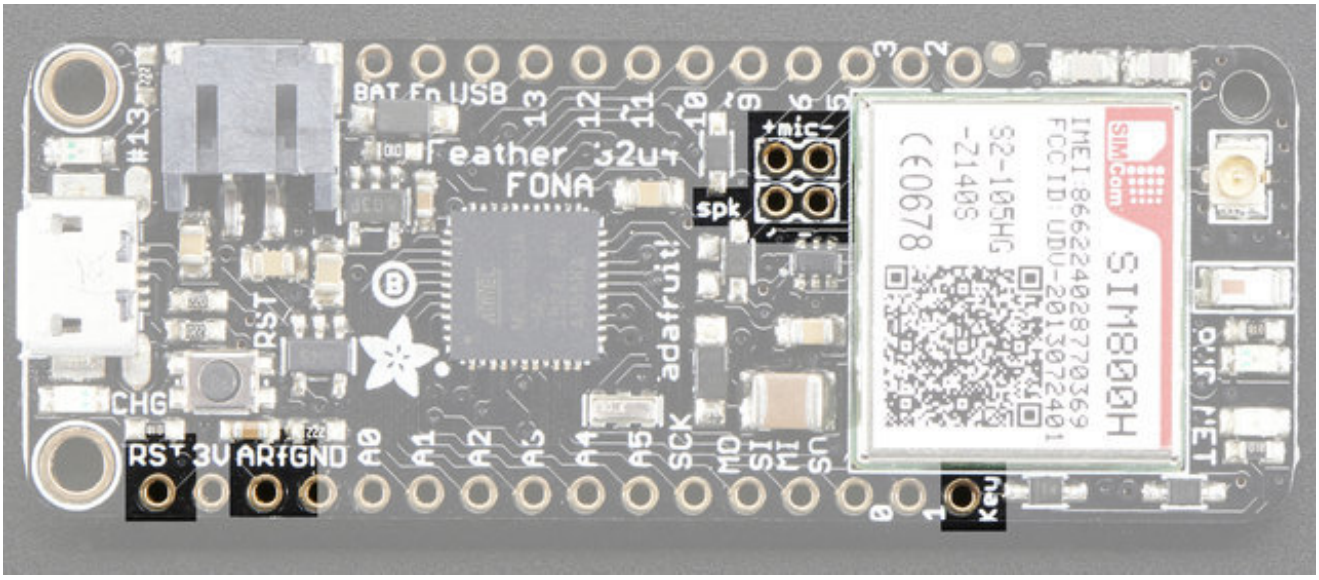
Now to the fun part, the cellular module. THere's a few pins that are used to control the module. It uses SoftwareSerial to communicate with the microcoller

- **#8** - used as the FONA **TXD** (data out from module to AVR)
- **#9** - used as the FONA **RXD** (data out from AVR to module)
- **#7** - used as the FONA **RI** (ring interrupt) pin, you can use this to alert you when an SMS or phone call comes in
- **#4** - used as the FONA **Reset** pin. You can pulse this pin low to reset the FONA, handy when starting up

Optional pins:

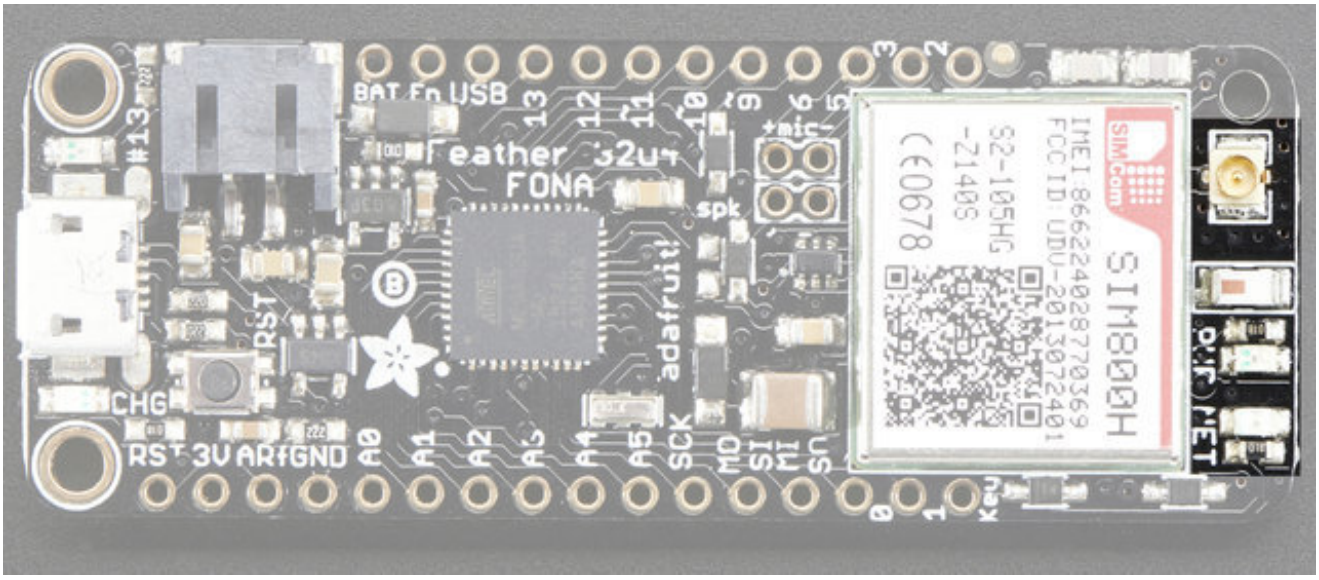
- **#5** - This pin is available on the breakout and is also connected to the FONA's DTR pin if you want to use it for powersaving functionality, which is not enabled by default
- **#6** - This pin is available on the breakout and is connected to FONA RTS in case you want to use flow control, which is not enabled by default

Other Pins!



- **RST** - this is the Reset pin, tie to ground to manually reset the AVR, as well as launch the bootloader manually
- **ARef** - the analog reference pin. Normally the reference voltage is the same as the chip logic voltage (3.3V) but if you need an alternative analog reference, connect it to this pin and select the external AREF in your firmware. Can't go higher than 3.3V!
- **Key** - this is by default tied to ground, cut the trace on the bottom and wire to a microcontroller pin to manually turn the module on and off. (Pulse low for a few seconds to change from on to off) This is the only way to truly disable the cellular module.
- **Mic+ and Mic-** connections for attaching an electret microphone for audio applications (external audio interface)
- **Spk+ and Spk-** connections for attaching a 8 ohm 1W speaker for audio applications (external audio interface)

FONA connections & LEDs



All the way to the right we have the cellular-only connection parts. Up top is a standard uFL connector, you attach your GSM antenna here

Below that is the bluetooth antenna (small white rectangle with red marking)

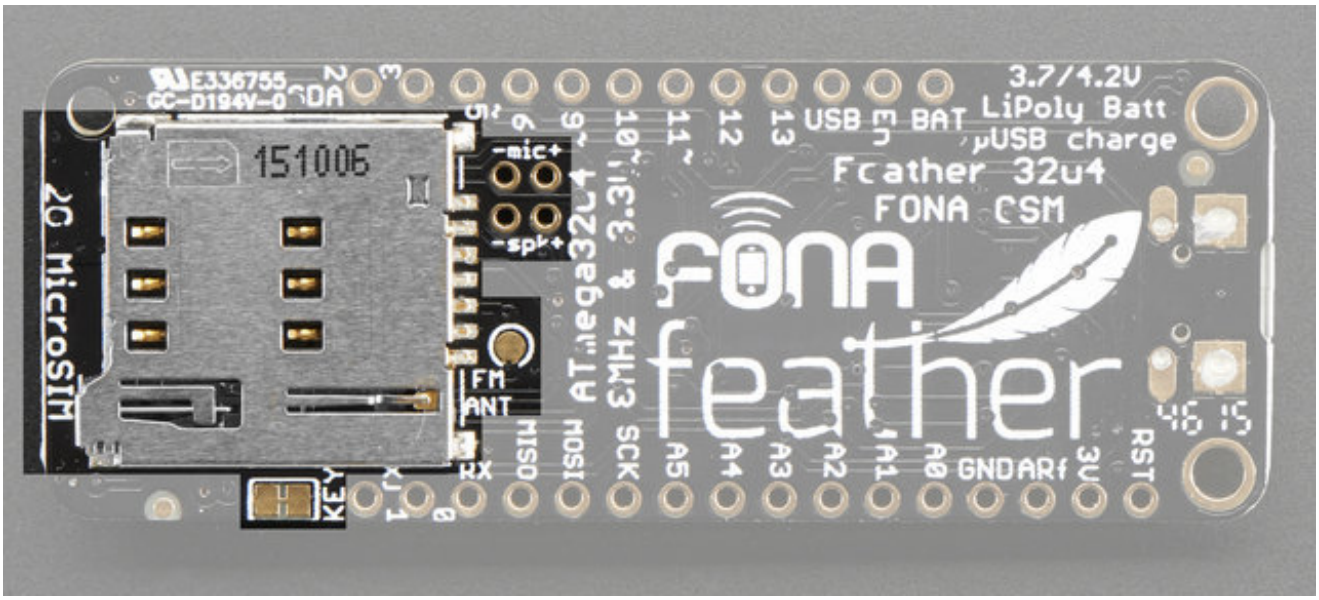
At the bottom are the two cell-status LEDs.

The middle **PWR** LED will light up green whenever the cellular module is active and has good power.

The **NET** LED will blink in blue let you know the status of the cellular connection You can use this for checking the current state without sending an AT command:

- **64ms on, 800ms off** - the module is running but hasn't made connection to the cellular network yet
- **64ms on, 3 seconds off** - the module has made contact with the cellular network and can send/receive voice and SMS
- **64ms on, 300ms off** - the GPRS data connection you requested is active

By watching the blinks you can get a visual feedback on whats going on



On the bottom is a microSIM push-push holder. Slot your microSIM in here and press in until it clicks

There's also a spot you can solder an antenna if you want to use the FM receiver capability.

Above that is nicely labeled Mic/Speaker pads

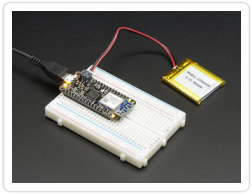
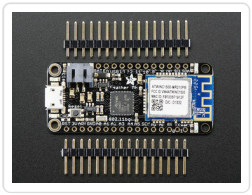
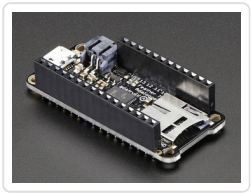
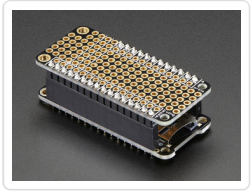
At the very bottom is the jumper for the **KEY** pad - cut this to let you control the module's on/off key manually. By default KEY is tied to ground so the module is always powered and on.

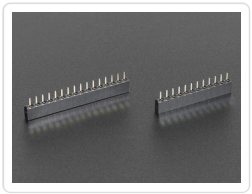
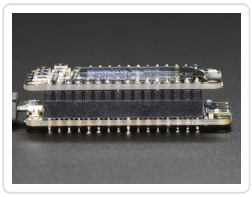
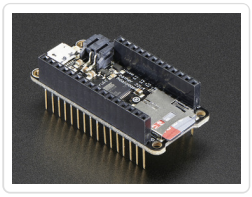
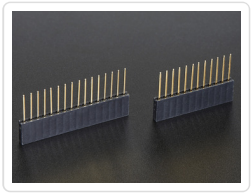
Assembly

We ship Feathers fully tested but without headers attached - this gives you the most flexibility on choosing how to use and configure your Feather

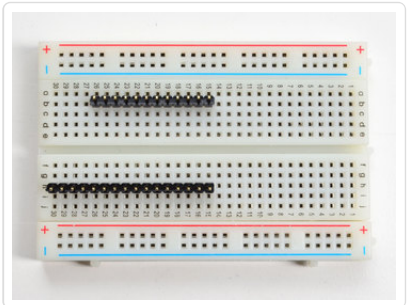
Header Options!

Before you go gung-ho on soldering, there's a few options to consider!

- 
  The first option is soldering in plain male headers, this lets you plug in the Feather into a solderless breadboard
- 
- 
  Another option is to go with socket female headers. This won't let you plug the Feather into a breadboard but it will let you attach featherwings very easily
- 

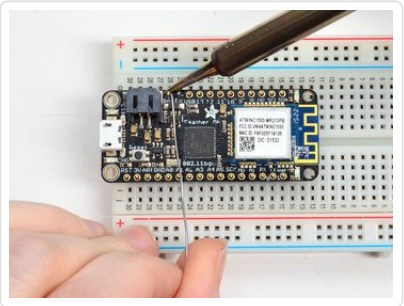
- 
  We also have 'slim' versions of the female headers, that are a little shorter and give a more compact shape
- 
- 
  Finally, there's the "Stacking Header" option. This one is sort of the best-of-both-worlds. You get the ability to plug into a solderless breadboard *and* plug a featherwing on top. But its a little bulky
- 

Soldering in Plain Headers

- 

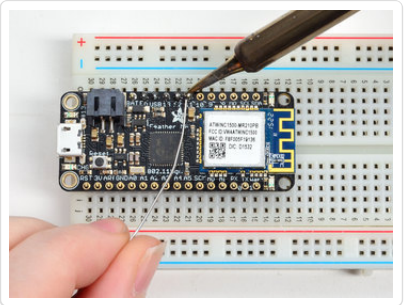
Prepare the header strip:

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:

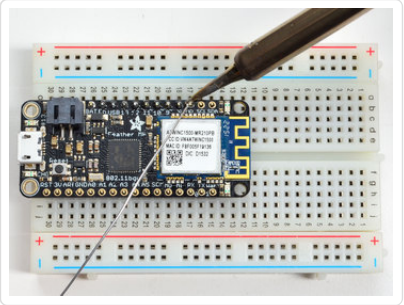
- Place the breakout board over the pins so that the short pins poke through the breakout pads

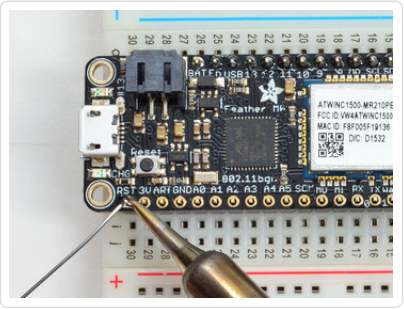
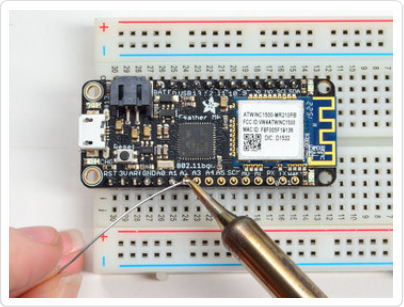
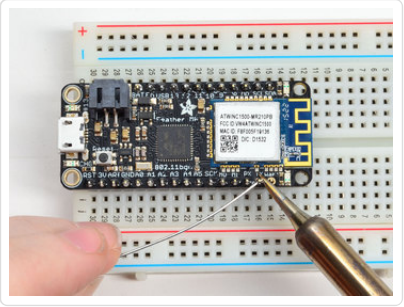
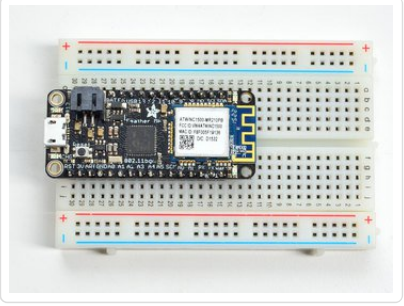


And Solder!

- 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).

- 

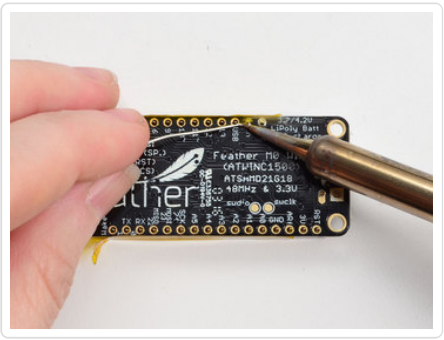
-  Solder the other strip as well.
- 
- 
-  You're done! Check your solder joints visually and continue onto the next steps

Soldering on Female Header

- 

Tape In Place

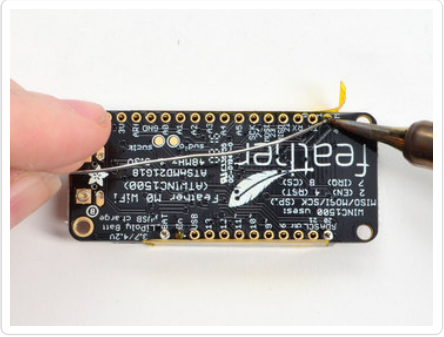
For sockets you'll want to tape them in place so when you flip over the board they don't fall out

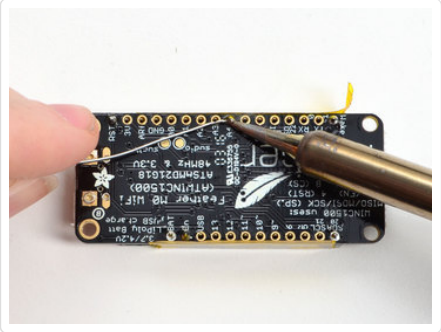
- 

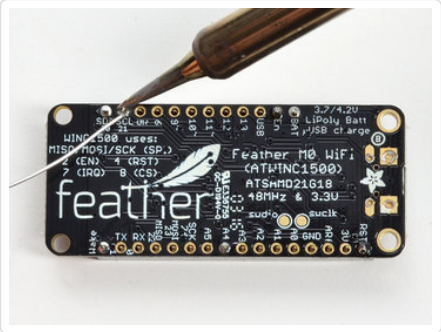
- 

Flip & Tack Solder

After flipping over, solder one or two points on each strip, to 'tack' the header in place

- 

- 

- 

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).

