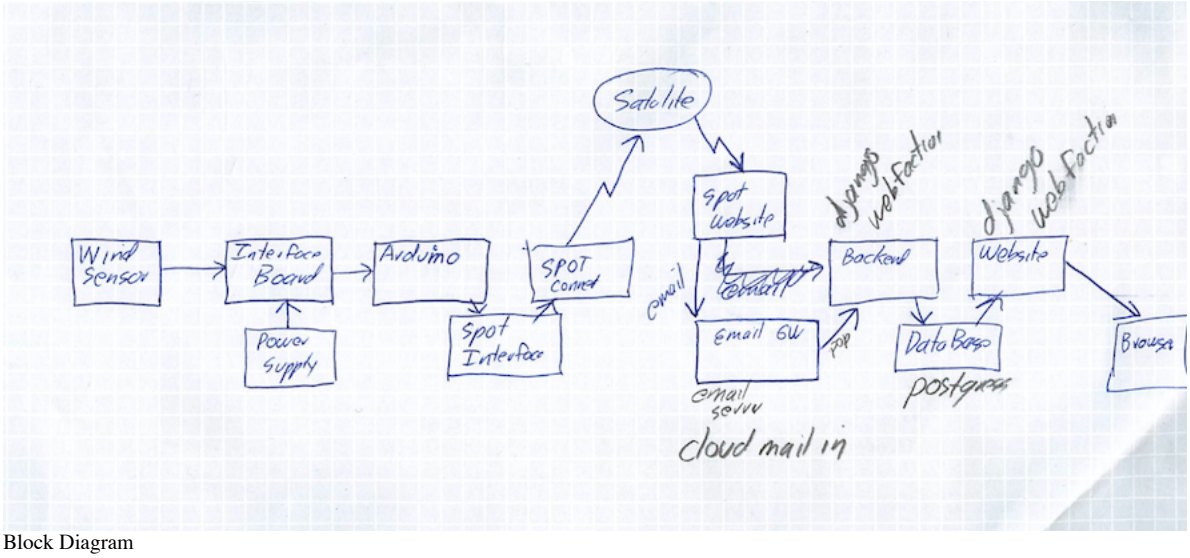
Overview:

The system uses an a small arduino computer to sense the wind and temperature and then connects to a SPOT Connect to send it up to a satellite. This results in an email sent to cloudmailin which then does an HTTP POST to the website with the email data. The website is a Django app. The basic block diagram of the modules is shown below.



Block Diagram

Wind Sensor

The anemometer has a reed switch that connects the circuit once per revolution. Any advice on good sensors that work in winter conditions with ice and snow and can be left alone for long periods of time? And cheap too :-)

