THE OPEN STUDIO

A BLOG BY FIFTYTHREE

Measuring Battery Voltage Is Simple, Or Is It?

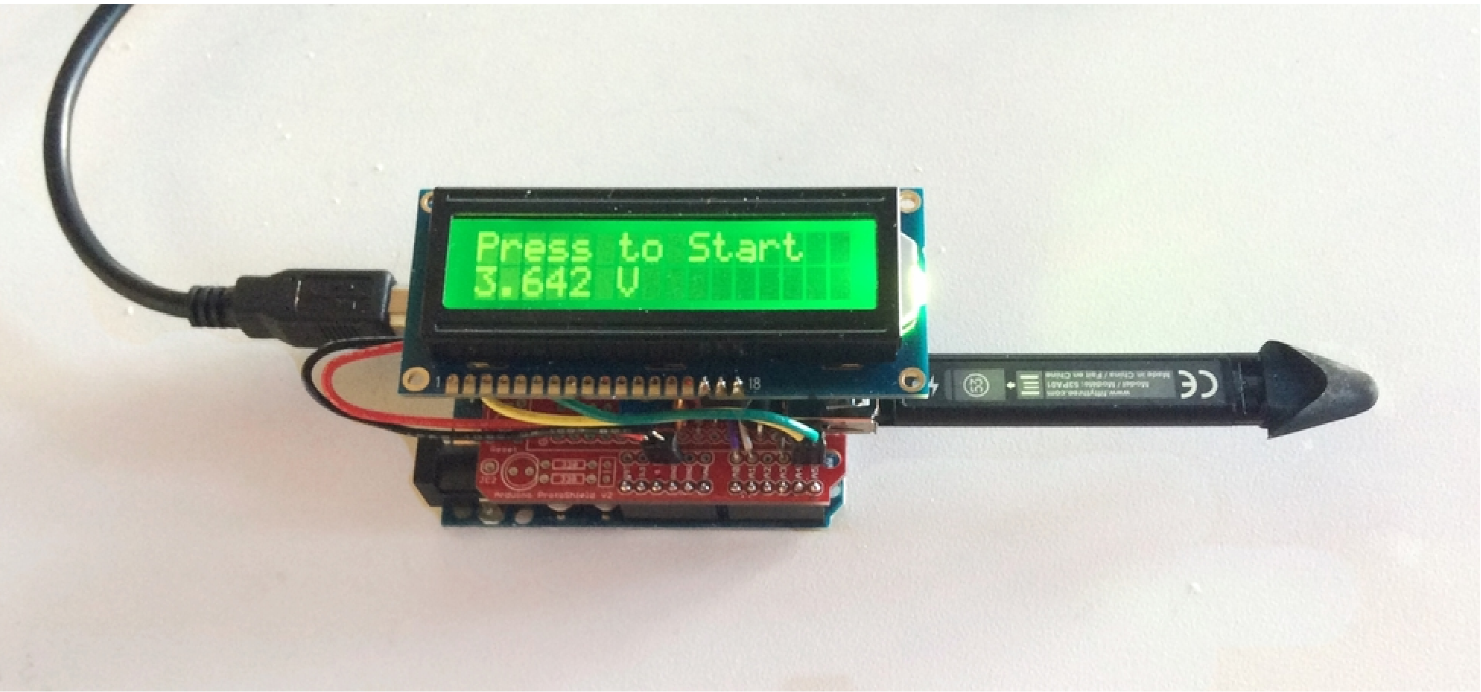
AUGUST 17, 2015

Inside Pencil by FiftyThree stylus

An important decision when designing a Bluetooth® Low Energy, aka "Bluetooth® Smart," device like our stylus Pencil is selecting a battery; too small and the device will need constant recharging, too large and the device becomes too big, heavy, or expensive.

Measuring battery characteristics provides critical data to help select the right battery for a device, and one of the characteristics we measure is how a battery's voltage changes as it drains. This article details this specific measurement and the lessons we learned in automating the collection of voltage data over time.