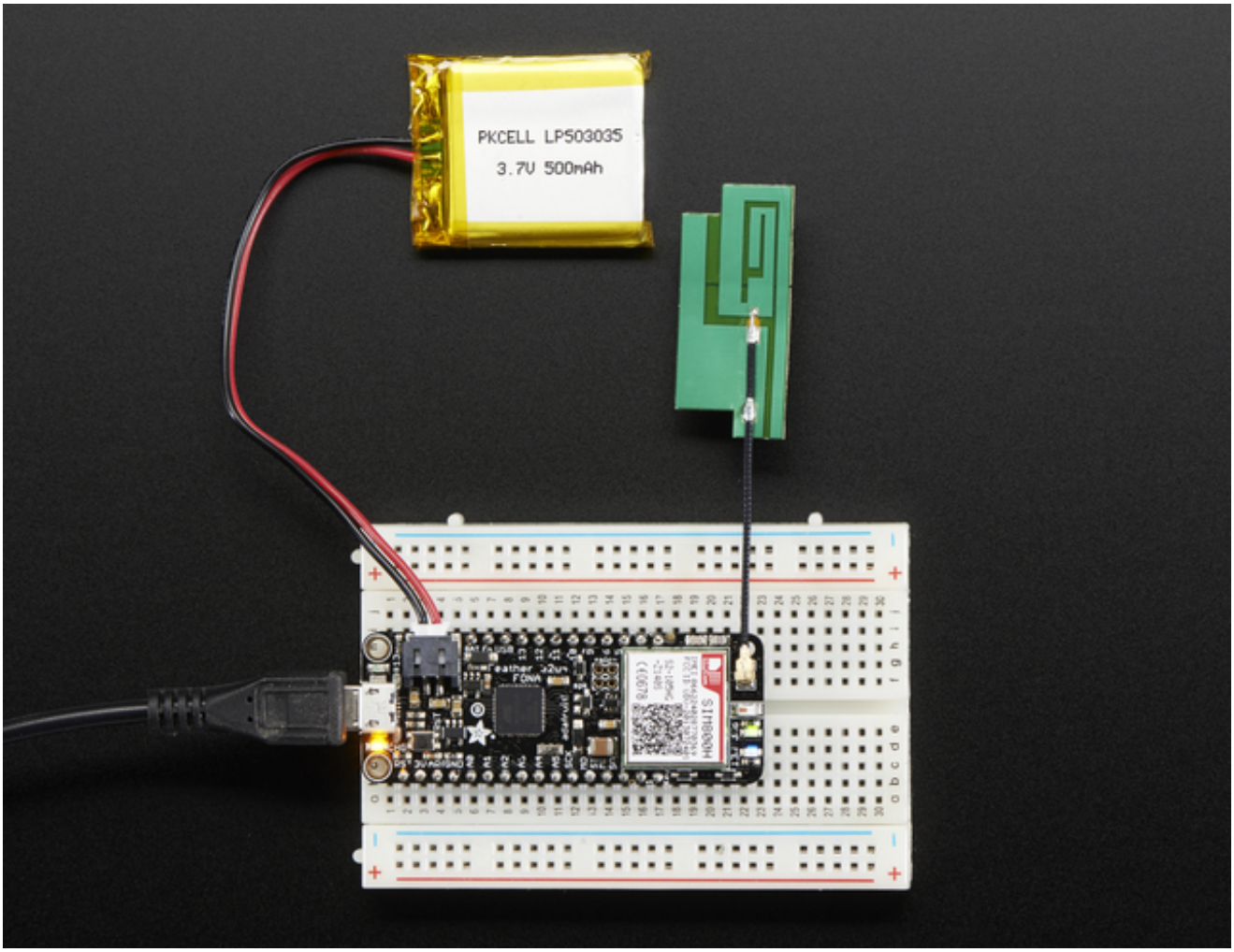
- 

- 

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

- 

Power Management



Battery + USB Power

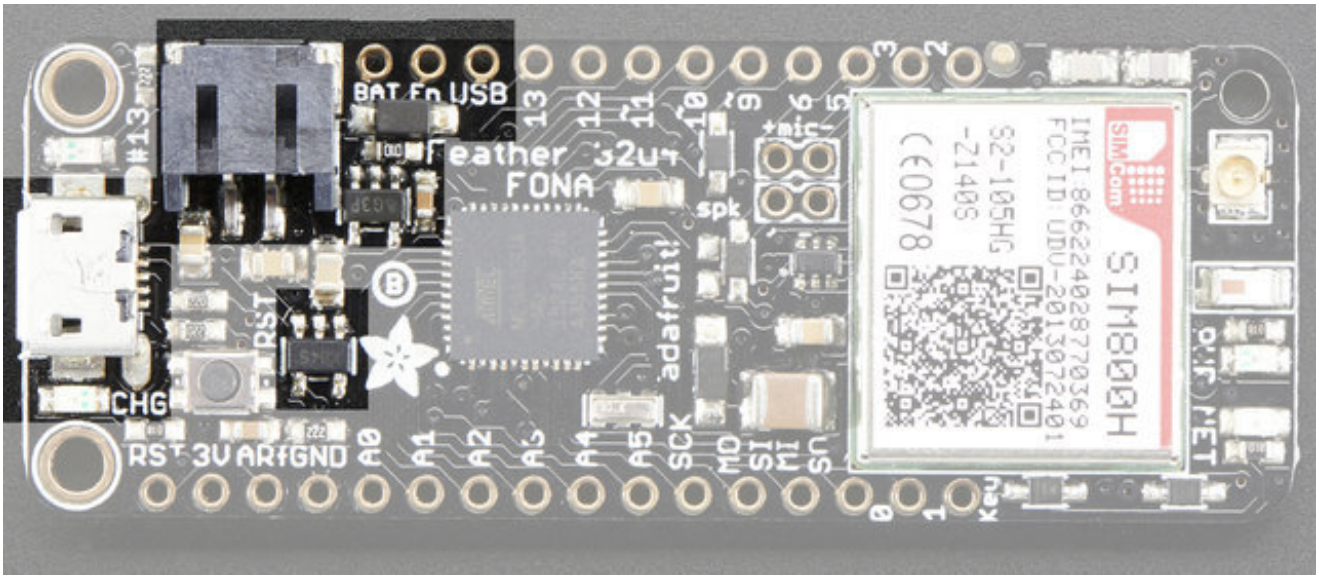
Feather FONA is a little special and different with how it manages battery and USB power. Whereas other Feathers can run direct from 3.3V and thus from USB or battery...

> You must have a Lipoly/LiIon battery plugged in at all times for using the Feather FONA

This is because the cellular module cannot run off of 3.3V, and it has thin but common spikes of an amp or two when connecting/sending data on the cellular network. We could have gone with a huge voltage regulator but instead we decided to just power the FONA module direct from the lipoly battery. If this isn't plugged in, the battery charger ends up trying to source an amp, which it cant, and the board will reset

So, yeah. Use a battery! It will keep topped up when using USB, and allow the cellular module to draw current spikes without issue

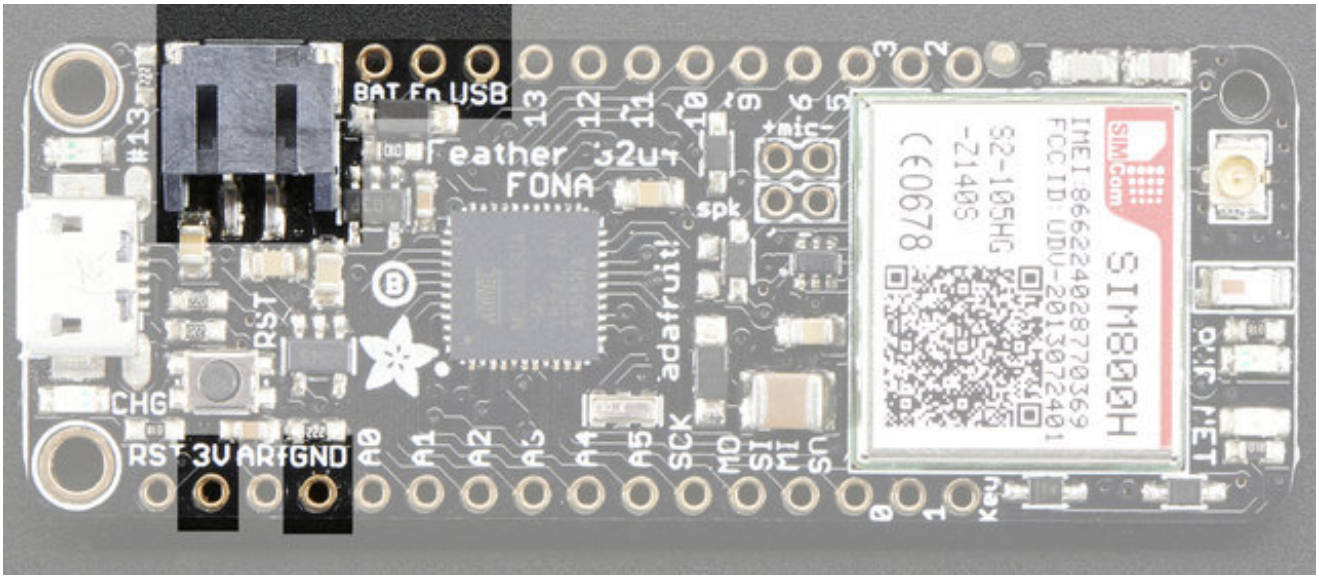
> The JST connector polarity is matched to Adafruit LiPoly batteries. Using wrong polarity batteries can destroy your Feather



The above shows the Micro USB jack (left), Lipoly JST jack (top left), as well as the 3.3V regulator and changeover diode (just to the right of the JST jack) and the Lipoly charging circuitry (to the right of the Reset button). There's also a **CHG** LED, which will light up while the battery is charging. This LED might also flicker if the battery is not connected or when the cellular module is in action

Power supplies

You have a lot of power supply options here! We bring out the **BAT** pin, which is tied to the lipoly JST connector, as well as **USB** which is the +5V from USB if connected. We also have the **3V** pin which has the output from the 3.3V regulator. We use a 500mA peak AP2112. While you can get 500mA from it, you can't do it continuously from 5V as it will overheat the regulator. It's fine for, say, powering an ESP8266 WiFi chip or XBee radio though, since the current draw is 'spiky' & sporadic. Note that the regulator *doesn't* power the cellular module, that's directly powered from VBAT



Measuring Battery

If you're running off of a battery, chances are you wanna know what the voltage is at! That way you can tell when the battery needs recharging. Lipoly batteries are 'maxed out' at 4.2V and stick around 3.7V for much of the battery life, then slowly sink down to 3.2V or so before the protection circuitry cuts it off. By measuring the voltage you can quickly tell when you're heading below 3.7V

Other Feather's have a resistor divider to read the battery voltage. We decided to skip this and instead let you read the battery voltage via the cellular module using the **AT+CBC** command, which will give you the battery voltage in millivolts

ENable pin

If you'd like to turn off the 3.3V regulator, you can do that with the **EN**(able) pin. Simply tie this pin to **Ground** and it will disable the 3V regulator. The **BAT** and **USB** pins will still be powered

Note that this will not disable power to the cellular module! If you want to depower the cell module, cut the KEY trace on the bottom of the board, wire KEY to an unused pad, and toggle the pin low for 100ms to completely turn on/off the module

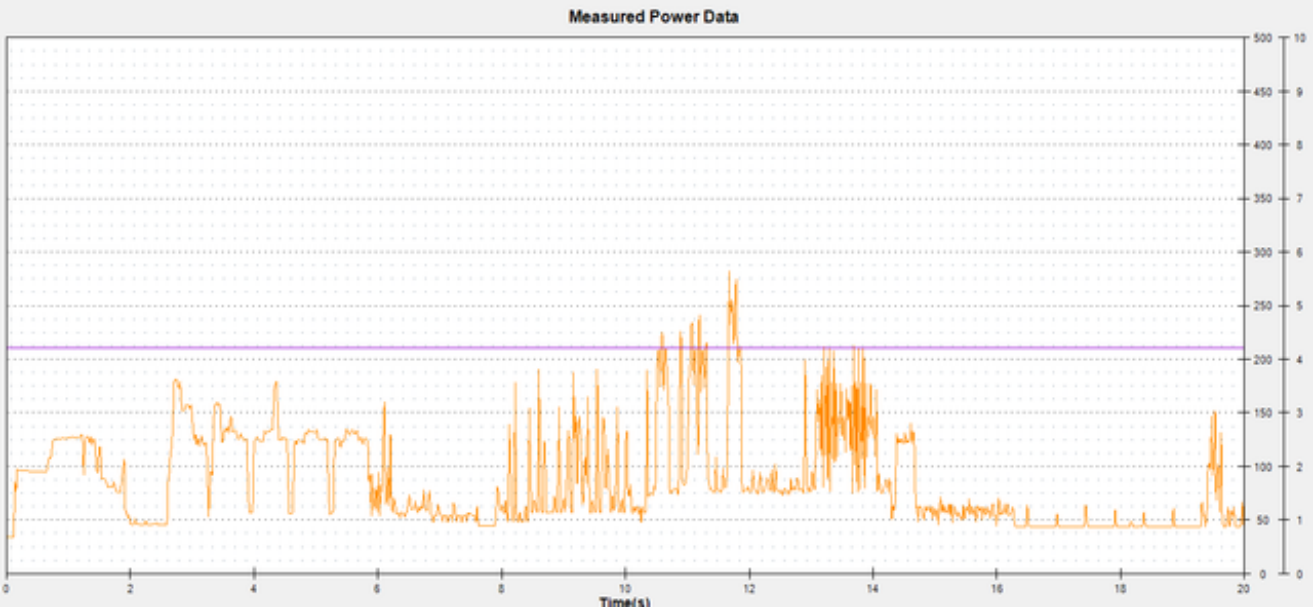
Cellular Power Usage

If you think WiFi is power hungry, you will be surprised at how much power draw you'll need to manage with a cellular module.

Here's some power traces for common events with a cellular module:

Turning on the FONA Feather

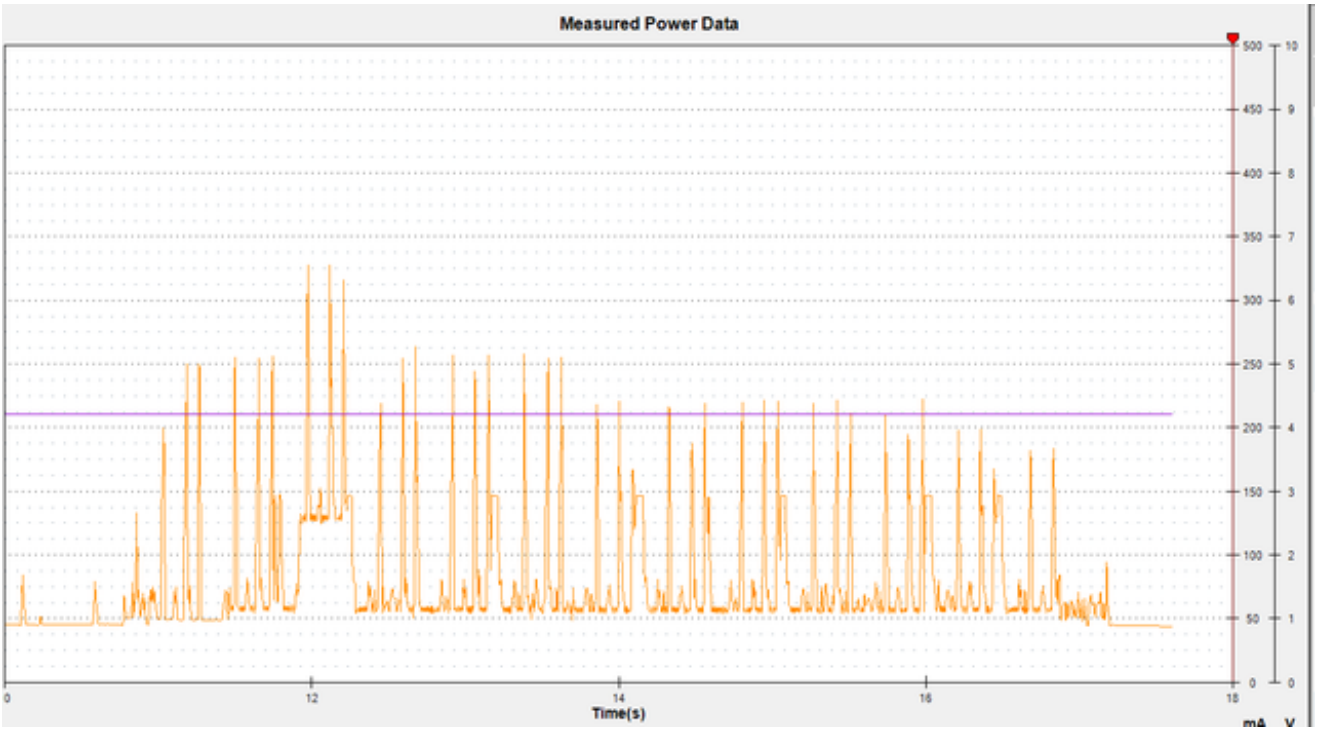
Booting cell module + connecting to network