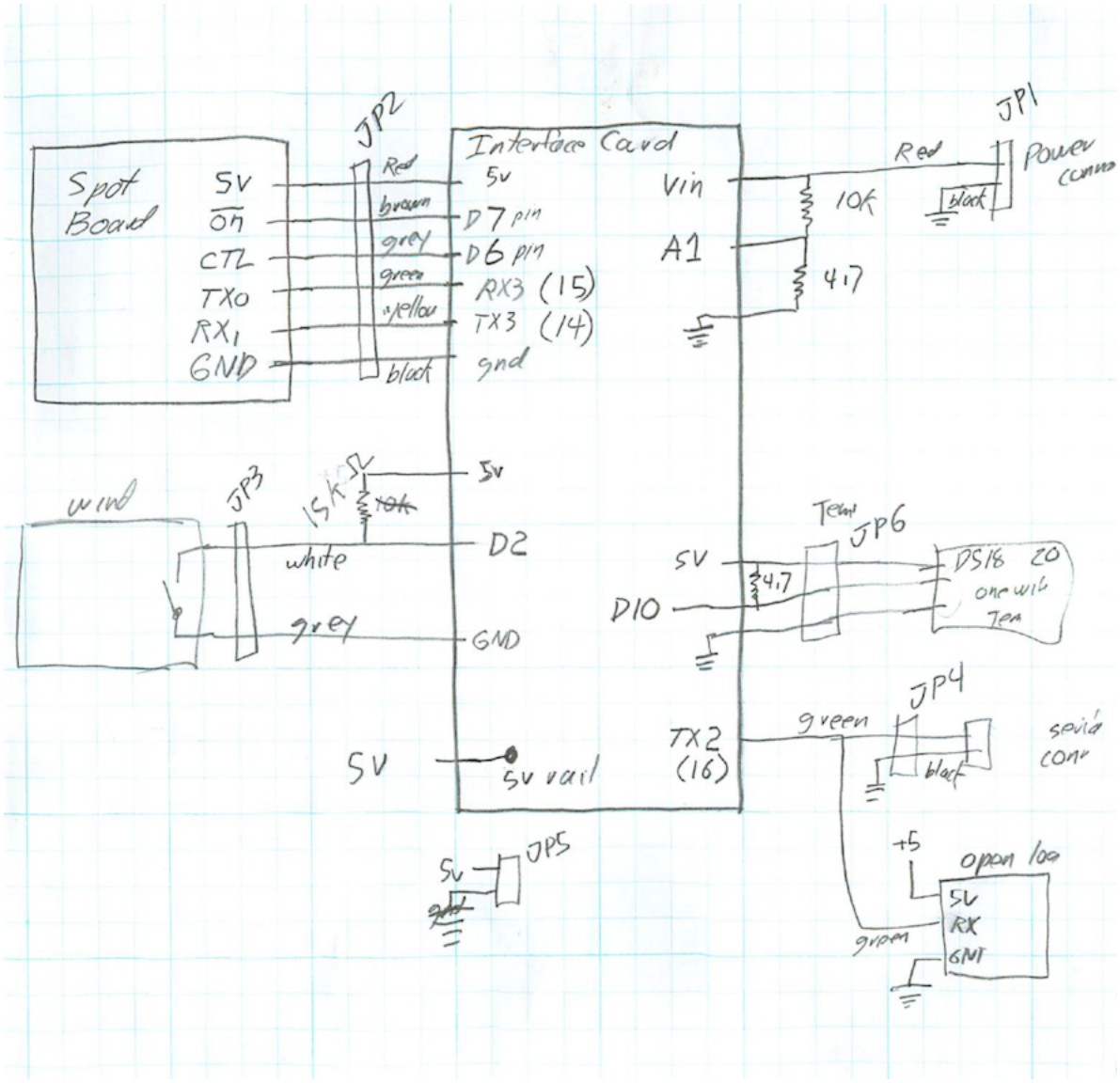
Temperature Sensor

Right now the temperatures are not even close to correct and report way too high. I have tired both the DS18B20 and DS18S20 - I don't have a heat sink attached but they seem to report too high.