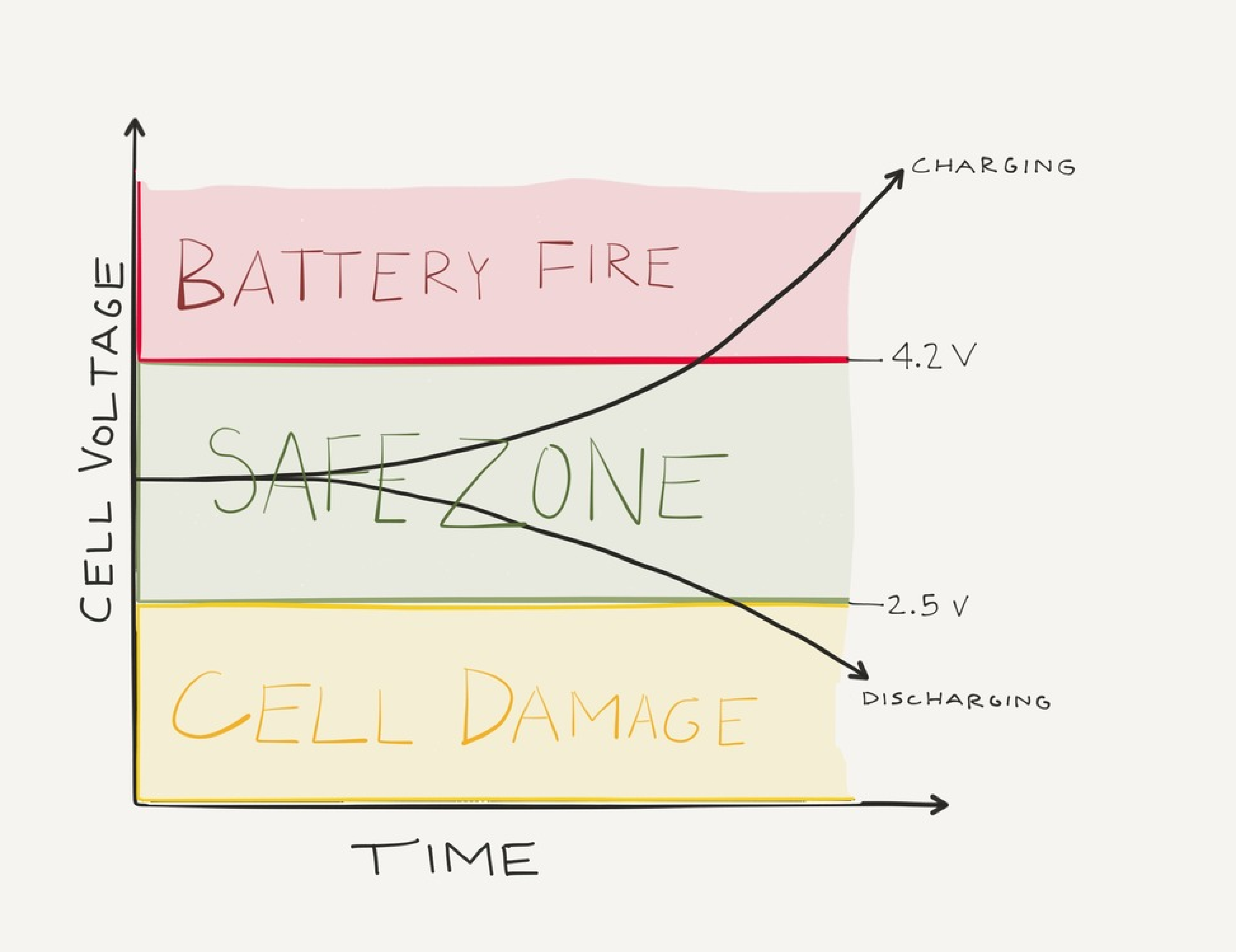
Why automate? Why not simply take a multimeter and stick the probes on the battery? Multimeters are great, but they're expensive and require someone to manually capture their readings. Collecting large amounts of data from multiple batteries would become an onerous task, which means we needed some basic automation to get enough data to make informed decisions. Since we regularly use Arduino open-source hardware in our prototyping process, we decided to use them to automate the entire battery test process. This seemed straightforward enough, but we soon learned that measuring voltage accurately was more subtle then we first thought.