Sending an SMS

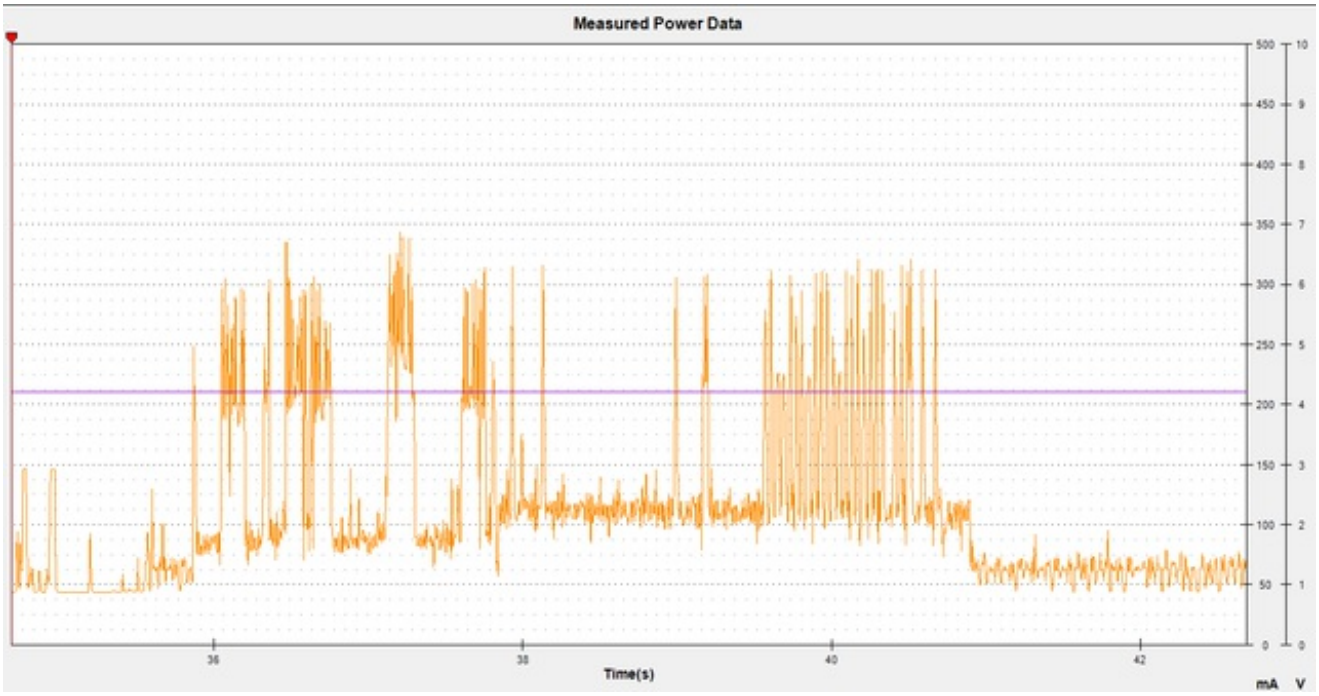
send SMS: 6.5s, 150uAh, 300mW, 52mA

recv sms: 6.5s, 140uAh, 330mW, 78mA



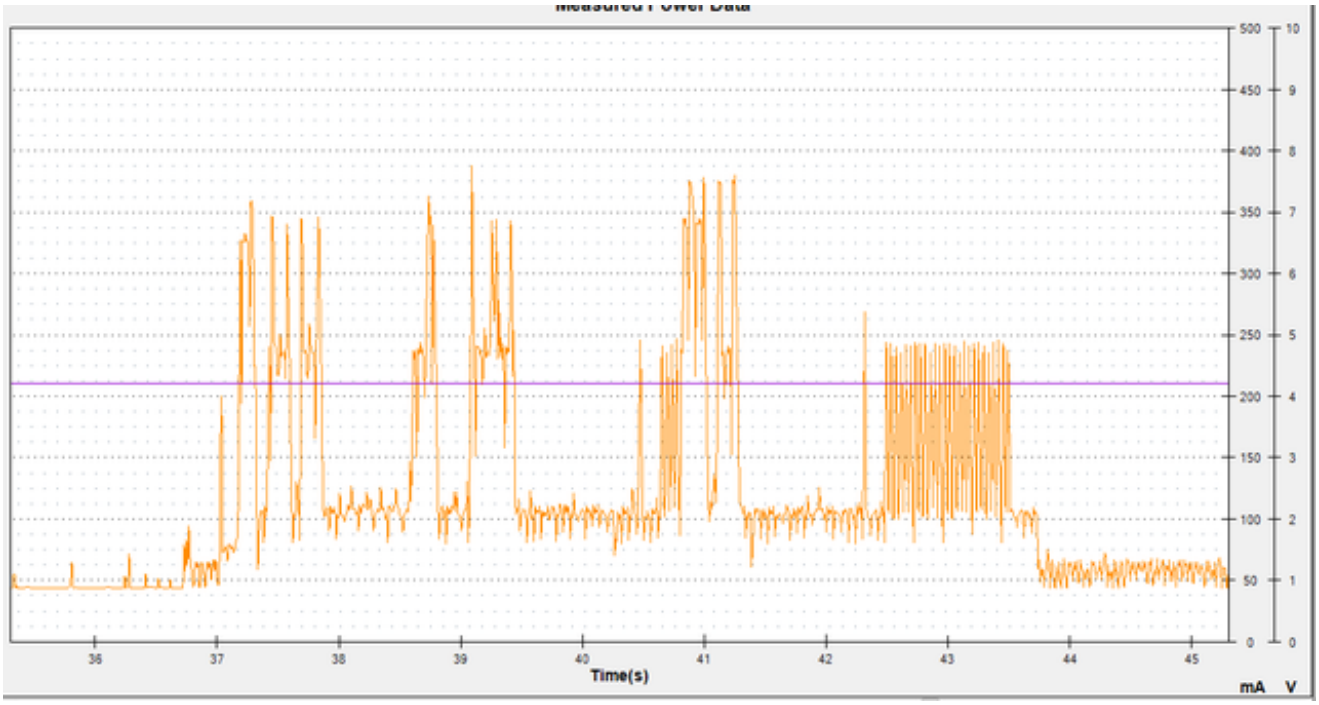
Enabling GPRS

enabling GPRS: about 8 seconds, 850uAh, 300mW, 70mA avg

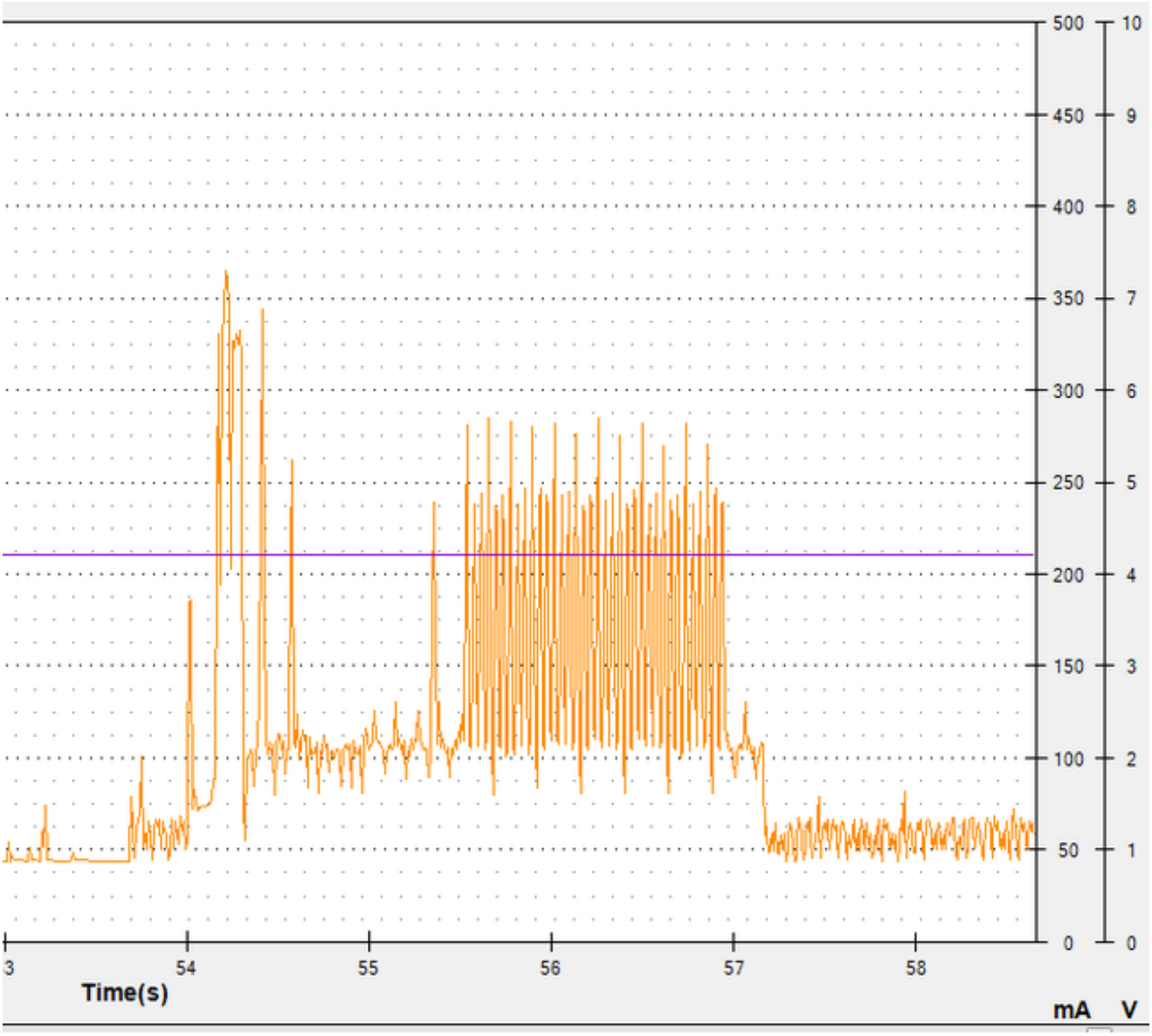


TCPIP connection

grab mini webpage: 4.5 sec, 203uAh, 650mW, 150mA avg

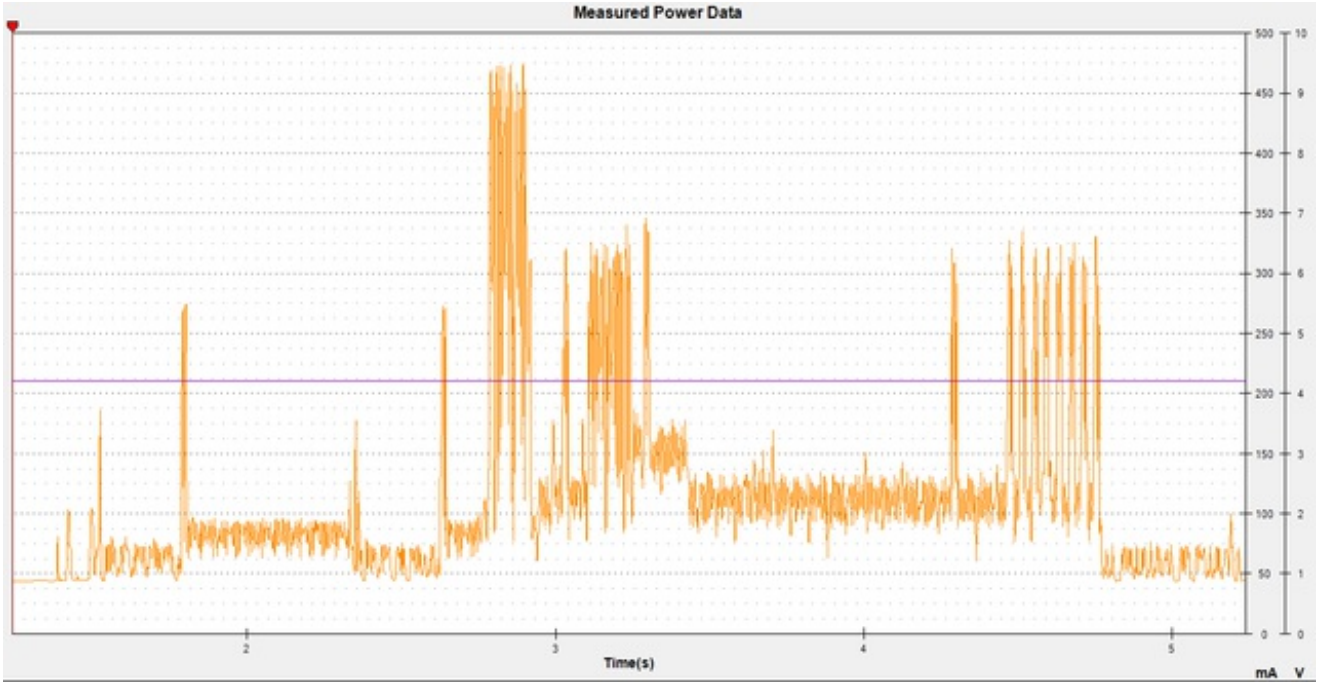


Sending an MQTT packet (about 200 bytes)

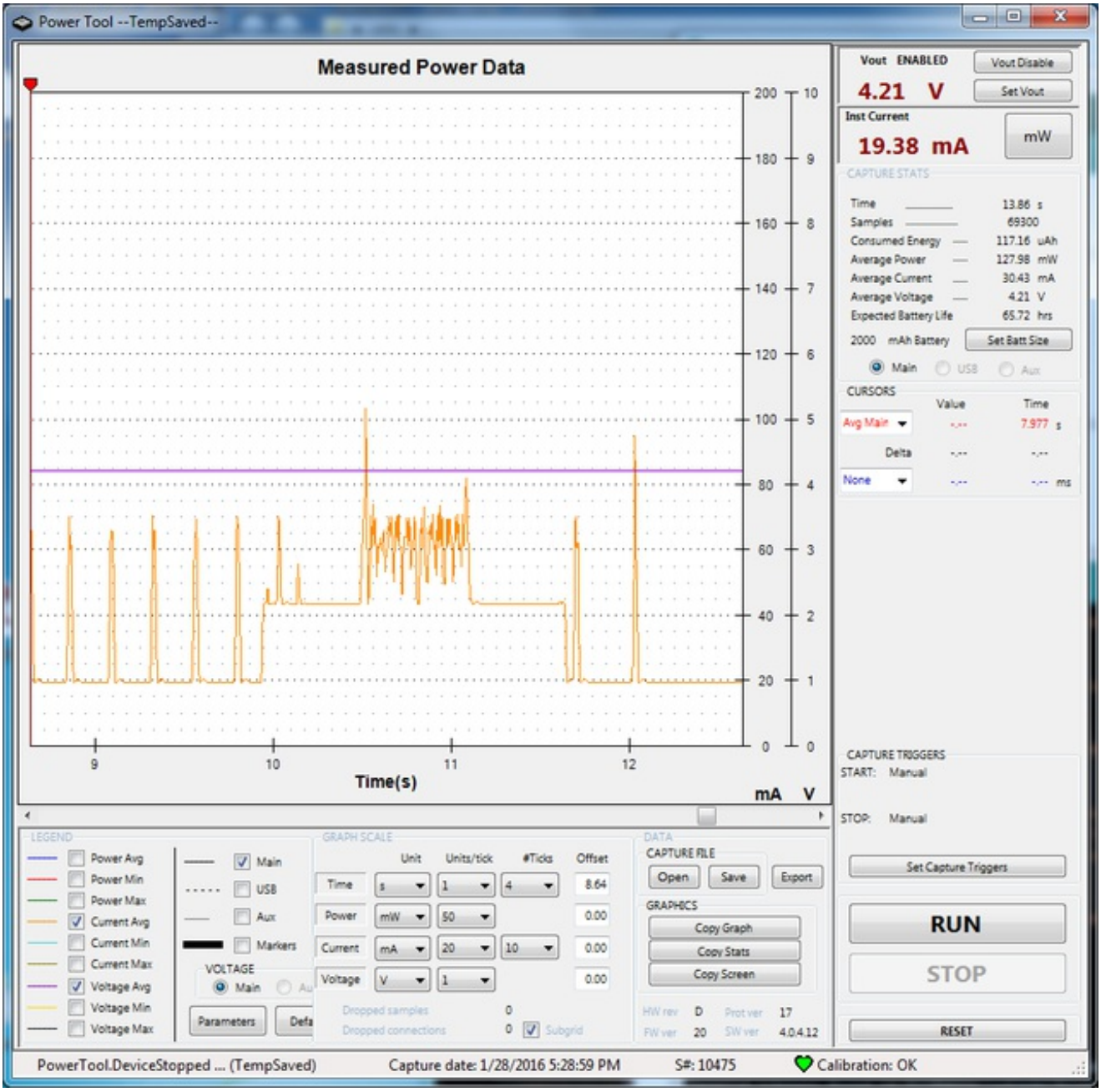


Disabling GPRS

disabling GPRS: about 4 seconds, 120uAh, 480mW, 113 mA avg



You can put the FONA into sleep mode (with the **AT+CSCLK** command) which will drop the current draw but keep the cellular connection open so you can still receive an SMS and/or wakeup quickly.



Note that the quiescent current drops from 40mA to 20mA and of that 20mA, about ~12mA is the ATmega32u4. Like we said, the best way to really reduce power for long-term usage is to completely turn off the module with KEY

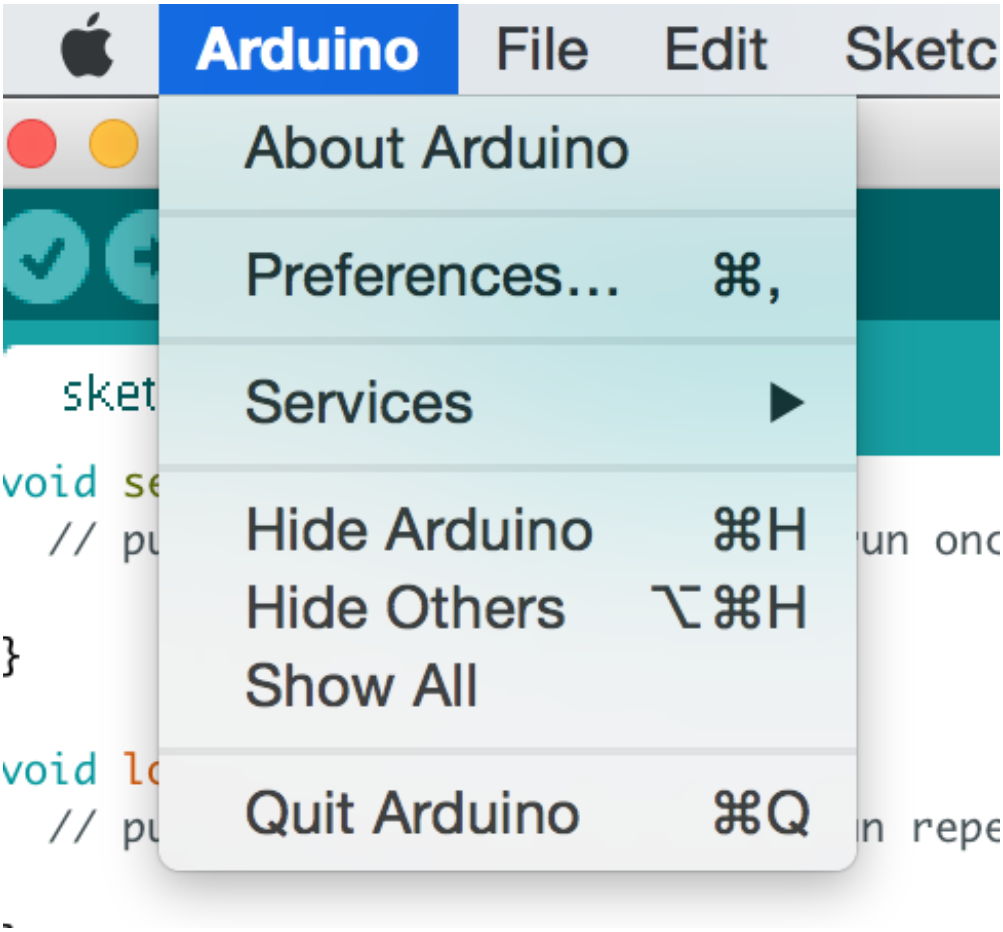
Arduino IDE Setup

The first thing you will need to do is to download the latest release of the Arduino IDE. You will need to be using **version 1.6.4** or higher for this guide.

