Arduino Calibration Caveats

By Robert C. Mazur, VA3ROM, October 2014.

1. THE PROBLEM WITH ANALOG AND DIGITAL CONVERSIONS AND Vcc

Note: Many Arduinos and clones use 10-bit sampling and this is okay, but 12 or 16-bit sampling is preferred. In the specific case of the Arduino Uno, converting an analog value to a digital value is simply **analog value x (Vcc / 1024)** where (Vcc / 1024) = 0.0048828125 volts resolution. Vcc is the positive power rail voltage and often assumed to be *exactly* 5 volts (DC) and "1024" is the number of discrete levels used for 10-bit (2^{10}) sampling (values from 0 to 1023) or 4.8828125 mV (millivolt) resolution. To convert back, we use **digital value x 204.8** where 204.8 = (1024 / Vcc). Sometimes the values 1023 and 204.6 are quoted and this slight difference introduces a 0.1% error which may or may not be an issue (usually not with 10-bit sampling).

However, there's a bigger problem because Vcc is never exactly 5.0 volts (the onboard voltage regulator tolerance is usually +/- 5 to 10%) and worse yet, no two boards' Vcc are exactly the same! Here're some measurements of my Uno's "5.0 V" Vcc using different power sources.

Power Source	Measured "5V" Vcc	Error
Laptop USB (5m cable)	4.80 VDC	4.0%
Laptop USB (1m cable)	4.94 VDC	1.2%
12V AA-battery pack	4.96 VDC	1.0%
9V AC/DC "wall-wart"	4.96 VDC	1.0%

So, it's always a good idea to measure your board's Vcc (use a voltmeter or see the readVcc sketch) and compensate accordingly.

One way to improve things is to use the Arduino's very accurate internal 1.1 volt reference to expand the scale for sensors which don't produce voltages greater than 1.1 volts; the conversion is (1.1 / 1024); for sensors producing less than 3.3 volts but more than 1.1 volts, you can connect the AREF input to the 3.3V output with a simple voltage divider (fixed resistors or multi-turn potentiometers) in combination with the AREF input's 32K internal resistor and we use (AREF / 1024). **IMPORTANT: You <u>must</u> instruct the Arduino that you are using a different analog reference voltage <u>before</u> you attempt to read any analog sensor because you can create a short circuit between AREF and AVCC and destroy your board!** To really improve accuracy, the only way to go is with external 2 or 4-channel 12-bit (4096 samples) and 16-bit (65536 samples) A/D boards with their software adjustable channel signal amplifiers (they also free up the Arduino's limited analog inputs).

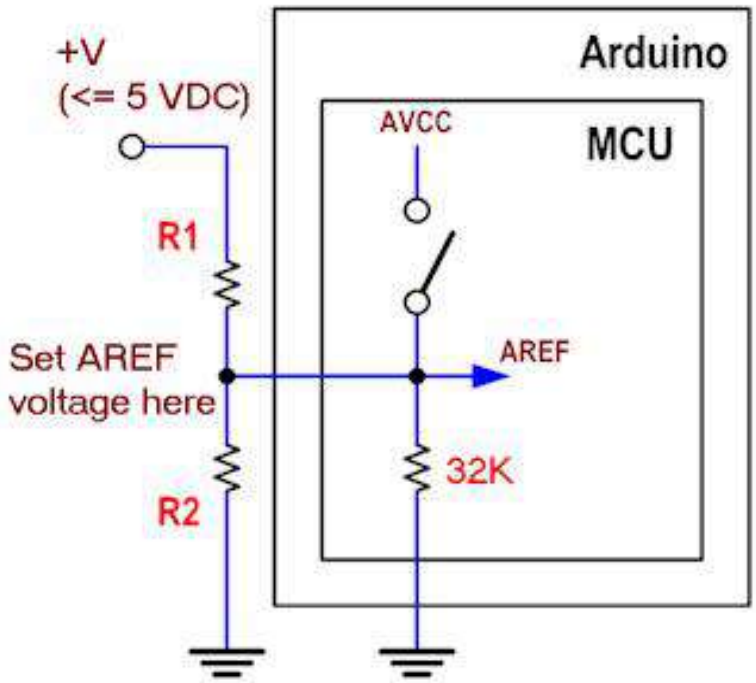

Figure 1: Example External AREF Voltage Divider
R1 and R2 can be individual resistors or multi-turn potentiometers (for precise voltage adjustment(s))

2. TEMPUS FUGIT

Another issue is with uncorrected Arduino timing and processing lag which is caused by assuming that the Arduino's 16 MHz clock crystal is *exactly* 16 MHz with a 0 ppm (parts-per-million) error (typical ppm error is actually +/- 25 ppm) and no two boards have the same ppm errors. A 1-minute delay may be a bit more or a bit less, and as the sketch runs longer and longer, this starts adding up and affects any time critical events. For better timing accuracy, with the ability to do other things during a delay (albeit more complex to use) see Dr. Simon Monk's timer library and examples. Or, you could use hardware fixes with either an external GPS time signal (very, very accurate) or a RTC (real-time clock) module (+/- 1 second per day).

3. “FLOAT” MY GROUND—NOT!

Analog sensors easily pickup up electronic noise to go with the signals they are monitoring/measuring because every solid state device generates some internal noise (to go with the Arduino’s onboard electronics) while other noise is picked up by unconnected or “floating” analog inputs from the ether (the Universe is a very noisy place!). Fortunately, there’s an easy way to fix most external noise by “pulling” unused/unconnected analog inputs to ground via their internal pull-up resistors (100K ohms) and some simple sketch code:

```
pinMode(An, OUTPUT);
digitalWrite(An, LOW);
```

Where ‘An’ is the analog input number (A0, A1, A2, etc.). Analog inputs are dual purpose and can also be used as digital inputs/outputs so we can “pull” this trick off by knowing this fact. *Note: The internal pull-up resistors are too “weak” for current limiting or logic level purposes and you must use external resistors ranging from 220 ohms to 10K ohms for these purposes.*

There is still some residual “dark signal” noise but you can remove it once you determine its value (beyond the scope of this article) but for most hobby applications, you can simply ignore it.

MY FINAL

Well, that’s it!—73

REFERENCES AND RESOURCES

All Things Digital
http://tinyurl.com/og2acxq

Introduction to Arduino (PDF)
http://www.introtoarduino.com

Timer Library (Dr. Simon Monk)
http://www.doctormonk.com/2012/01/arduino-timer-library.html

Using Arduino's Reference Voltages
http://arduino.cc/en/Reference/AnalogReference.