LED display recording the battery voltage for Pencil

What We Learned

Lithium-polymer (LiPo) batteries are well suited for Bluetooth Low Energy (BLE) applications. Their low discharge rate is perfect for devices that don't draw considerable current, and the cell's compact footprint makes them easy to squeeze into tiny devices. The challenge with LiPo batteries lies in their charge and discharge profile because unlike nickel or lead-based batteries, LiPo cell voltage is not self-limiting. Without a specifically designed charger, the battery voltage would continually increase until it bursts into flames (a generally frowned upon outcome in electrical design). Discharging LiPo batteries without proper protection is only marginally better, and will result in cell damage without restricting the operating voltage to a very specific range.



Most small LiPo batteries, like the ones used in Pencil, have purpose-built circuitry to prevent this damage, cutting off battery voltages below a certain threshold. Although convenient in practice, this provides a challenge for testing because the cutoff voltage is surprisingly inconsistent. To further complicate matters, the discharge profile of batteries and the resulting product performance heavily depend on the types of loads applied.

Clearly, there was something to be learned about the voltage/time/load relationship here. To help steer design decisions, we developed a few core criteria that our battery measuring rig must comply with:

- Record the time/voltage relationship (real time clock involved)
- Measure voltage accurately (+/- 0.25%)
- Measure the battery current (+/- 1%)
- Vary the load applied to the battery
- Be compatible with any battery

- Record the external temperature (batteries are thermally dependent)

Although the goal of accurately measuring battery voltage is very specific in scope, we were guided by the overarching theme of automating data acquisition. With this in mind, we established two more criteria:

- Allow automation of battery tests
- Allow scaling of the test automation

Analog to Digital Conversion

A fluctuating battery voltage is a quintessential example of an analog signal, and clearly a sign that we needed to convert to a digital one. The ATmega chips found on most Arduinos come with built in 10-bit analog to digital conversion, so using this converter seemed like a good place to begin.

Calling analogRead() enables the ADC, which converts the input voltage on a certain pin to a number between 0 and 1023. This number is directly proportional to the reference voltage used by the Arduino. For example, a 3.3 volt input with a 5V reference would yield an output of 675 (1023/5*3.3). The Arduino comes with a convenient 5V internal reference, which seemed suitable for the 2.5-4.2 volt operating range of the batteries we were testing. We optimistically uploaded the simplest possible sketch:

```
//Positive (+) to pin A0
//Negative (-) to GND

void setup() {
    Serial.begin(9600);
    analogReference(INTERNAL);
}

void loop() {
    Serial.println(analogRead(A0)/1023.0*5.0);
}
//Case closed... or not
```

Reference and Gain Bias

As we quickly realized, there were two main issues with this approach. The first was noise, indicated by rapid