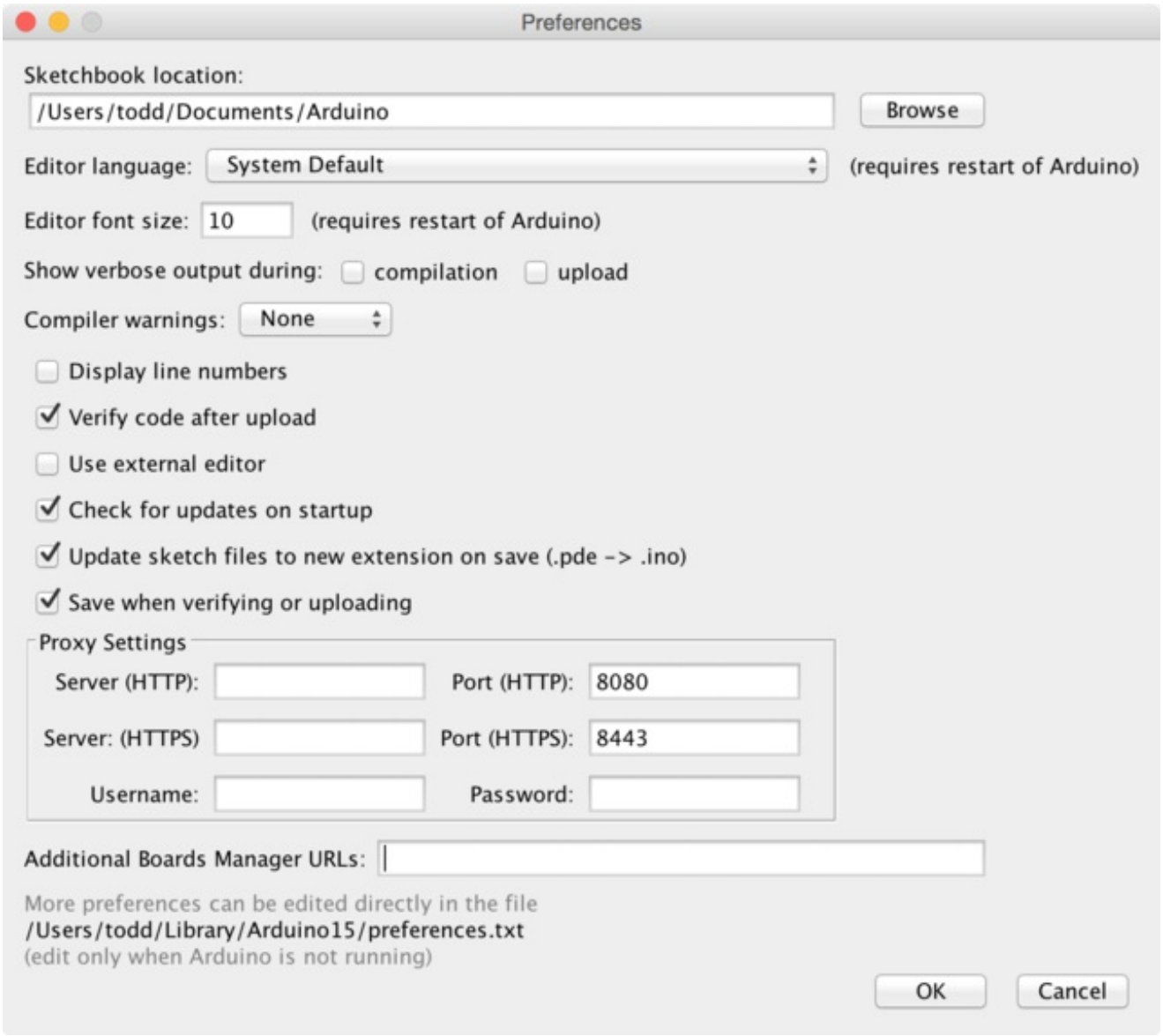
Arduino IDE v1.6.4+ Download

http://adafru.it/f1P

After you have downloaded and installed **v1.6.4**, you will need to start the IDE and navigate to the **Preferences** menu. You can access it from the **File** menu in *Windows* or *Linux*, or the **Arduino** menu on *OS X*.



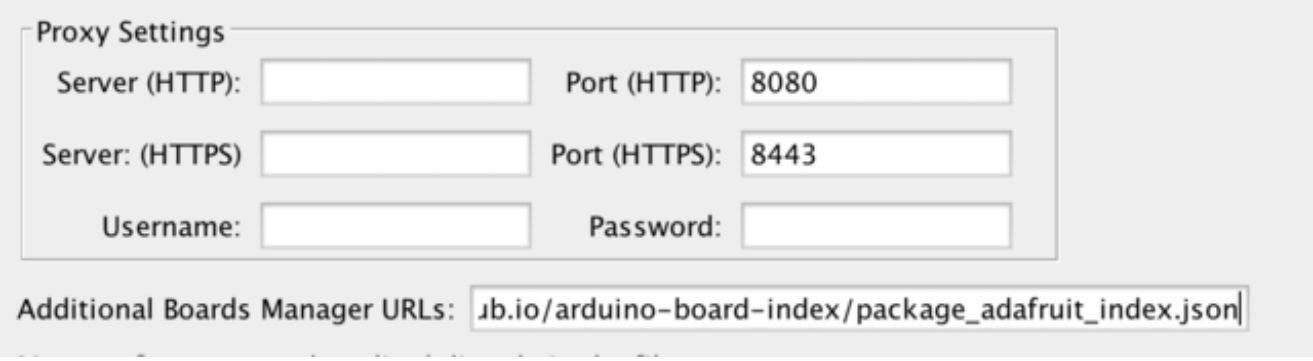
A dialog will pop up just like the one shown below.



We will be adding a URL to the new **Additional Boards Manager URLs** option. The list of URLs is comma separated, and *you will only have to add each URL once.* New Adafruit boards and updates to existing boards will automatically be picked up by the Board Manager each time it is opened. The URLs point to index files that the Board Manager uses to build the list of available & installed boards.

To find the most up to date list of URLs you can add, you can visit the list of third party board URLs on the Arduino IDE wiki (http://adafru.it/f7U). We will only need to add one URL to the IDE in this example, but ***you can add multiple URLS by separating them with commas***. Copy and paste the link below into the **Additional Boards Manager URLs** option in the Arduino IDE preferences.

```
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
```



Make sure you remove the apt.adafruit.com proxy setting from the Arduino preferences if you have previously added it.

Here's a short description of each of the Adafruit supplied packages that will be available in the Board Manager when you add the URL:

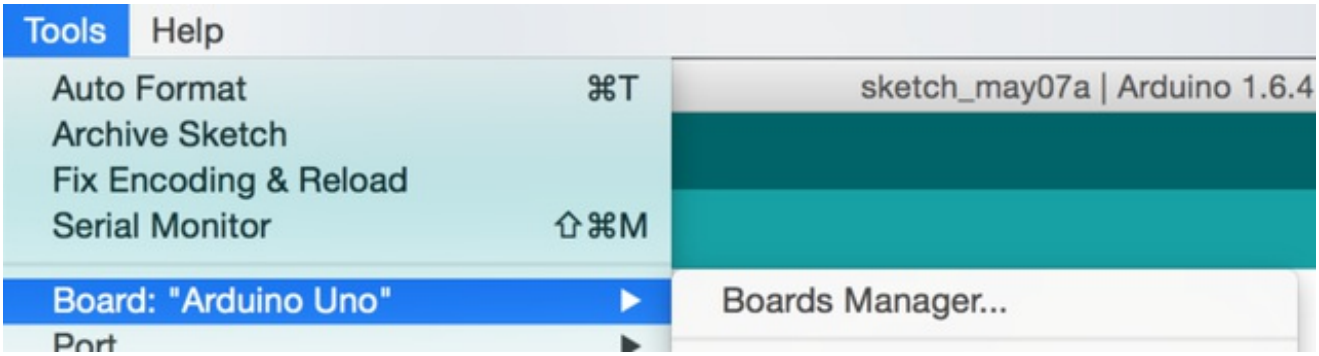
- **Adafruit AVR Boards** - Includes support for Flora, Gemma, Feather 32u4, Trinket, & Trinket Pro.
- **Adafruit SAMD Boards** - Includes support for Feather M0
- **Arduino Leonardo & Micro MIDI-USB** - This adds MIDI over USB support for the Flora, Feather 32u4, Micro and Leonardo using the arcore project (http://adafru.it/eSI).

Click **OK** to save the new preference settings. Next we will look at installing boards with the Board Manager.

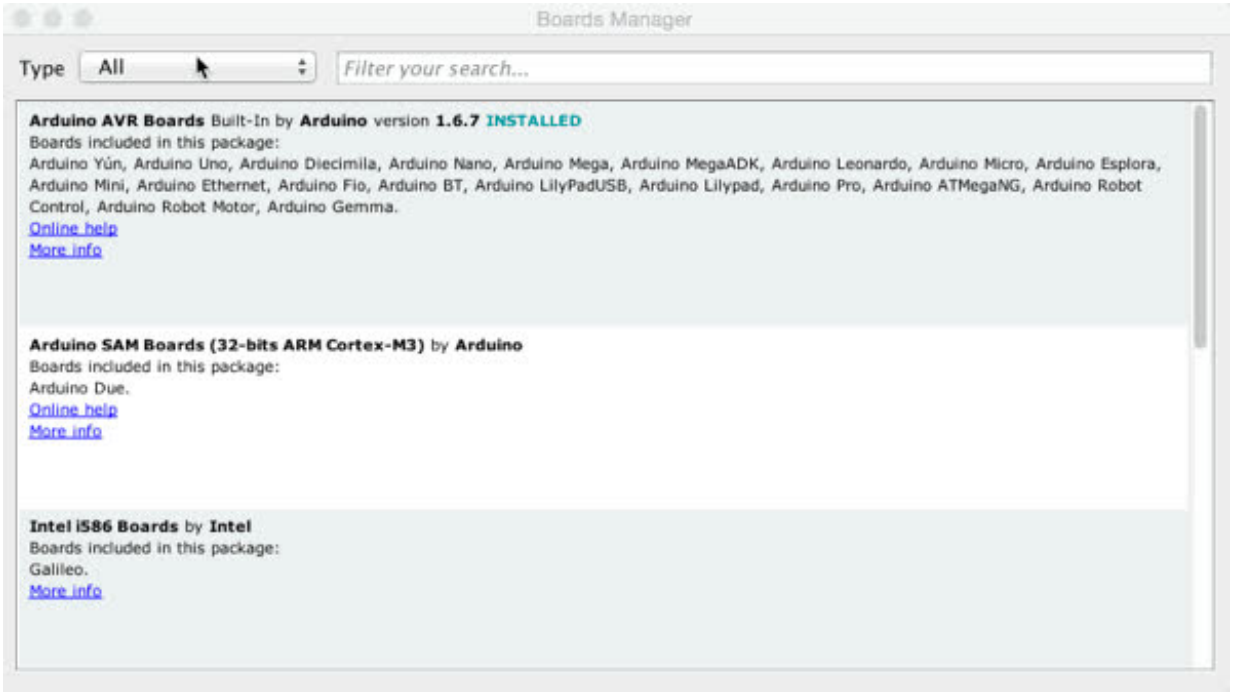
Using with Arduino IDE

Since the Feather 32u4 uses an ATmega32u4 chip running at 8 MHz, you can pretty easily get it working with the Arduino IDE. Many libraries (including the popular ones like NeoPixels and display) work great with the '32u4 and 8 MHz clock speed.

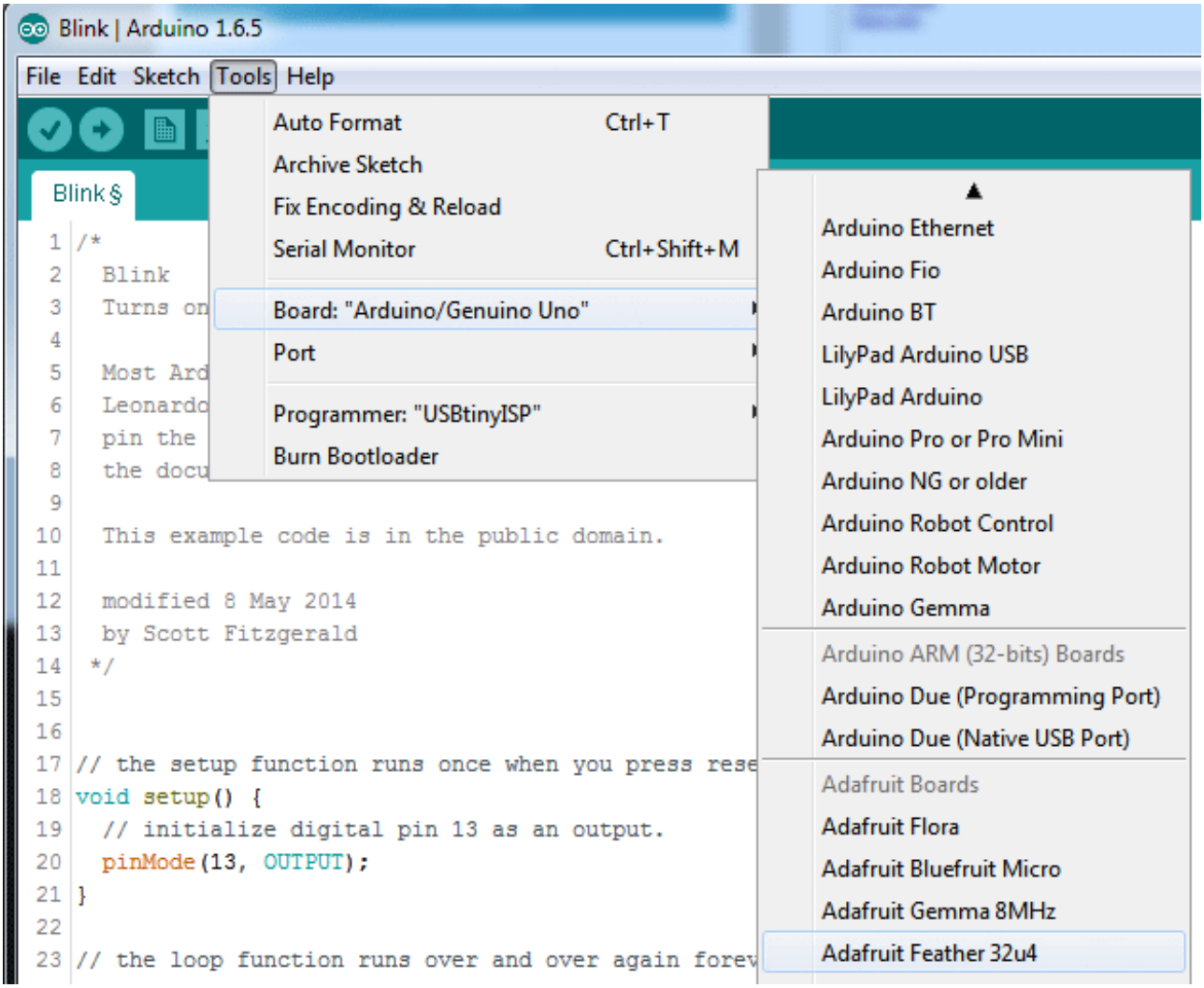
Now that you have added the appropriate URLs to the Arduino IDE preferences, you can open the **Boards Manager** by navigating to the **Tools->Board** menu.



Once the Board Manager opens, click on the category drop down menu on the top left hand side of the window and select **Contributed**. You will then be able to select and install the boards supplied by the URLs added to the prefrences. In the example below, we are installing support for ***Adafruit AVR Boards***, but the same applies to all boards installed with the Board Manager.



Next, **quit and reopen the Arduino IDE** to ensure that all of the boards are properly installed. You should now be able to select and upload to the new boards listed in the **Tools->Board** menu.