Interface Board

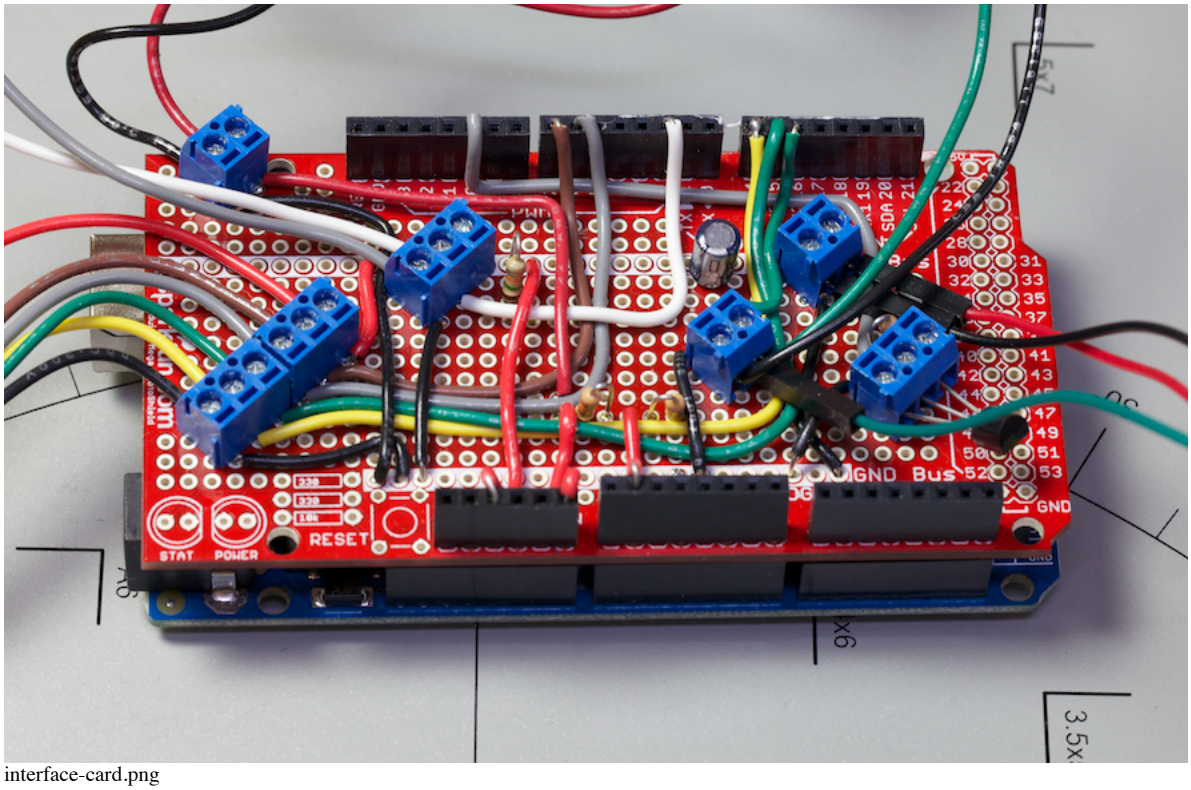
The interface board connects the rest of the stuff up to the Arduino. It provide a 4.7k resistor for the 1-wire bus to the temperature sensor. It has a 10k pull up to provide power to the anemometer read switch. It connects serial port 3 to SPOT board and provides serial port to to the log system. It splits the battery voltage roughly in third and monitors that.

Schematic is:



schematic.png

Photo is:



interface-card.png

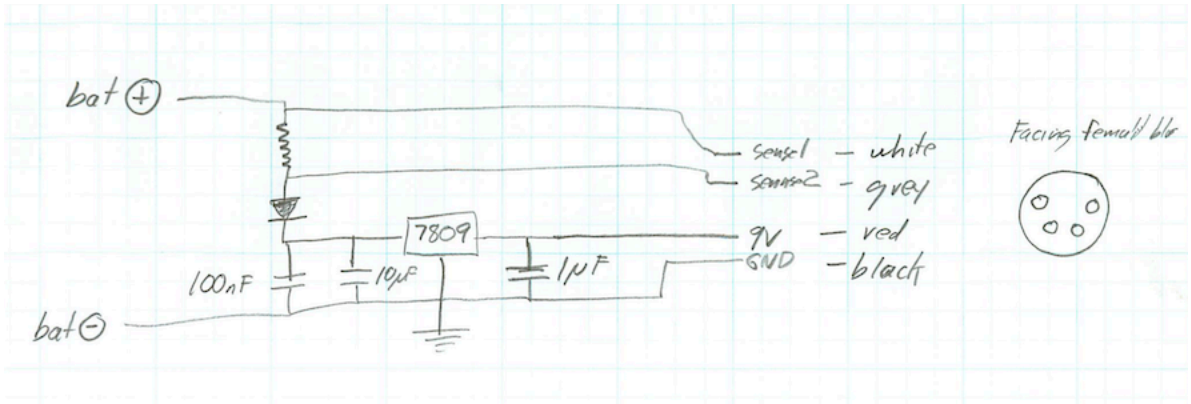
Power Supply

Currently using a "car battery". Actually it is battery that supports "deep cycles" designed more for boats and RVs than for starting cars.