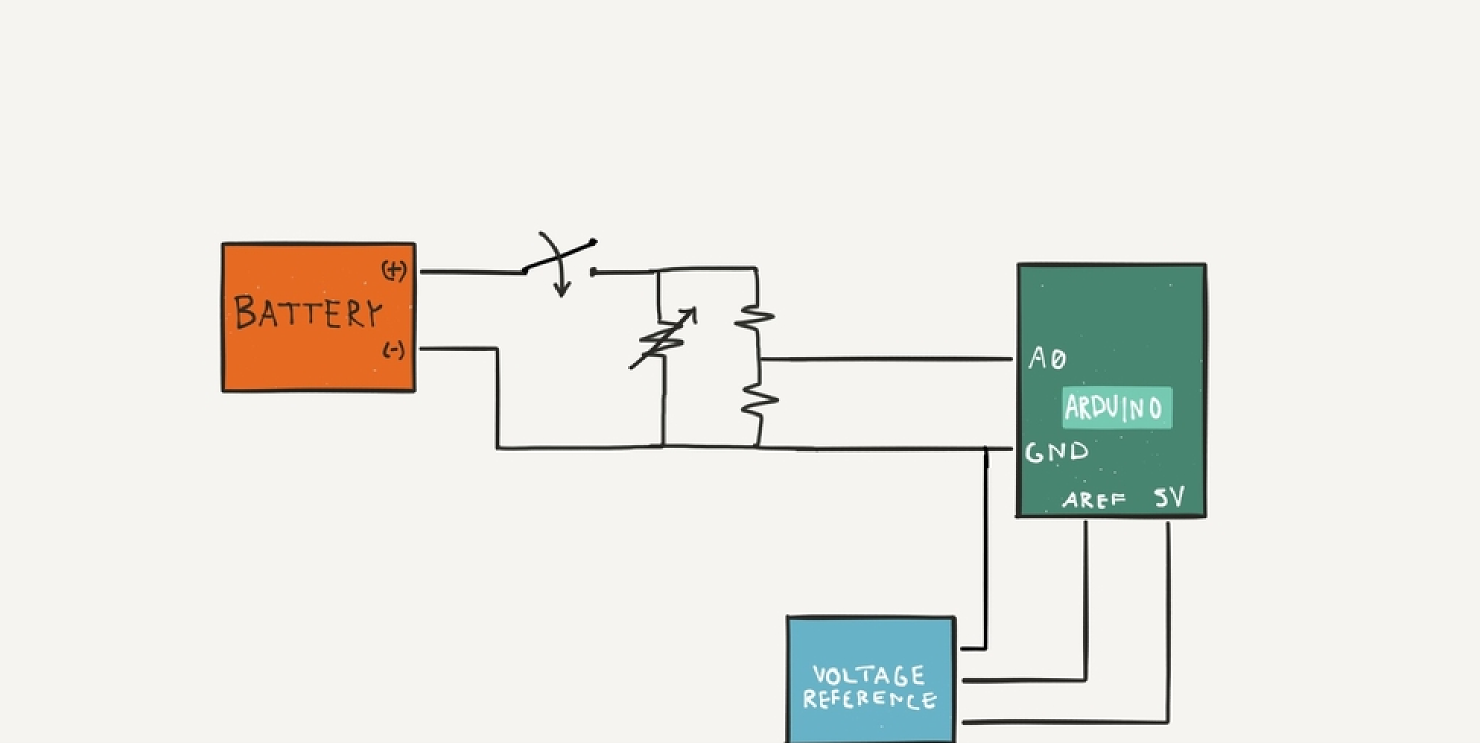
As we quickly realized, there were two main issues with this approach. The first was noise, indicated by rapid, seemingly random fluctuations in values. The second problem was one of offset. There was a consistent difference of approximately 100mV between the Arduino and more accurate voltmeter readings. We initially attributed this to gain bias, a phenomenon which plagues analog to digital converters (ADC). No ADC is perfectly made, and there will always be a certain amount of offset between the actual input voltage and what the ADC reads. The ATmega chip has specs about what kinds of gain bias to expect, which indicated that one hundred millivolts was too substantial to be ADC inaccuracy alone. So, what was the issue?

The entire time we had been taking something for granted: the five volt internal reference on the Arduino. Applying a high quality voltmeter to the "5V" pin reveals a very unsatisfactory 5.12 volts. However, in our sketch we assumed that the reference voltage was precisely 5.0V, and the difference was skewing our readings downwards.

The Arduino has a built-in solution to this problem: the analog reference pin (AREF). Applying a more accurate reference voltage to the AREF pin and calling analogReference(EXTERNAL) switches the reference voltage to the external source. There are a number of reference components of varying precision and accuracy, and our original measurement specification of +/- 0.25% dictated our use of a low dropout, high precision, 2.5V reference. A quick voltmeter check reveals that this device outputs 2.504 volts.