Install Drivers (Windows Only)

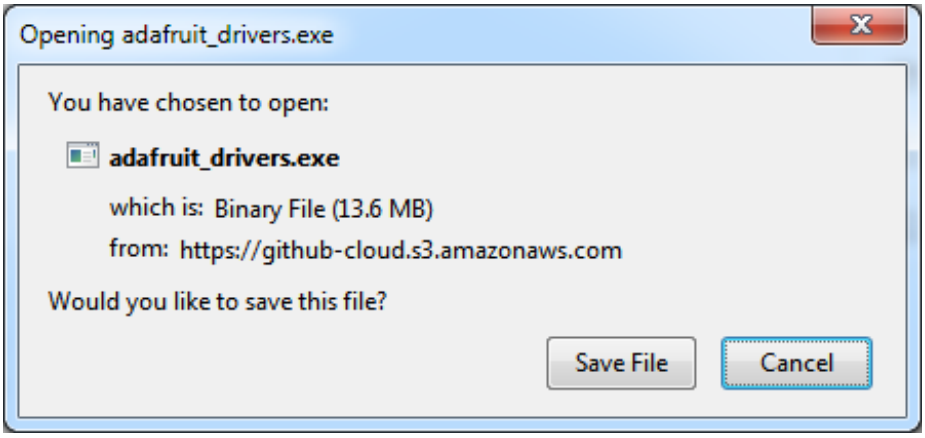
When you plug in the Feather, you'll need to possibly install a driver

Click below to download our Driver Installer

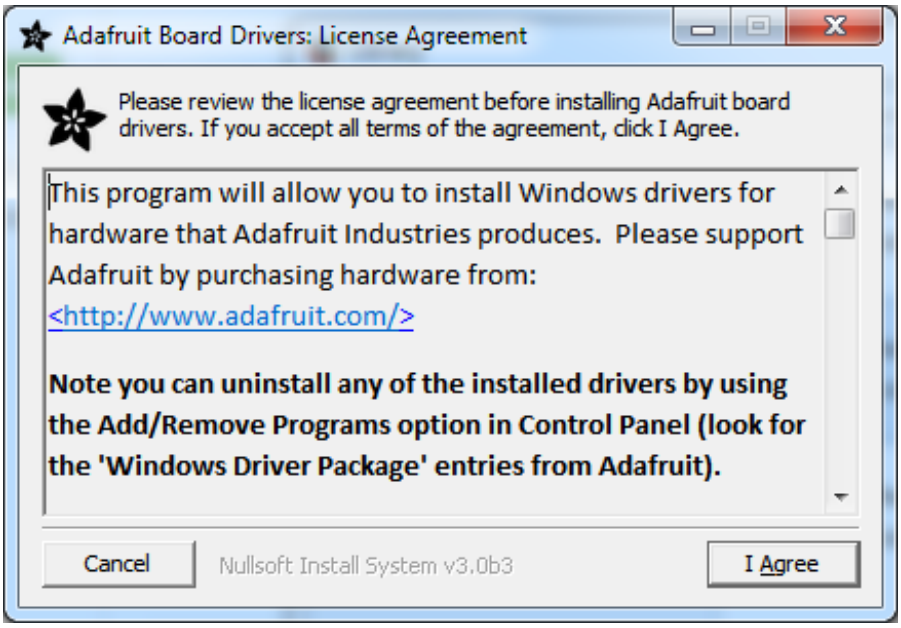
Download Adafruit Drivers Installer

http://adafru.it/mai

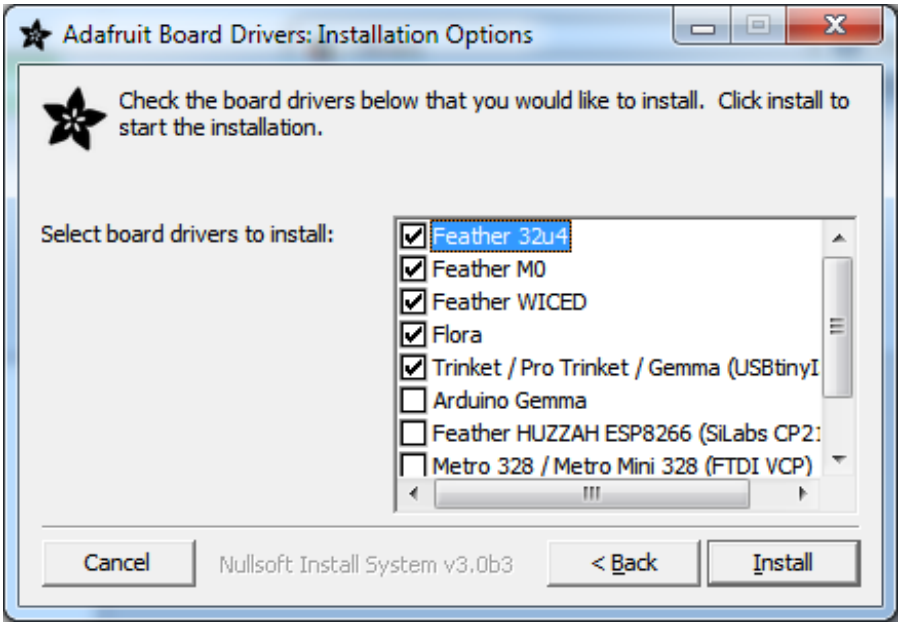
Download and run the installer



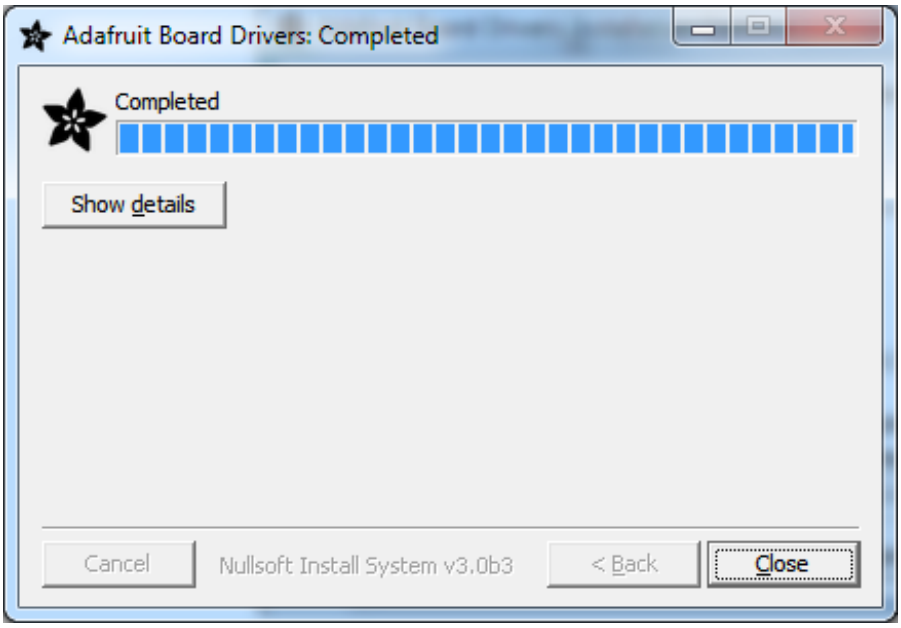
Run the installer! Since we bundle the SiLabs and FTDI drivers as well, you'll need to click through the license



Select which drivers you want to install:



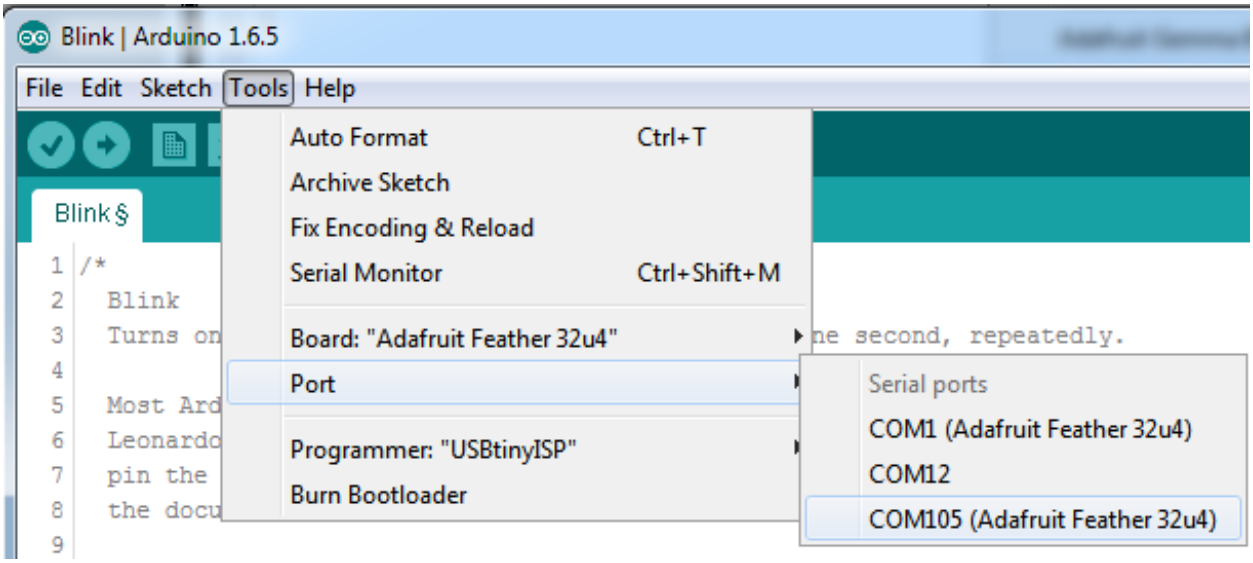
Click **Install** to do the installin'



Blink

Now you can upload your first blink sketch!

Plug in the Feather 32u4 and wait for it to be recognized by the OS (just takes a few seconds). It will create a serial/COM port, you can now select it from the dropdown, it'll even be 'indicated' as Feather 32u4!



Now load up the Blink example

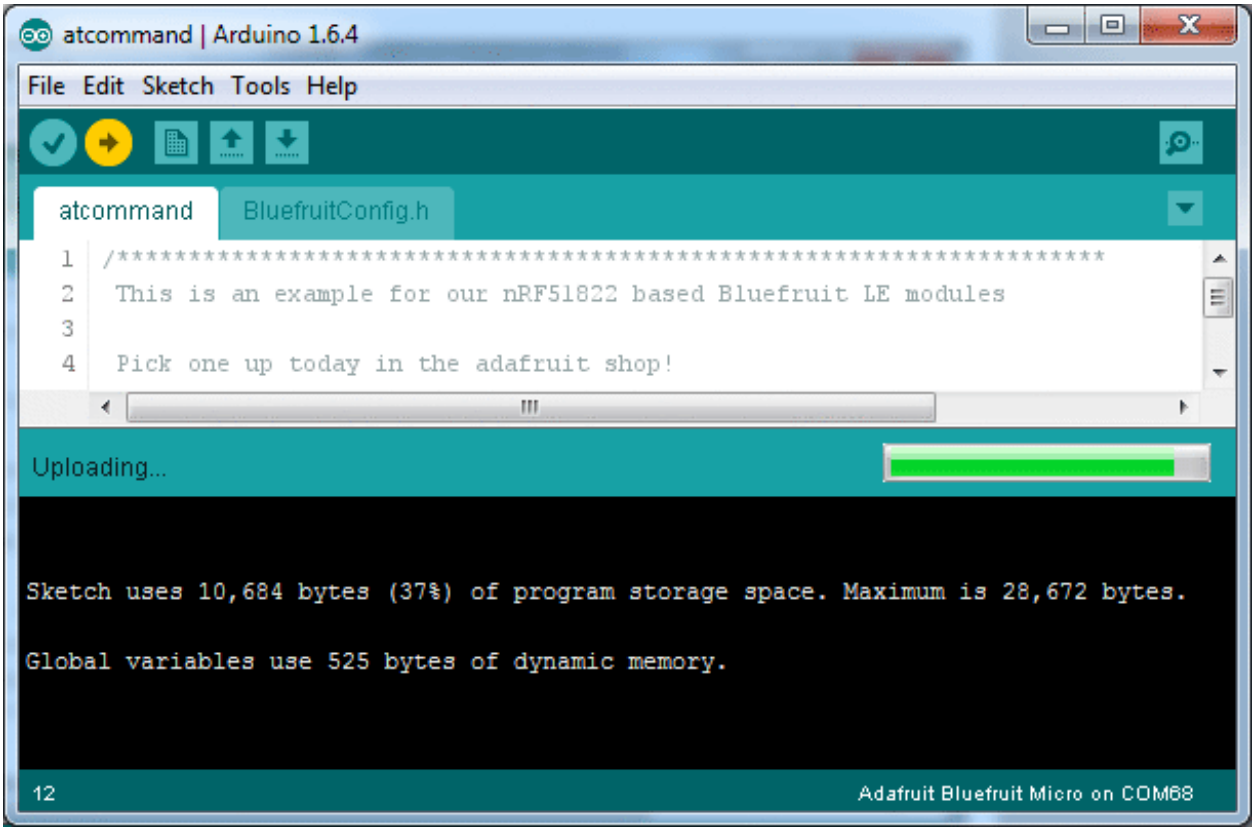
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

And click upload! That's it, you will be able to see the LED blink rate change as you adapt the **delay()** calls.

Manually bootloading

If you ever get in a 'weird' spot with the bootloader, or you have uploaded code that crashes and doesn't auto-reboot into the bootloader, click the **RST** button to get back into the bootloader. The red LED will pulse, so you know that its in bootloader mode. Do the reset button press right as the Arduino IDE says its attempting to upload the sketch, when you see the Yellow Arrow lit and the **Uploading...** text in the status bar.



Don't click the reset button **before** uploading, unlike other bootloaders you want this one to run at the time Arduino is trying to upload

Ubuntu & Linux Issue Fix

Note if you're using Ubuntu 15.04 (or perhaps other more recent Linux distributions) there is an issue with the modem manager service which causes the Bluefruit LE micro to be difficult to program. If you run into errors like "device or resource busy", "bad file descriptor", or "port is busy" when attempting to program then you are hitting this issue. (http://adafru.it/fP6)

The fix for this issue is to make sure Adafruit's custom udev rules are applied to your system. One of these rules is made to configure modem manager not to touch the Bluefruit Micro board and will fix the programming difficulty issue. Follow the steps for installing Adafruit's udev rules on this page. (http://adafru.it/iOE)

FONA Test

Download Adafruit_FONA

To begin reading sensor data, you will need to download Adafruit_FONA Library from our github repository (http://adafru.it/dDC). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit_FONA library

http://adafru.it/dDD

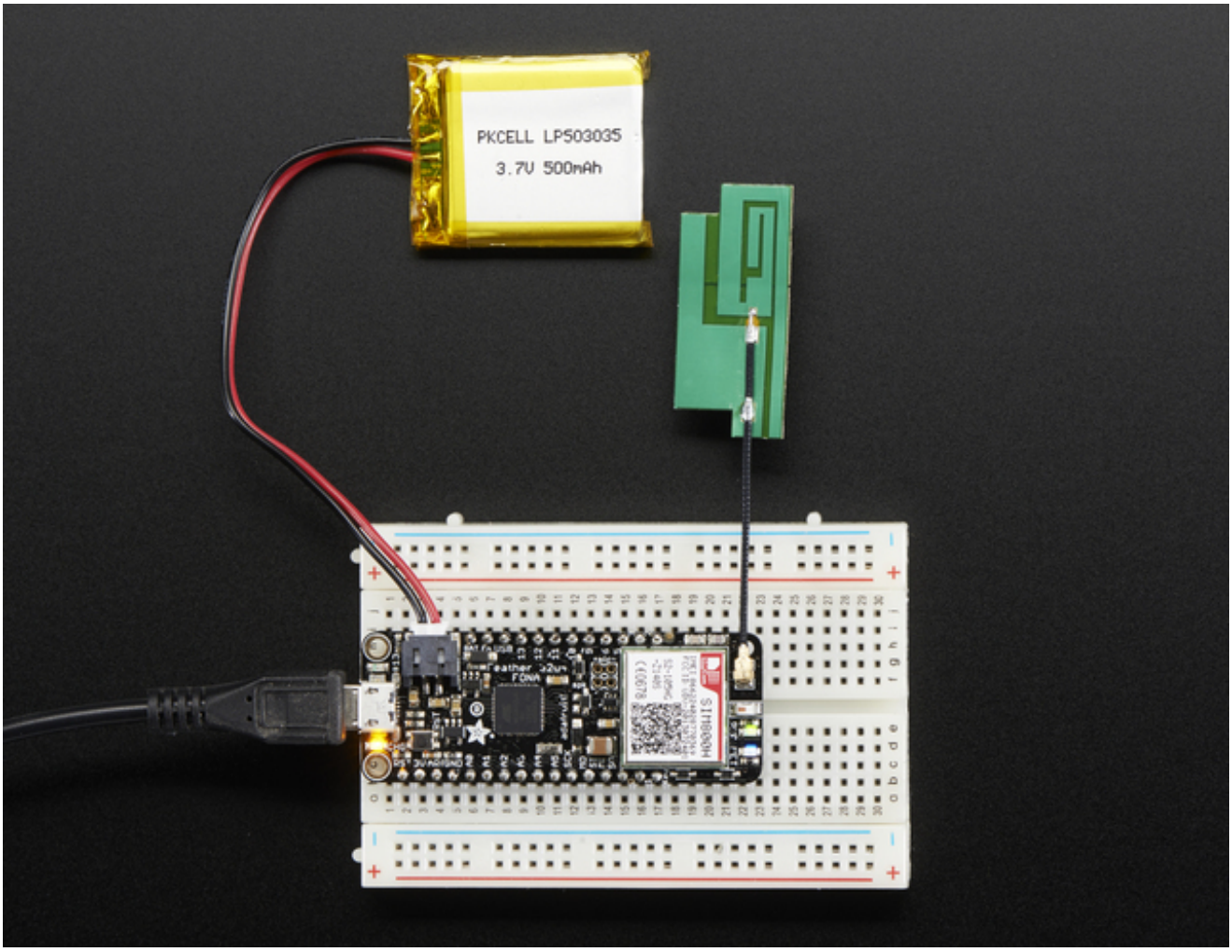
Rename the uncompressed folder **Adafruit_FONA** and check that the **Adafruit_FONA** folder contains **Adafruit_FONA.cpp** and **Adafruit_FONA.h**

Place the **Adafruit_FONA** 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 (http://adafru.it/aYM)

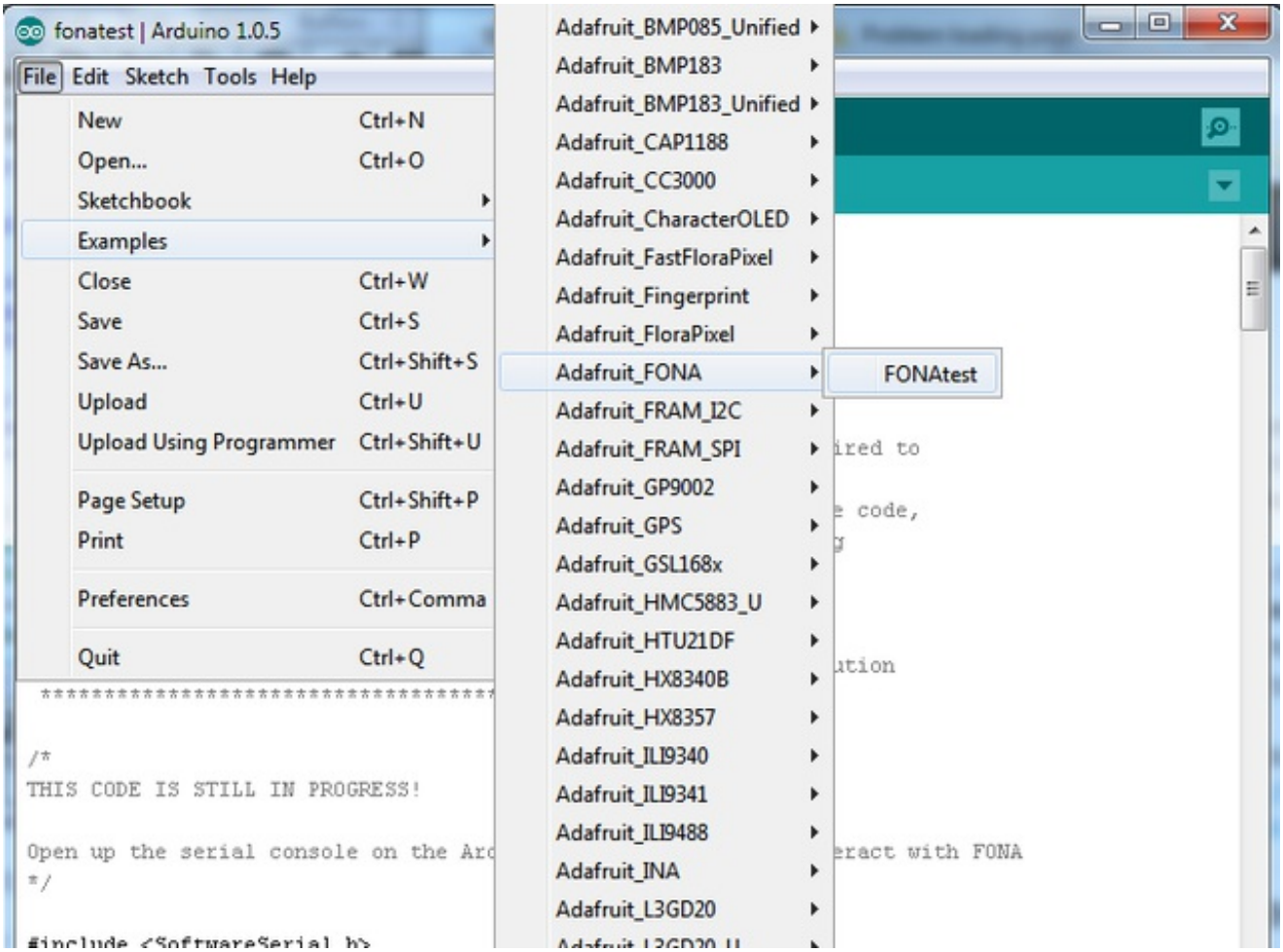
Wire up and Power!