Need to look into how much power is used at varies times. It looks like roughly speaking, when SPOT is not in use, the rest of the system consumes around 100 mA. This is very high and could be substantially reduced - starting by removing all the LEDs. The SPOT looks like it consumes roughly another 100 mA on average for 20 minutes each time a messages is sent. It is higher when the GPS is trying to locate the position. It goes very high for a brief time when the messages is being transmitted and looks like the consumes around 800 mA for 2.7 seconds. All of these measured at a roughly 12.5 volt supply.

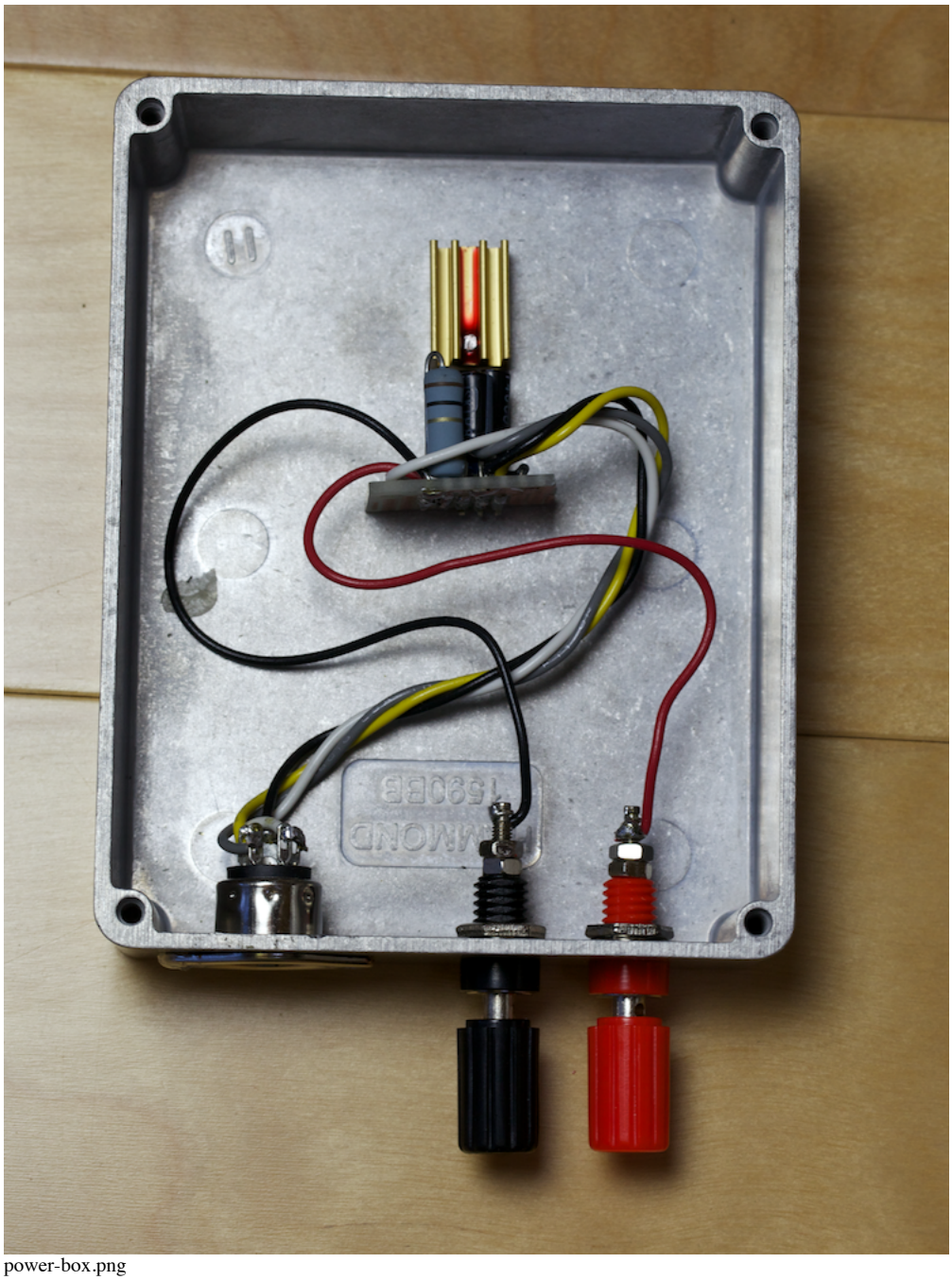
The voltage divider circuit does not have any over power protection (note, add zener diode on next design) so the input voltage must not go over 15 volts.