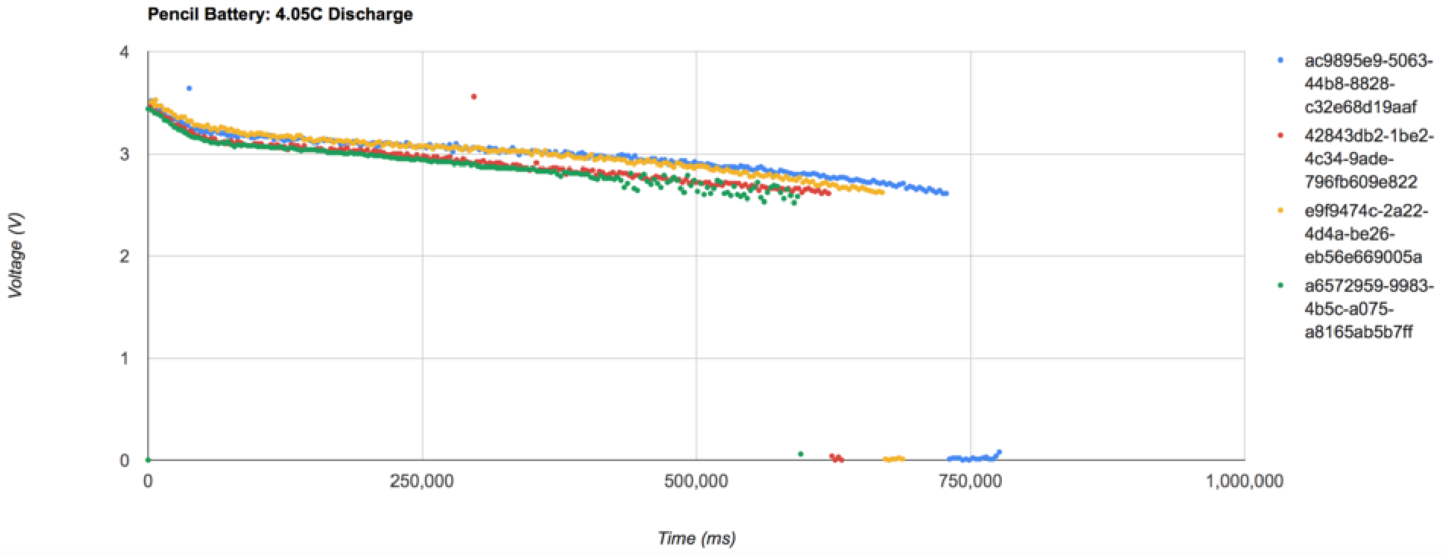
Variable Current Draw

The current drawn from a battery is more or less inversely proportional to the resistance seen by its terminals (real batteries have some intrinsic, internal resistance). So, varying current draw is as simple as changing the resistance seen by the battery's terminals. A variable resistor in parallel with our voltage divider performs exactly this purpose. The barebones schematic can be seen below.