Insert a micro SIM, attach a GSM antenna, then connect battery and micro USB



Load Demo

Open up **File->Examples->Adafruit_FONA->FONAtest**



Don't upload the sketch yet!

You'll need to make a simple change to the sketch. At the top find these lines:

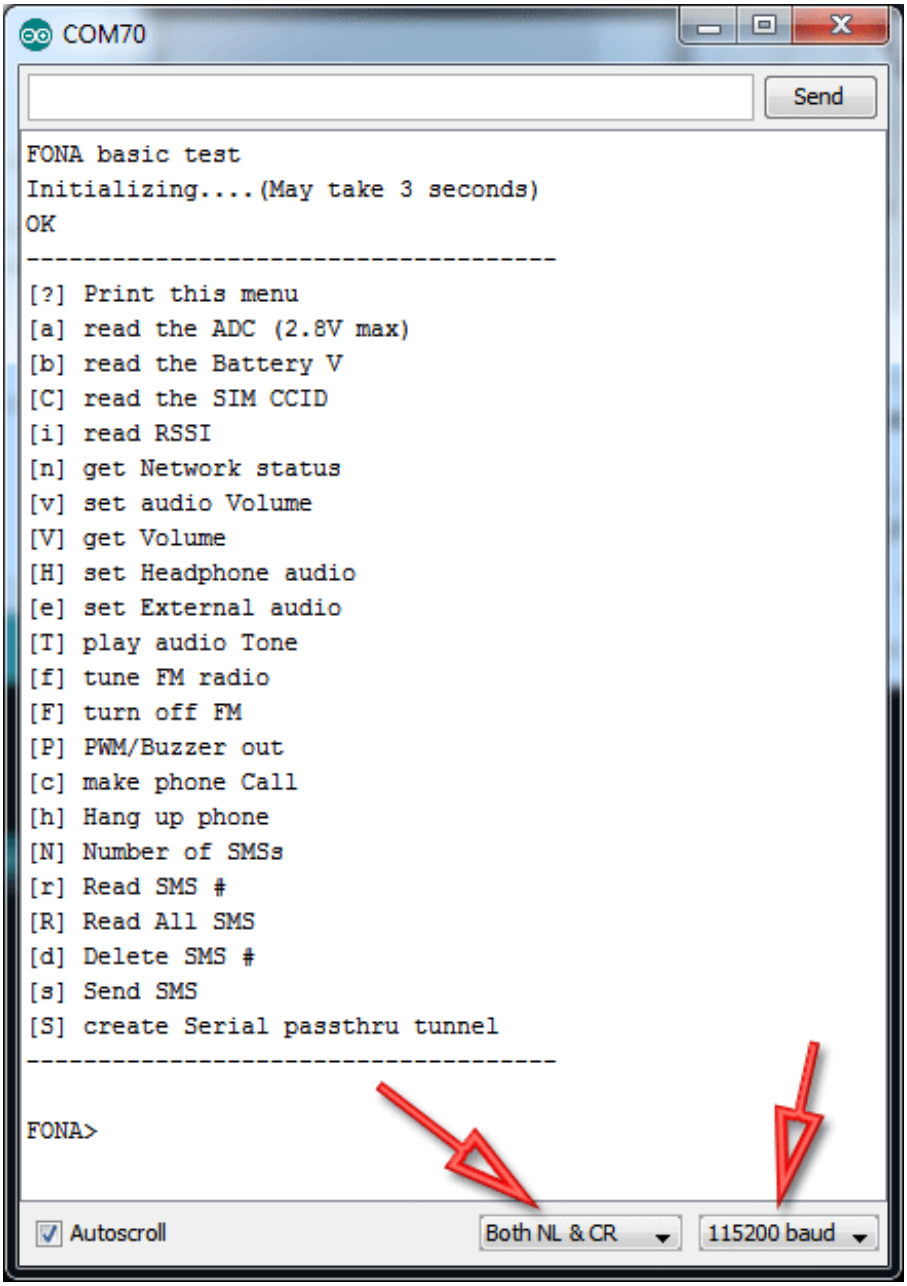
```
#include "Adafruit_FONA.h"

#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4
```

and change them to:

```
#define FONA_RX  9
#define FONA_TX  8
#define FONA_RST 4
#define FONA_RI  7
```

Once uploaded to your Arduino, open up the serial console at **115200 baud speed** to begin the tester sketch



Make sure you also have **Both NL & CR** for the serial command sender option. This means when you send data to the Arduino via the console, it will put a newline/return at the end.

Using the Test Sketch

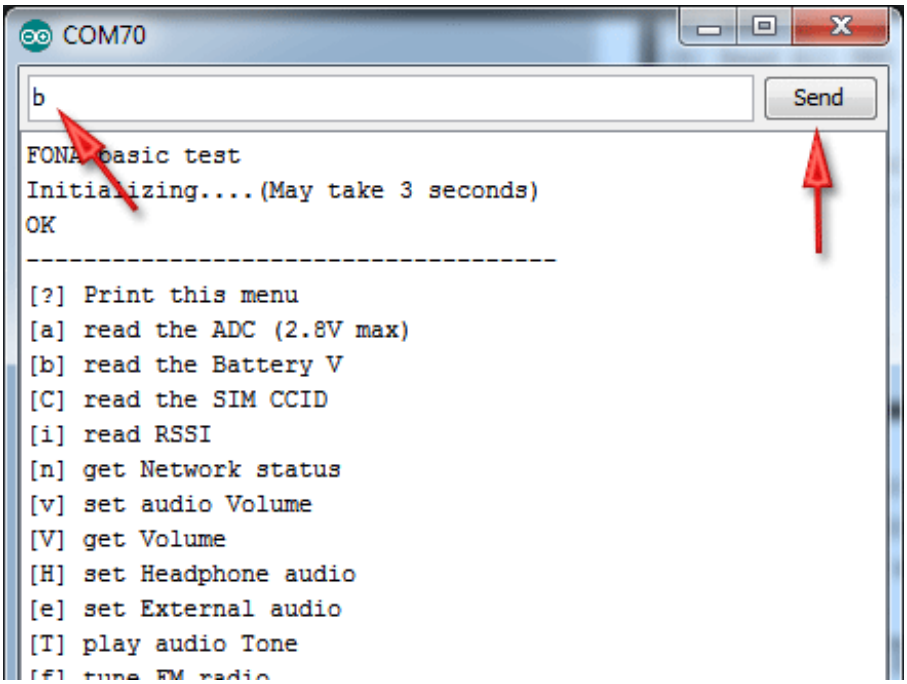
The test sketch has a menu interface so you can test out just about everything the FONA can do. The menu may change slightly as we add more functionality and update code!

Continue onto the next few sections to see what functionality you can test with the sketch

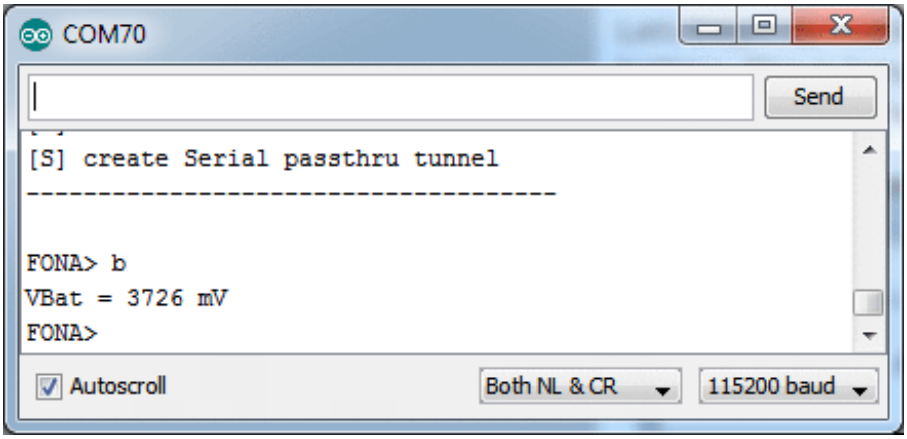
Hardware Test

Battery voltage

Lets begin by reading the battery voltage. That's the lipoly battery. This is handy if you need to track when the battery is low! type **b** into the command window and hit **Send**

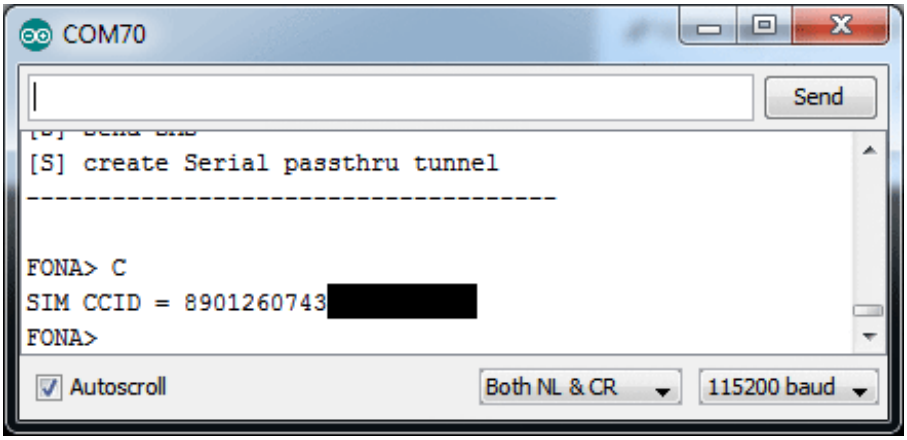


You'll see a print-out of the battery voltage in mV, so in this case its 3.726V



Check SIM CCID

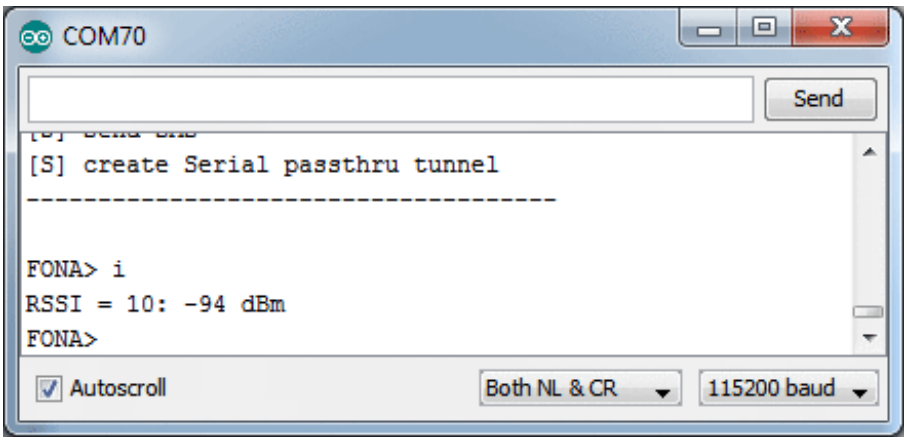
You can verify that the SIM is inserted and correct by reading the CCID, which is the unique identifier printed on it with **C**


COM70
Send
[S] create Serial passthru tunnel
FONA> C
SIM CCID = 8901260743
FONA>
Autoscroll
Both NL & CR
115200 baud

Network Test

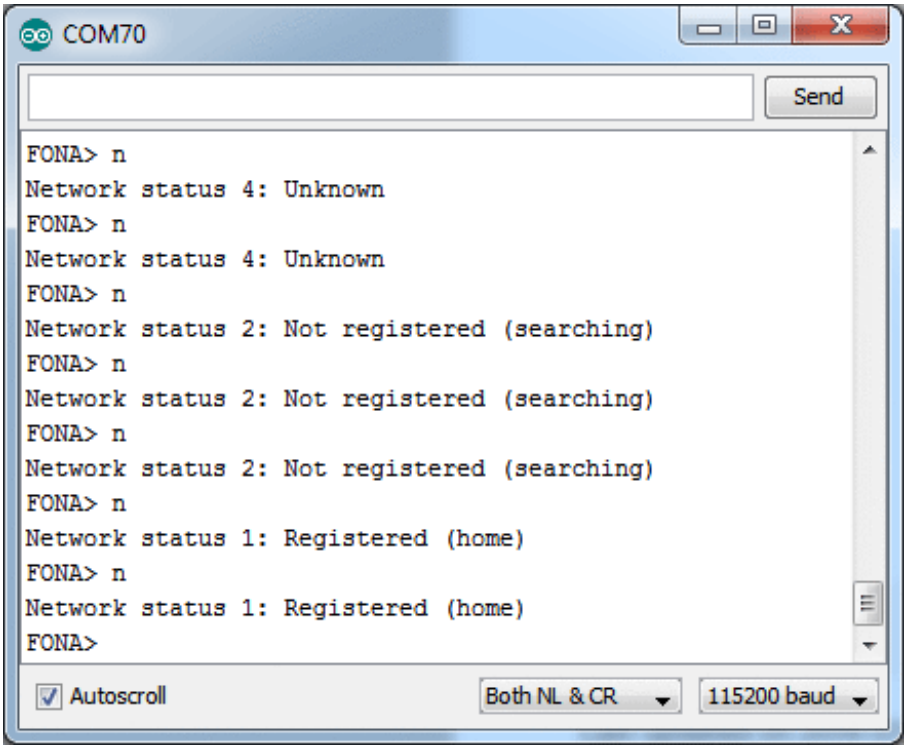
Check RSSI (network signal strength)

You can ask the FONA for the signal strength with the command **i**. The reply is a number, but you can convert it to dBm. Try to have the signal strength higher than 5 in order to make calls, SMSs, etc. In this case, I've got a 10



Checking Network Registration

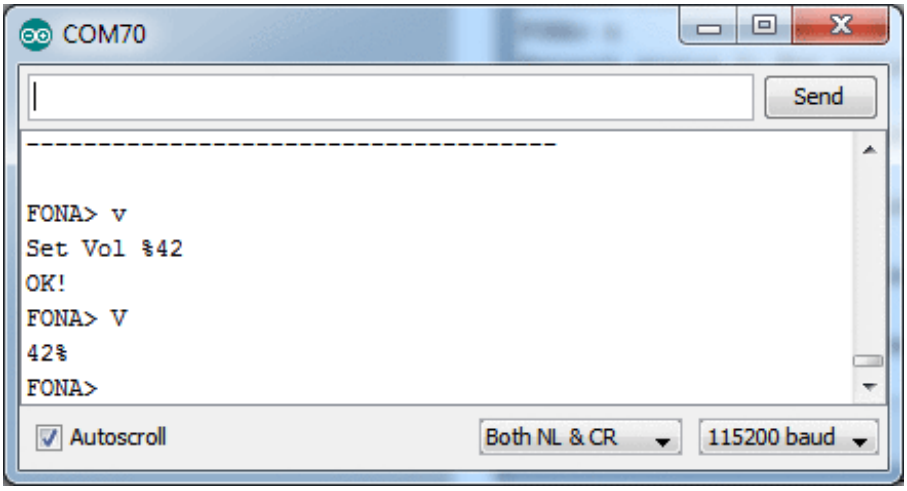
If the FONA has good signal it will immediately try to locate a cell tower and register to it.
You can check the status of the network with **n**
Once it's Home Registered, give it like 5-10 more seconds before trying to access/send SMS's or phone calls.