The schematic for the power supply is:



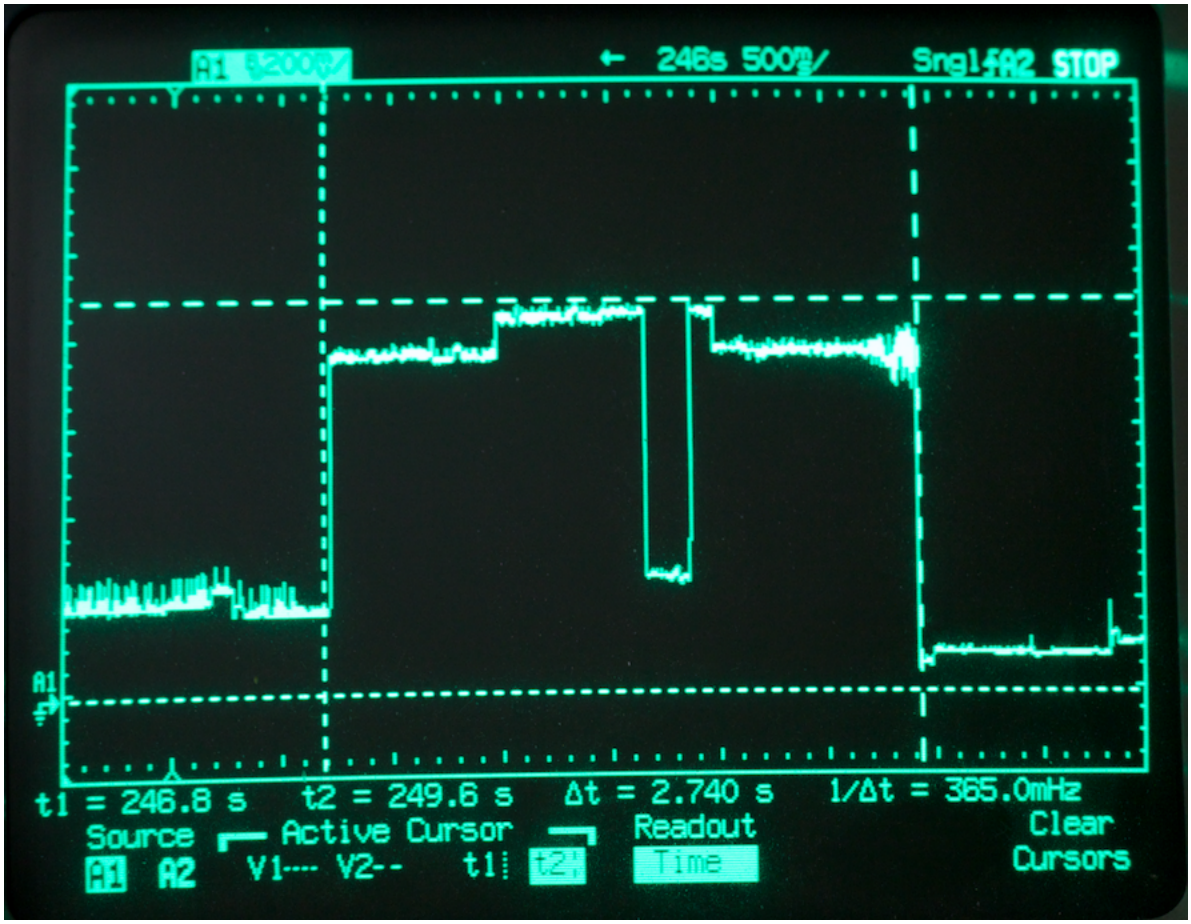
power-schematic.png

An image of the final box is:



power-box.png

The following shows the power usage spike when the SPOT device transmits a message:



power-usage.png

Arduino

The system is using an Arduino Mega 2560 because that was what was lying around and it has multiple serial ports but probably a normal Arduino would work. If moving to a normal Arduino, I would put the debug information on a soft serial line and use the main UART to talk to the SPOT.

You need to install the OneWire library from http://www.pjrc.com/teensy/td_libs_OneWire.html

The reed switch on the anemometer generates an interrupt which increments a counter. Approximately every 10 seconds, this counter is read to compute the current wind speed which is then used to update the min and max wind. The counter is also used to compute an average wind over a one hour period.

The battery voltage and current temperature are computed ever 5 minutes. [Note the current system is creating temperature measurements that seem way off].

Every hour, the system gets the current values, formats them into a JSON message and sends it using the SPOT. If the SPOT does not manage to send the messages within 30 minutes, it gets shutdown as this is probably some sort of weird error state on the SPOT.

The JSON messages is a single JSON object that contains an array named "v" that has 6 floating point values. The first is the current wind speed in m/s, the second value is the minimum wind speed over last hour, the next the average, then the maximum. The next value is the temperature in degrees C (totally inaccurate at this point) followed by the battery voltage in volts. An example JSON object is