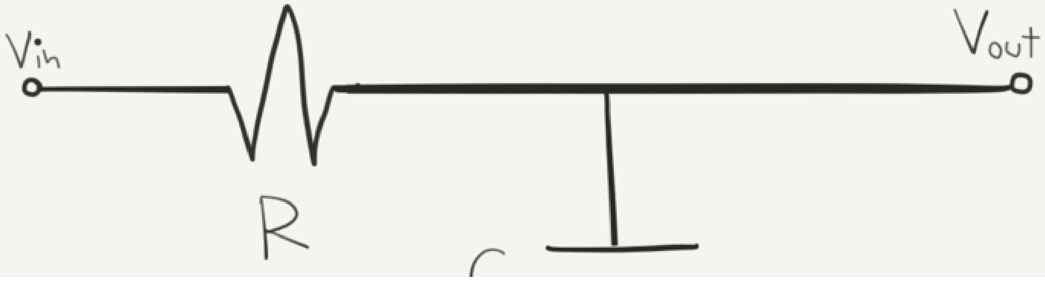


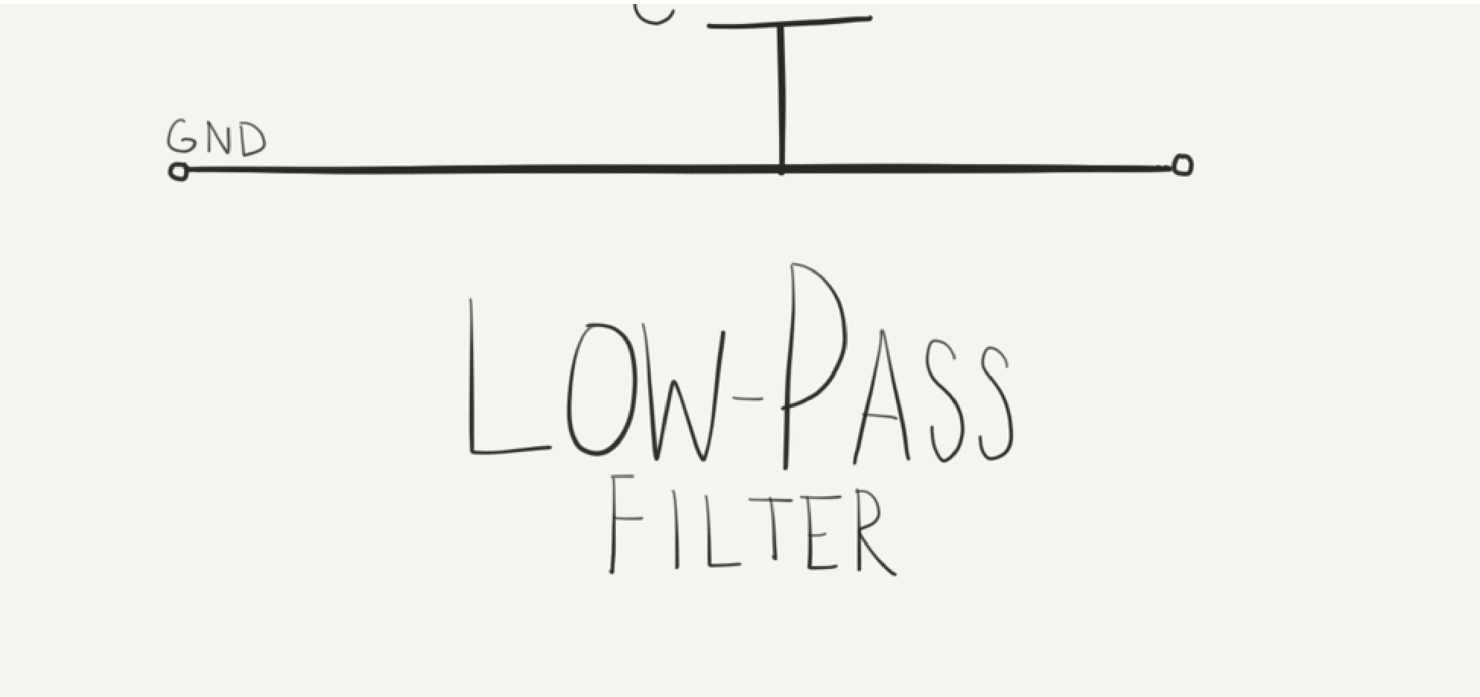
Handling Noise: The Low Pass Filter

With everything in place, we were ready to run a test and see what our data looked like. Much to our disappointment, there still appeared to be considerable noise and an unacceptable lack of precision in our readings.