COM70
Send
FONA> n
Network status 4: Unknown
FONA> n
Network status 4: Unknown
FONA> n
Network status 2: Not registered (searching)
FONA> n
Network status 2: Not registered (searching)
FONA> n
Network status 2: Not registered (searching)
FONA> n
Network status 1: Registered (home)
FONA> n
Network status 1: Registered (home)
FONA>
Autoscroll
Both NL & CR
115200 baud

Audio Settings & Test

Set and Get audio volume

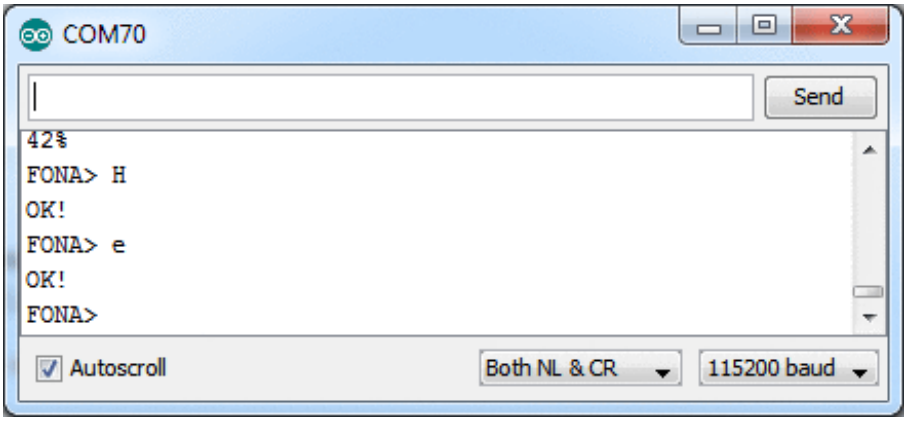
You can set the audio volume with **v** and retrieve it with **V** - its in % so ranges from 0 to 100



Setting Headset or External audio

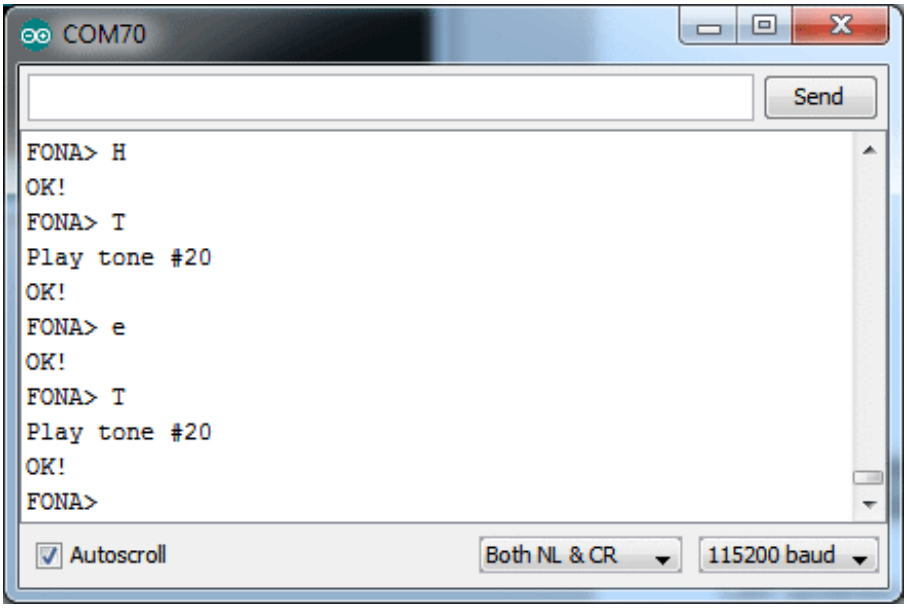
There are two audio paths on the FONA. One is the headset, thru the 3.5mm audio jack. The other is "external" - using the two speaker and mic pins for wiring up external speaker and mic. FM audio, phone calls, tones, etc can be routed to one or the other.
To set the audio to headset, use the command **H**
To set the audio to external, use the command **e**

> Note the FONA 808 only has Headset audio, so setting External audio wont do anything. The Feather FONA does not have headphone brought out, so use external only!



Playing Toolkit Tones

You can test the audio path with the toolkit tones. These are tones that mimic what some phone services sound like. For a full list of tones, you can check the **AT+STTONE** command in the AT command datasheet. We'll use tone #20 which is the American dial tone.
You can switch to headset mode, play a tone, then try it on the external audio mode. This is a very easy way to try out both speakers for debugging

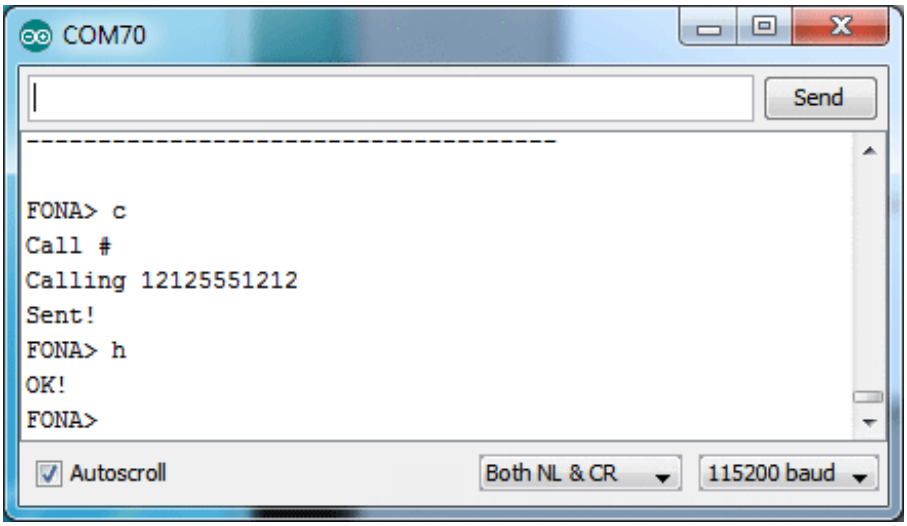


Phone Calls

Make Phone Calls

OK now we're onto the good stuff. You can make a phone call with FONA pretty easily. Make sure you have the right audio interface selected (external or headset!) before you go forward

Make a call with **c** - the call happens in the 'background'. When you're done then you can hang up with **h**

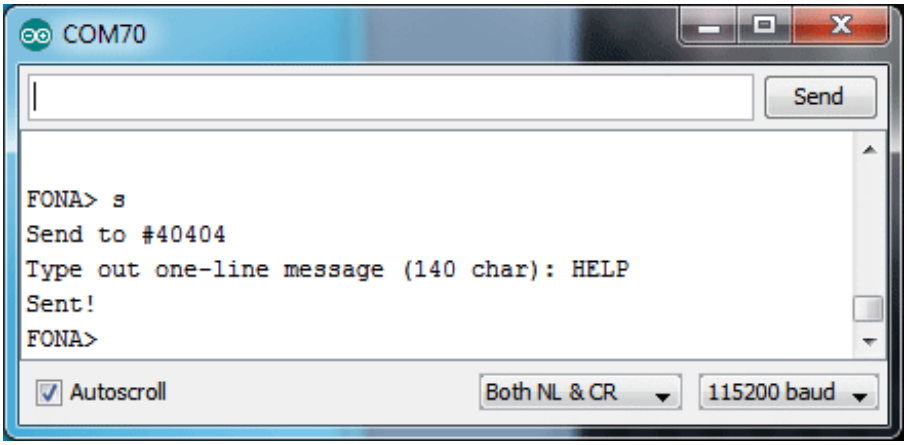


SMS

Send and Read SMS

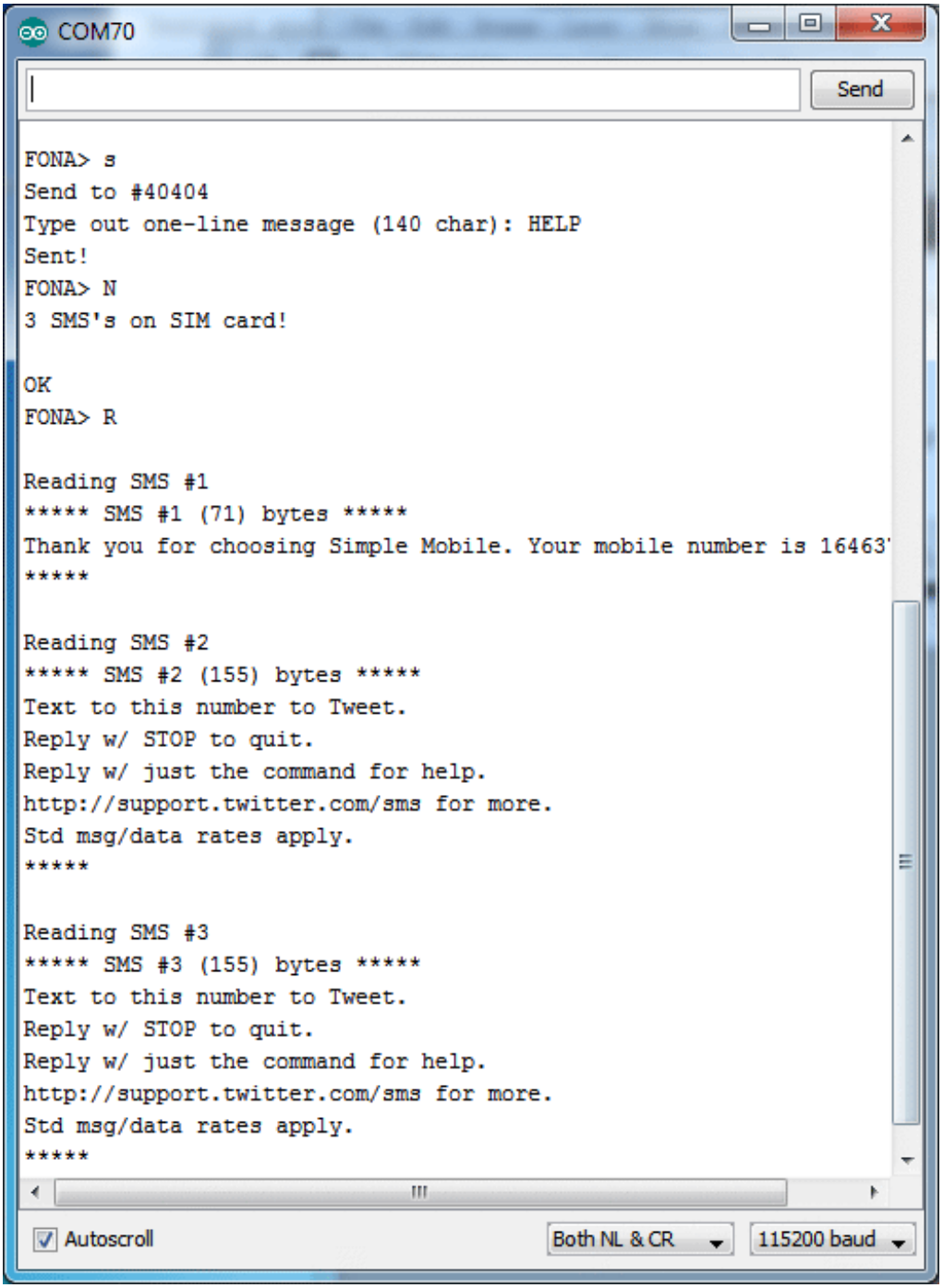
Another easy thing you can do is send and receive SMS messages. Lets start by sending an SMS. We'll use twitter's 40404 short code, which will auto respond, making it easy to verify both sending and receiving

You *can* send multi-line SMS's using the library API but for this example, its easier to parse the data if its a single line!

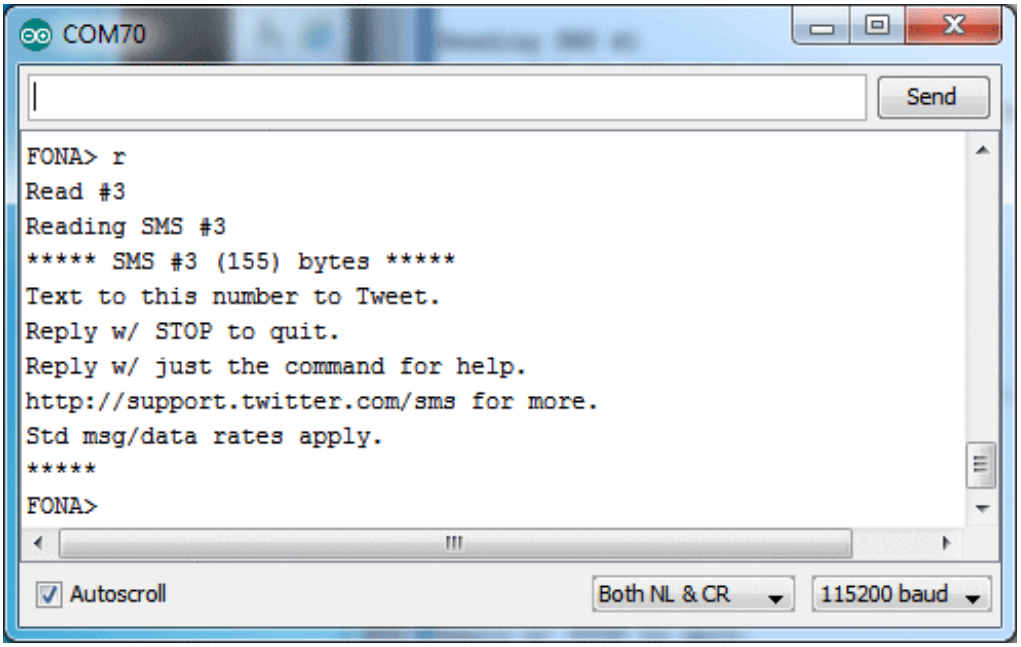


You can then ask the SIM how many SMS's it has with **N** and read all of them with **R**

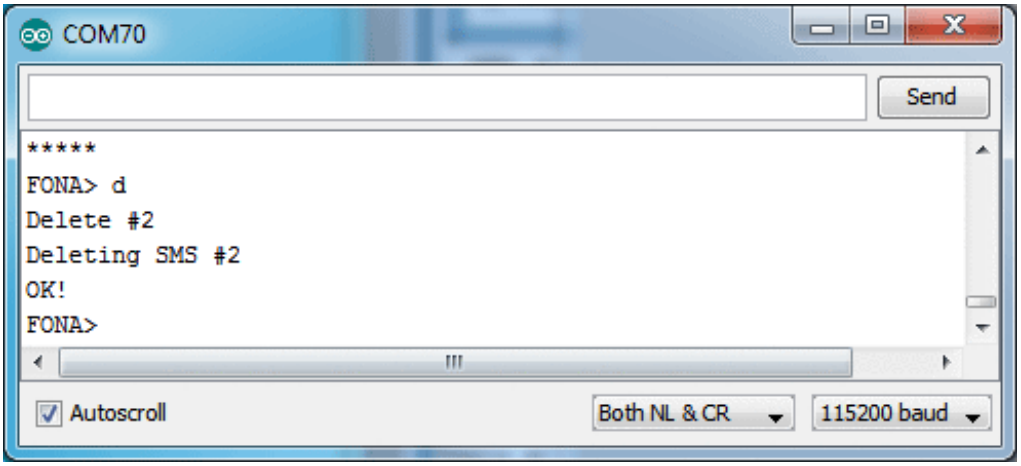
Note that SMS's are referred to by slots but the number does not include empty slots. We'll show this in detail in a bit



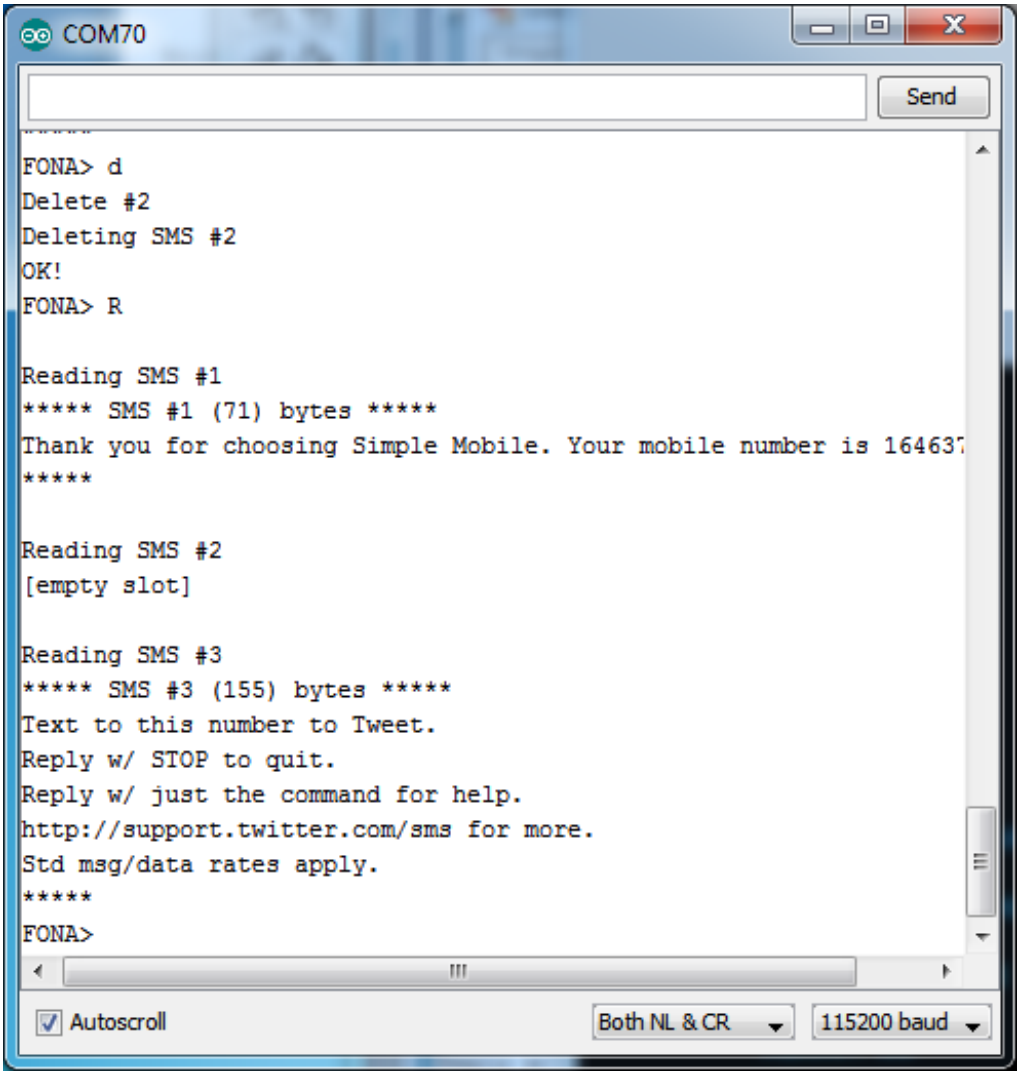
You can read individual SMS's with **r**



And delete SMS's by slot # with **d**



Note that before I deleted SMS #2, so if I read them again, that SMS # will be an empty slot. SMS number #3 doesn't "move slots"!