{"v":[0.0, 0.0, 0.0, 0.0,-10.4,12.38]}

Data Log

This is optional but convenient for debugging. All the debug information printed from the Arduino is recorded on a flash card in the logger.

Currently using an OpenLog https://www.sparkfun.com/products/9530 to record all the debug info. It produces about 55 kB/hour in the data file (0.5 GB / year).

Right now the serial is configured for 19.2 kbps so the the CONFIG.TXT files for the open log looks like: 19200,26,3,0

Spot Interface

I'm using the SatUplink Shield from sparkfun [https://www.sparkfun.com/products/11088]. This provides the various voltages the spot requires along with a way to turn the power to the SPOT on and off.

Spot Connect

Before you take apart a spot, make sure it works with your mobile phone, you have the website and account set up and working, and have upgraded to latest firmware on the spot. It's till pretty unclear how it signals which contact list to use on the website so I have just one group configured and it has a few emails addresses that get notified when a new spot messages is received. More complicated things probably work but that is what I have.

One of the key problems with the SPOT system is that there is no acknowledgment that the messages SPOT sent was received. This makes it much harder to use and make it unclear how many times the messages should be retransmitted. It would be nice to reduce the number of retransmissions as this consumes lots of power.

Most the information about the data to and from the SPOT device came from looking at the documentation from globalstar on the STX2 modem (which was used in the earlier SPOT devices) and just looking at the serial data that the SPOT device sent to Arduino.