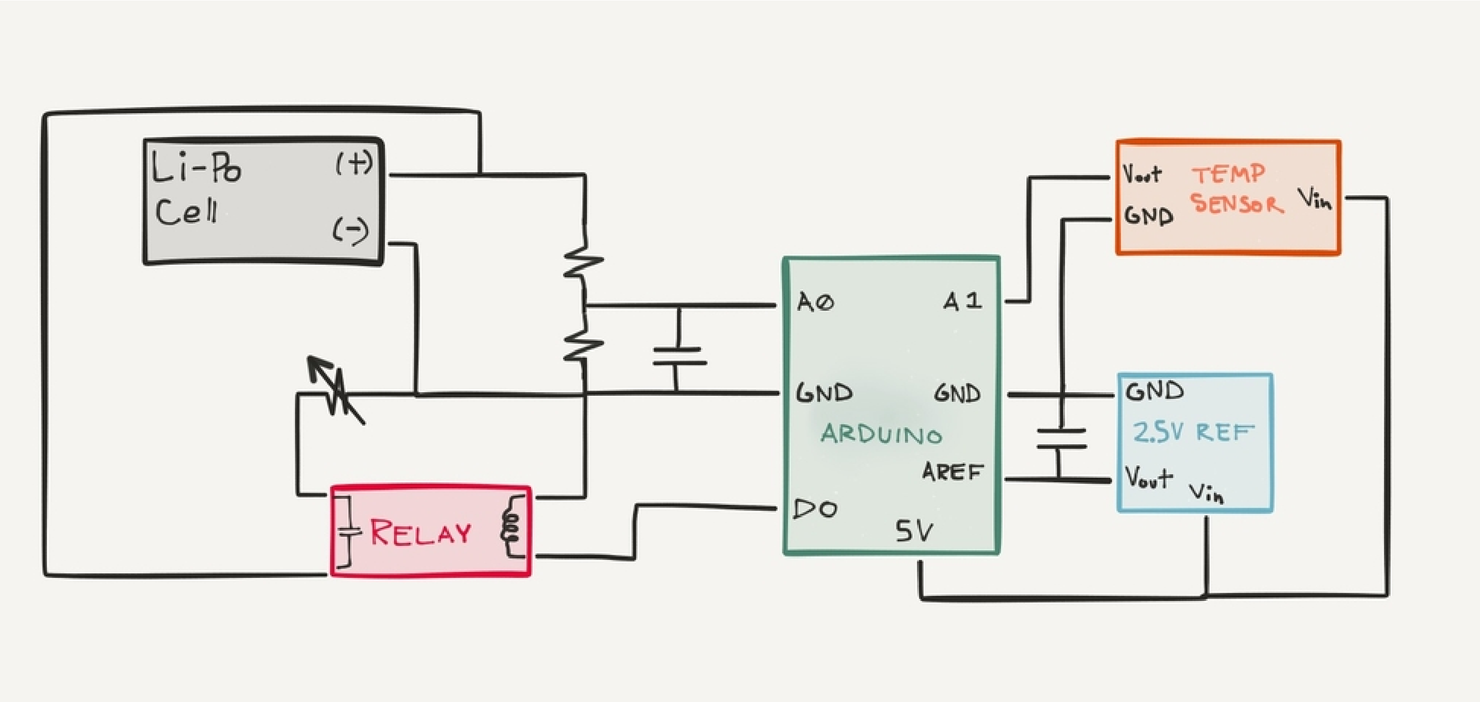


With a solid reference, and precise, highly accurate resistors, what could be causing this issue? We had one last trick up our sleeve: a low pass filter. Low pass filters operate on the principle that the voltage across a capacitor cannot instantaneously change. Momentary spikes or drops in voltage are absorbed by the filter, and as a result, the signal is smoothed.