COM70
Send
FONA> d
Delete #2
Deleting SMS #2
OK!
FONA> R
Reading SMS #1
***** SMS #1 (71) bytes *****
Thank you for choosing Simple Mobile. Your mobile number is 164637
*****
Reading SMS #2
[empty slot]
Reading SMS #3
***** SMS #3 (155) bytes *****
Text to this number to Tweet.
Reply w/ STOP to quit.
Reply w/ just the command for help.
http://support.twitter.com/sms for more.
Std msg/data rates apply.
*****
FONA>
Autoscroll
Both NL & CR
115200 baud

FM Radio (FONA800)

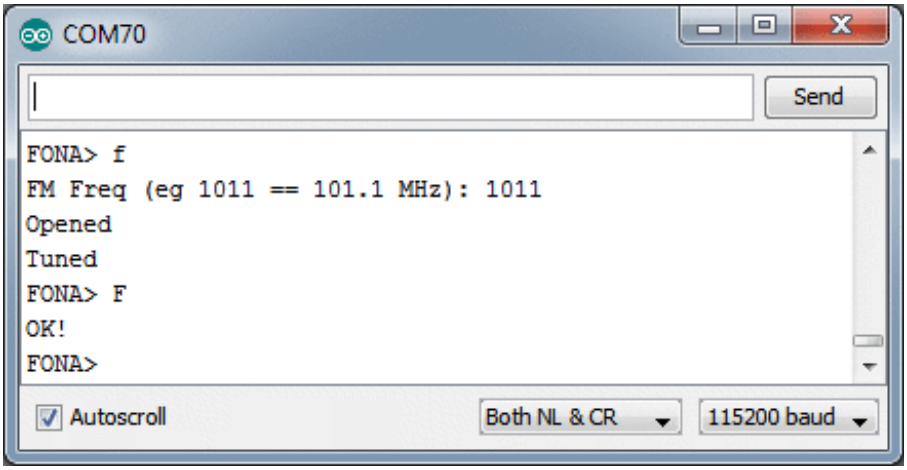
FM radio tuning/listening is only for FONA 800, the FONA 808 and FONA 3G does not contain a tuner

FM Radio (FONA 800 only)

The FONA has an FM receive in it. It uses the headset as the 'antenna' but it works pretty well considering! The FM radio goes thru whatever audio path you have set up

You can open and tune to an FM frequency in units of 100KHz. So if you want to tune to 88.1MHz, type in 881. For 102.3, type in 1023.

Use the **f** command to open and tune, and **F** to close it



Feather FONA FAQ

When I get an incoming call the Feather FONA 'resets' !
Wer're not sure why this is but it has something to do with the default audio being set to 'Headset'. You can fix this easily by calling

`fona.setAudio(FONA_EXTAUDIO)`

during init, or sending the fona the direct command `AT+CHFA=1`

You may also want to set the audio volume to 0 if you're not using audio output. We think it's the ringtone signal coupling in and resetting the SIM card?

My Feather FONA resets during use!
There's a few things that seem to be causing Feather FONAs to sponaneously reset.

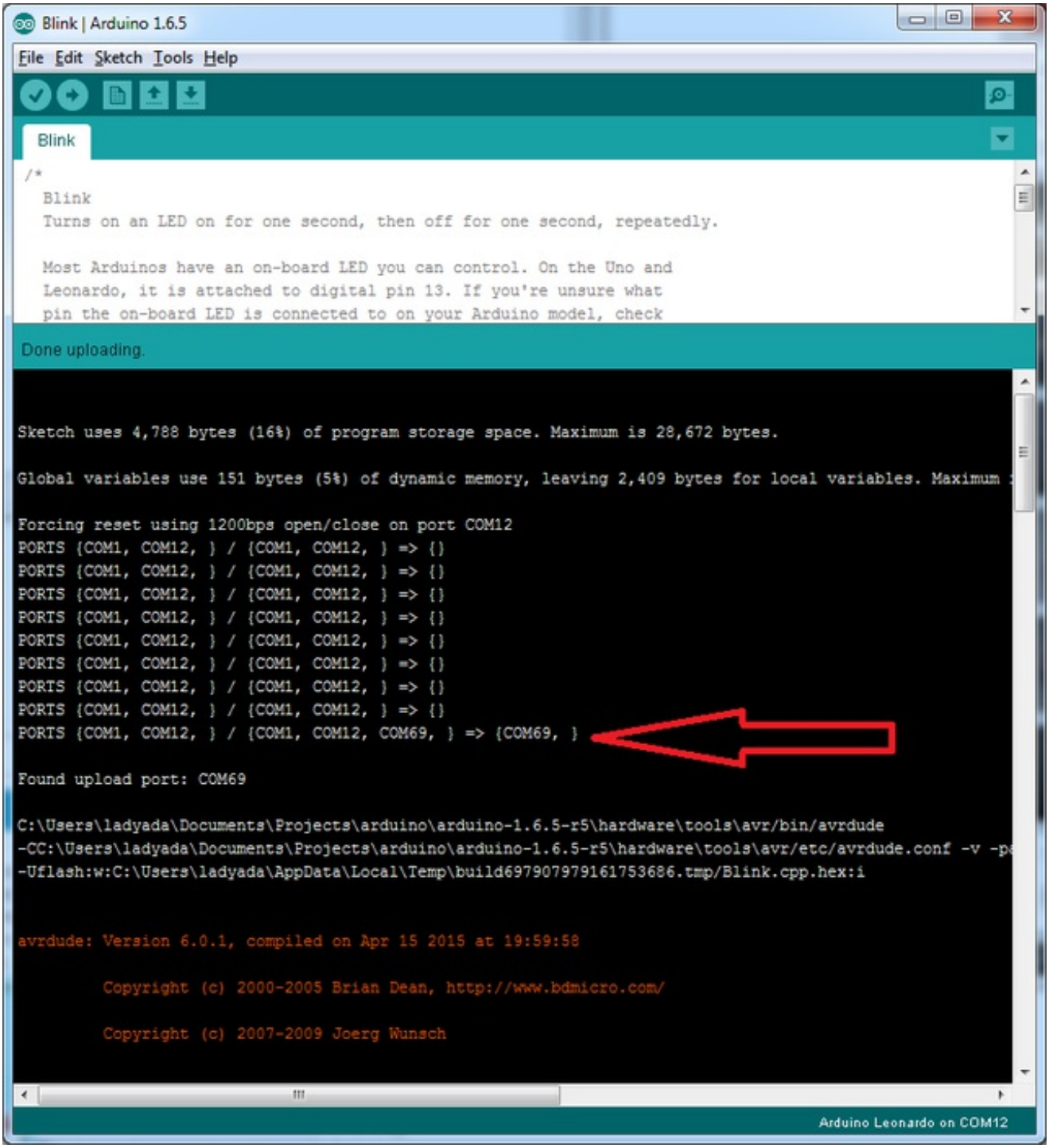
1. Battery charge is too low, and the FONA tries to talk on the network, which draws 1A from the battery, which drops the power supply voltage and the main chip resets. 500mAh is minimal, 1200mAh is better for battery sizing. Keep it charged, it acts as the main power source for the cell module
2. Antenna is too close to the Feather. The antenna is a *radiator of massive amounts of RF* keep its as far away as possible from the Feather itself. If you have an antenna attached put it so it sticks *out* and *away* from the Feather and definitely as far as possible from the processor chip!
3. The above incoming call reset issue, which is unclear why its happening - but easy to fix

HELP!

Ack! I "did something" and now when I plug in the Feather 32u4 it doesn't show up as a device anymore so I cant upload to it or fix it...
No problem! You can 'repair' a bad code upload easily. Note that this can happen if you set a watchdog timer or sleep mode that stops USB, or any sketch that 'crashes' your Feather

1. Turn on **verbose upload** in the Arduino IDE preferences
2. Plug in feather 32u4, it won't show up as a COM/serial port that's ok
3. Open up the Blink example (Examples->Basics->Blink)
4. Select the correct board in the Tools menu, e.g. Feather 32u4
5. Compile it (make sure that works
6. Click Upload to attempt to upload the code
7. The IDE will print out a bunch of COM Ports as it tries to upload. **During this time, press and release the reset button, you'll see the red pulsing LED that tells you its now in bootloading mode**
8. The Feather 32u4 will show up as the Bootloader COM/Serial port
9. The IDE should see the bootloader COM/Serial port and upload properly

For Feather M0, same as above, but *double click* the reset button, you'll see the red pulsing LED that tells you its bootloading

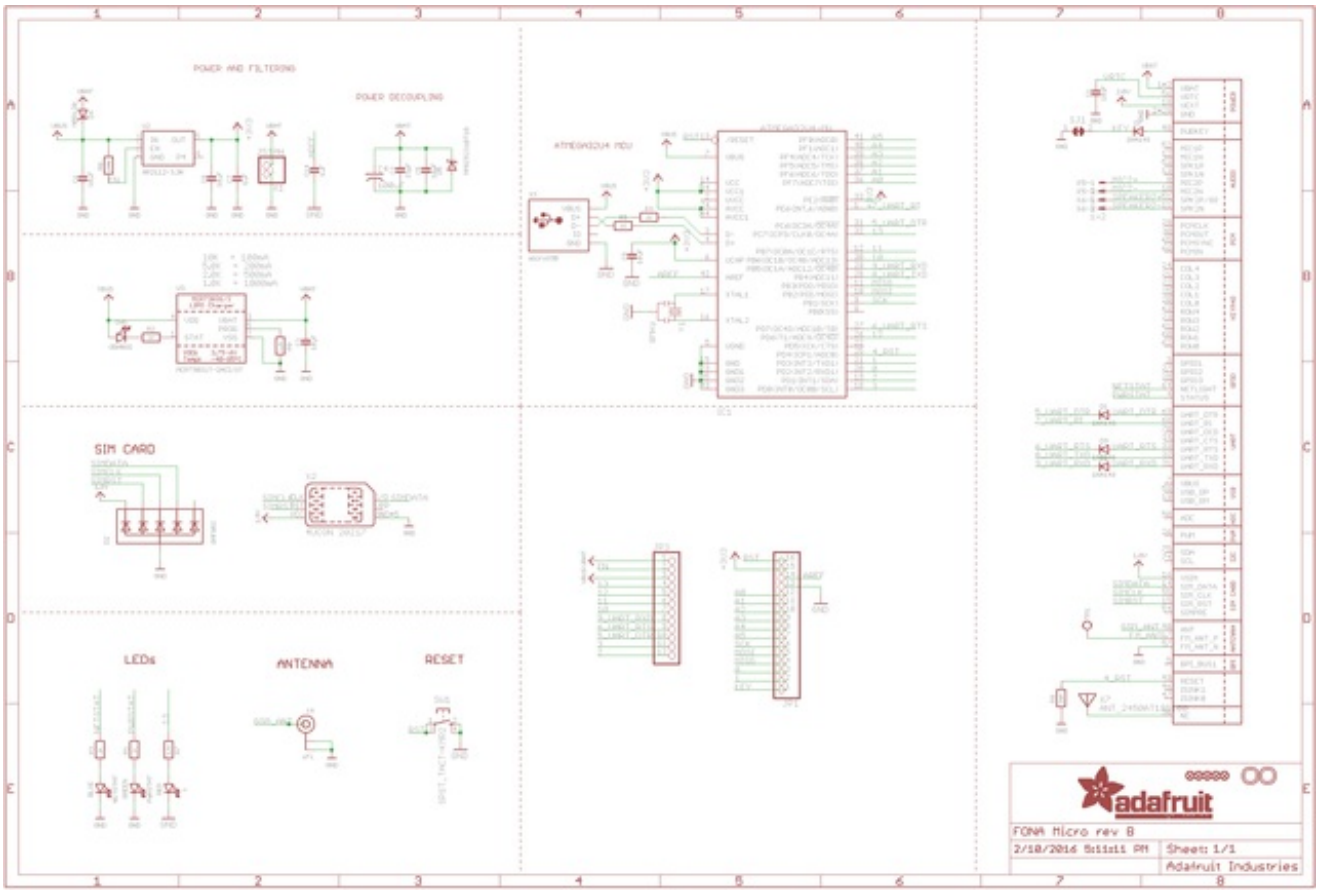

Blink | Arduino 1.6.5
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
Done uploading.
Sketch uses 4,788 bytes (16%) of program storage space. Maximum is 28,672 bytes.
Global variables use 151 bytes (5%) of dynamic memory, leaving 2,409 bytes for local variables. Maximum
Forcing reset using 1200bps open/close on port COM12
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, } => {}
PORTS {COM1, COM12, } / {COM1, COM12, COM69, } => {COM69, }
Found upload port: COM69
C:\Users\ladyada\Documents\Projects\arduino\arduino-1.6.5-r5\hardware\tools\avr/bin/avrdude
-CC:\Users\ladyada\Documents\Projects\arduino\arduino-1.6.5-r5\hardware\tools\avr/etc/avrdude.conf -v -p
-Uflash:w:C:\Users\ladyada\AppData\Local\Temp\build6979079791617536 86.tmp/Blink.cpp.hex:i
avrdude: Version 6.0.1, compiled on Apr 15 2015 at 19:59:58
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch
Arduino Leonardo on COM12

Downloads

- Adafruit FONA Arduino Library (http://adafru.it/ncj)
- Adafruit Feather 32u4 PCB files (http://adafru.it/nck)

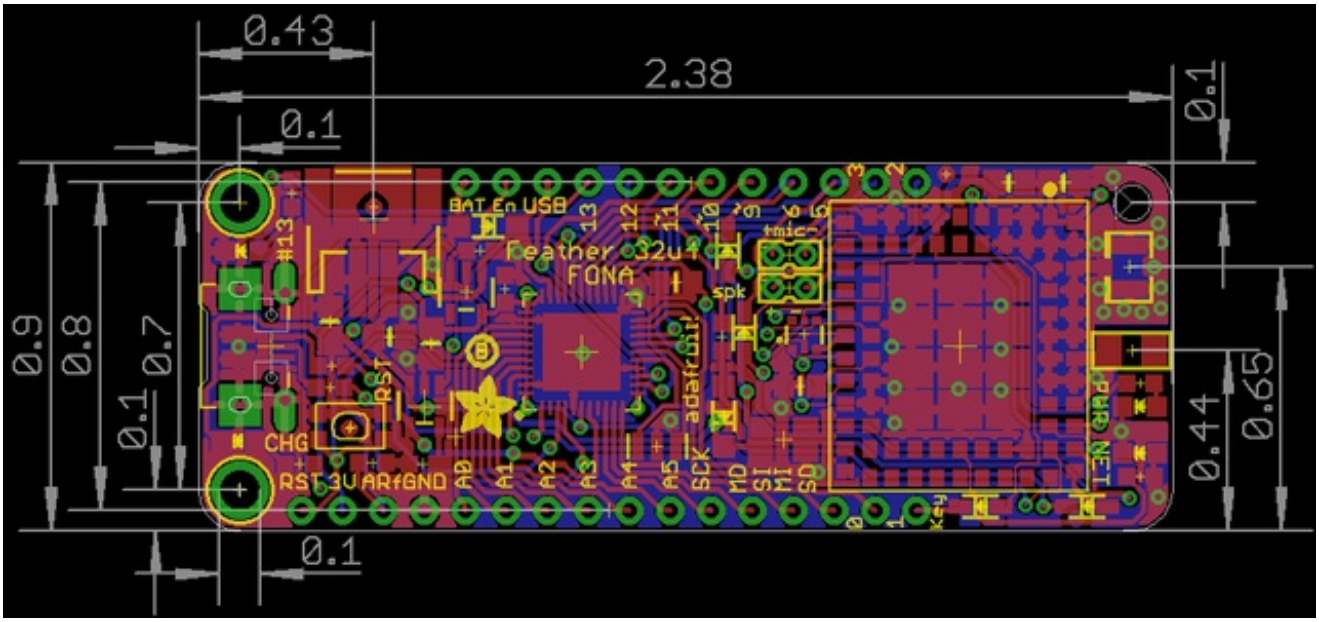
Schematic

Click to enlarge



Fabrication Print

Dimensions in Inches



Datasheets:

- SIM800 Command Manual (http://adafru.it/kDV) - All the basic commands that the module supports
- SIM800 Hardware Design (http://adafru.it/kDW)
- SIM800 Sleep App Note (http://adafru.it/kDX)
- SIM800 Embedded AT App Note (http://adafru.it/kDY)
- SIM800 Compiling Environments (http://adafru.it/kDZ)
- SIM800 Bluetooth App Note (http://adafru.it/kE0)
- SIM800 FM App Note (http://adafru.it/kE1)
- SIM800 FS App Note (http://adafru.it/kE2)
- SIM800 GNSS Location App Note (http://adafru.it/kE3)
- SIM800 GSM Location App Note (http://adafru.it/kE4)
- SIM800 IP App Note (http://adafru.it/kE5)
- SIM800 MMS App Note (http://adafru.it/kE6)
- SIM800 Multiplexer App Note (http://adafru.it/kE7)
- SIM800 NTP App Note (http://adafru.it/kE8)
- SIM800 PCM App Note (http://adafru.it/kE9)
- SIM800 Software Upgrade App Note (http://adafru.it/kEa)
- SIM800 SSL App Note (http://adafru.it/kEb)
- SIM800 STK App Note (http://adafru.it/kEc)
- SIM800 TCPIP App Note (http://adafru.it/kEd)
- SIM800HL Schematic and PCB Reference Design (http://adafru.it/kEe)
- SIM800H GCF I13GC9551_RSE-E Report (http://adafru.it/kEf)
- SIM800H CE Certificate (http://adafru.it/kEg)
- SIM800H CTTL + GCF Test Report (http://adafru.it/kEh)
- I14Z46950-GPM01 Test Report (http://adafru.it/kEi)

- SIM800H PTCRB Test Report (http://adafru.it/kEj)
- SIM800H RoHS Test Report (http://adafru.it/kEk)
- SIM800H REACH Test Report (http://adafru.it/kEl)
- SIM800H GCF I13GC9551 Test Report (http://adafru.it/kEm)
- SIM800H FCC PCB Grant Final (http://adafru.it/kEn)
- SIM800H FCC DSS Grant Final (http://adafru.it/kEo)
- SIM800H CE EMC Test Report (http://adafru.it/kEp)
- SIM800H CE RF-BT Test Report (http://adafru.it/kEq)
- SIM800H CE RF-GSM Test Report (http://adafru.it/kEr)
- SIM800H SAFETY Test Report (http://adafru.it/kEs)
- SIM800H EPL Certificate (http://adafru.it/kEt)
- FCC TCB BT (http://adafru.it/jSb)
- R&TTE Statement of Opinion (http://adafru.it/jSc)
- FCC Part 15B Test Report (http://adafru.it/jSd)
- FCC Part 15C Test Report (http://adafru.it/jSe)
- FCC RF Test Report (http://adafru.it/jSf)
- FCC TCB (http://adafru.it/jSA)