Spot Website and Email Gateway

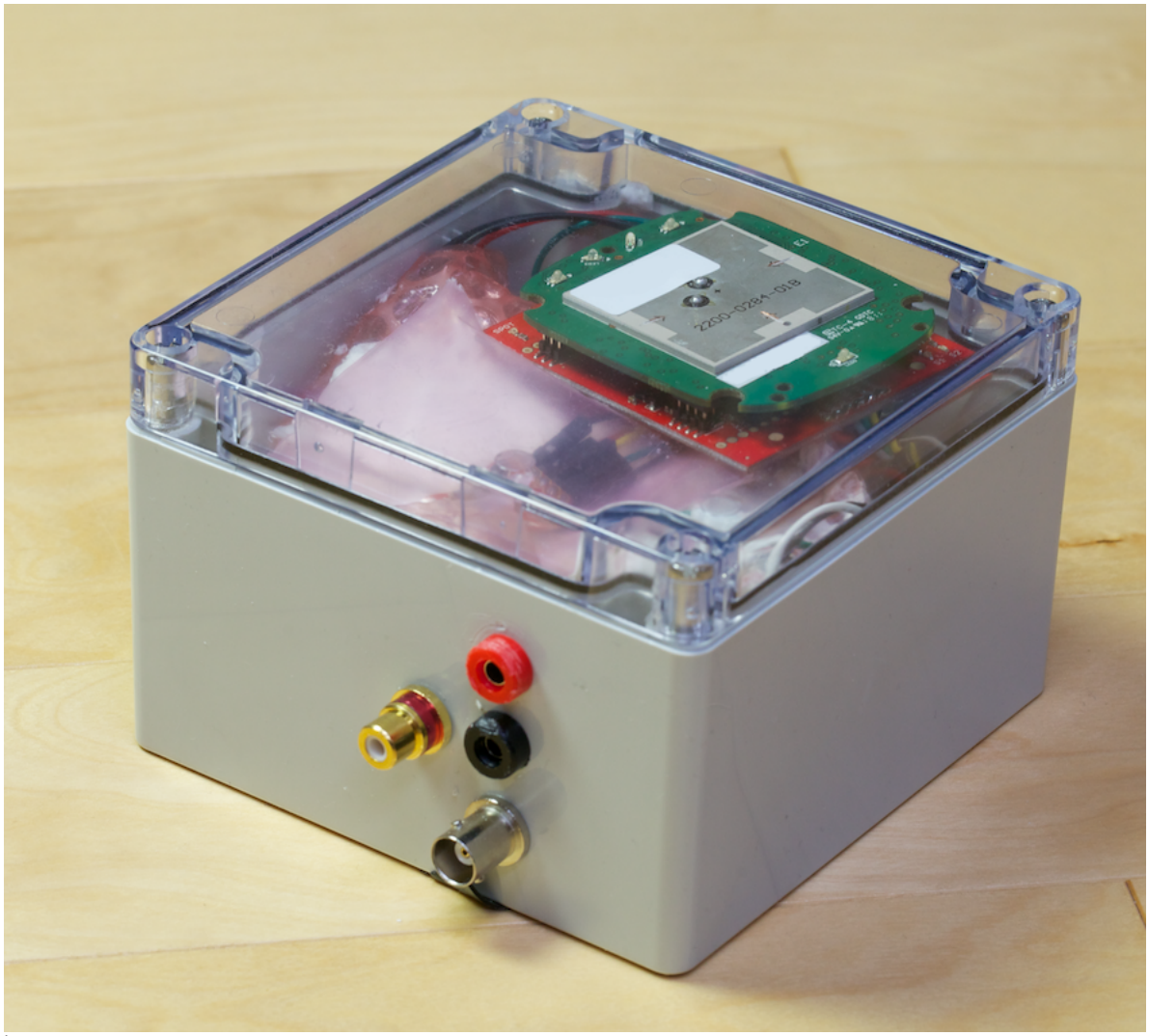
The findmespot website is configured to send messages from my SPOT to the email gateway at http://www.cloudmailin.com/ which is configured to use the JSON format to post to the Django website. [Note: some trivial python code could just access an email account directly and skip cloudmailin out of the picture. Plan to do that Real Soon Now.]