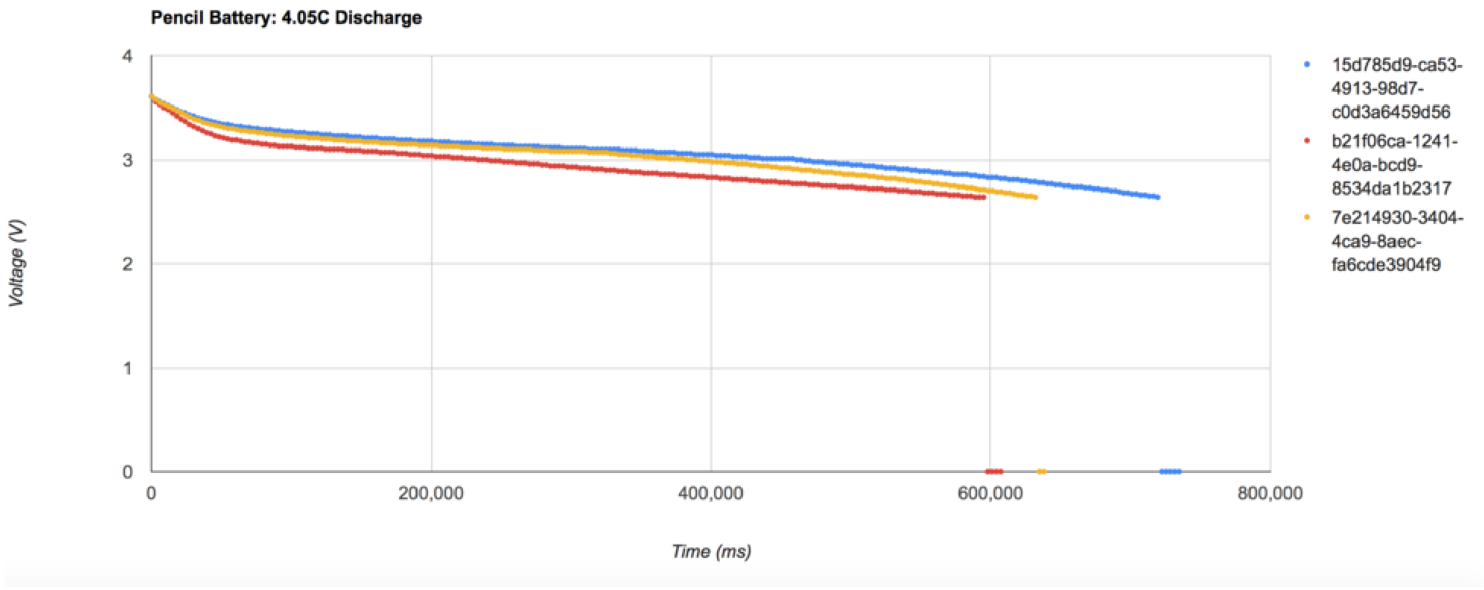




The relationship $f = 1/(2*\pi*R*C)$ describes the relationship between the capacitor, resistor, and the threshold frequency (above which signals are absorbed). LiPo batteries cannot be discharged quickly, and we were sampling every 3 seconds ($f = 1/3s$). Arduino analog inpt pins require as little as 0.5mA to operate, so $R = 10k\Omega$ is appropriate. Solving the aforementioned equation gives $C = 47\mu F$. Empirical testing, however, revealed that the noise was still greatly reduced with capacitors as small as 47 nF. This was some seriously high frequency noise!



We ran a few more tests with this setup, and were extremely pleased with the result. Success!



We now had the ability to collect high quality voltage data using an inexpensive Arduino micro controller, a high quality voltage reference costing less than $5US, and a handful of passive components.

If you'd like more details on this project, or have any questions about our products or team, feel free to reach out to us @FiftyThreeTeam. We welcome any feedback and hope you find our experience helpful in your project!

Makers

David Ferris, Scott Dixon, Julian Walker & Rachel Romano

Pencil by FiftyThree engineering FiftyThree Team

A Little Birdie Told Me...

From Paper To Photo Booth Business In A Snap!

What’s New in Paper 3.5

GREAT IDEAS START ON
PAPER

LEARN MORE

A BLOG BY