Website

The website is written in python using Django. I am using webfaction to host it as I find the cost of doing it there way cheaper than GAE, Heroku, Rackspace, or AWS. The website is using postgres for the database but pretty much any database would work.

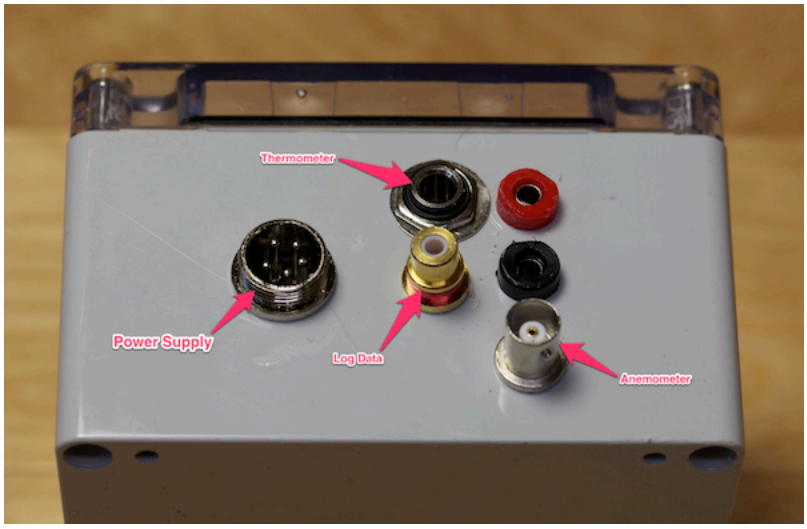
Connection on the box

Oh yah, I suck at putting electronics in boxes but the final things looks like:



box

The connectors look like:



connectors

The circular connector provides power and current / voltage sense lines. The BNC cable connects to the anemometer (outside is ground). The RCA jack has the debug serial data - outside is ground and inside is 19.2 kbps 8N1 serial data at 5V TTL voltage levels. The 1/4 inch stereo plug provides the 1-wire connection for the thermometer. Sleeve is ground, tip is +5 V, and ring is the 1-wire bus.

Cost of system

Very approximate costs in US$ as of Dec 2012

Equipment:

1. SPOT Device: 150
2. SatUplink Shield: 21
3. Shipping: 25
4. Arduino Mega 2560: 60
5. Proto Shield: 20
6. Temperature Sensor: 5
7. Misc parts: 25
8. Battery: 100
9. OpenLog: 26
10. FlashCard: 5
11. Power Supply: 10
12. Things I forgot: ??

Total: about $500

Annual Fees:

1. Spot Account: 100
2. Spot Messages (1/hour): 876
3. Domain Name: 15
4. DNS: 30
5. Website: 120
6. CloudMailIn: 108

Total: 1249 ($3.42 / day - less than Starbucks)

The next version:

Things to consider for the "next" version:

- Design for low power. Don't use arduino power supplies. Include a real time clock with alarm function to wake up the arduino so it can be put in sleep mode. Do all 3 volt design. Make sure sleep mode works. Remove all the LEDs from spot and arduino.

- Think about input power protection battery sense circuit.
- Think about local status reporting.
- Real time clock to sync reporting on the hour and reduce reporting at night.
- Figure out solar power.
- Use magnets for mounting boards to box ?
- Add local display of wind / voltage ?
- Use soft serial out for debug information
- Consider adding logging to SD card
- All cables are things you can buy - no soldering DIN connectors. Way to update firmware without removing board from box. Perhaps have all the connectors soldered main board instead of jumper wired to them.

On the wires going to wind and external temp, design some input protrection esd etc. Perhaps 1k resitors then then insisde that a pair of caliming shockey diodes to gnd and vcc - does arduiinon already do this ?