Printout

Sunday, April 19, 2015 4:44 PM

Samuel Lee Toepke

Samuel Lee Toepke is a senior software engineer, working in the National Capitol Region. This is his personal project blog…

Analog Calibration Example

Using a simple mathematical formula, it is possible to calibrate analog sensor readings. If you are using a COTS analog sensor, and have access to a high quality, or more reliable sensor, taking several readings can provide a coefficient and offset for calibration.

In this example, a voltage sensor is implemented with a voltage divider. A digital multimeter is used to glean more accurate readings, and a coefficient/offset is generated with Octave/MATLAB.

All source code can be found on GitHub.

Software:

- Windows 7 Professional x64
- MPIDE v0.23
- Source Code on GitHub
- Octave or MATLAB

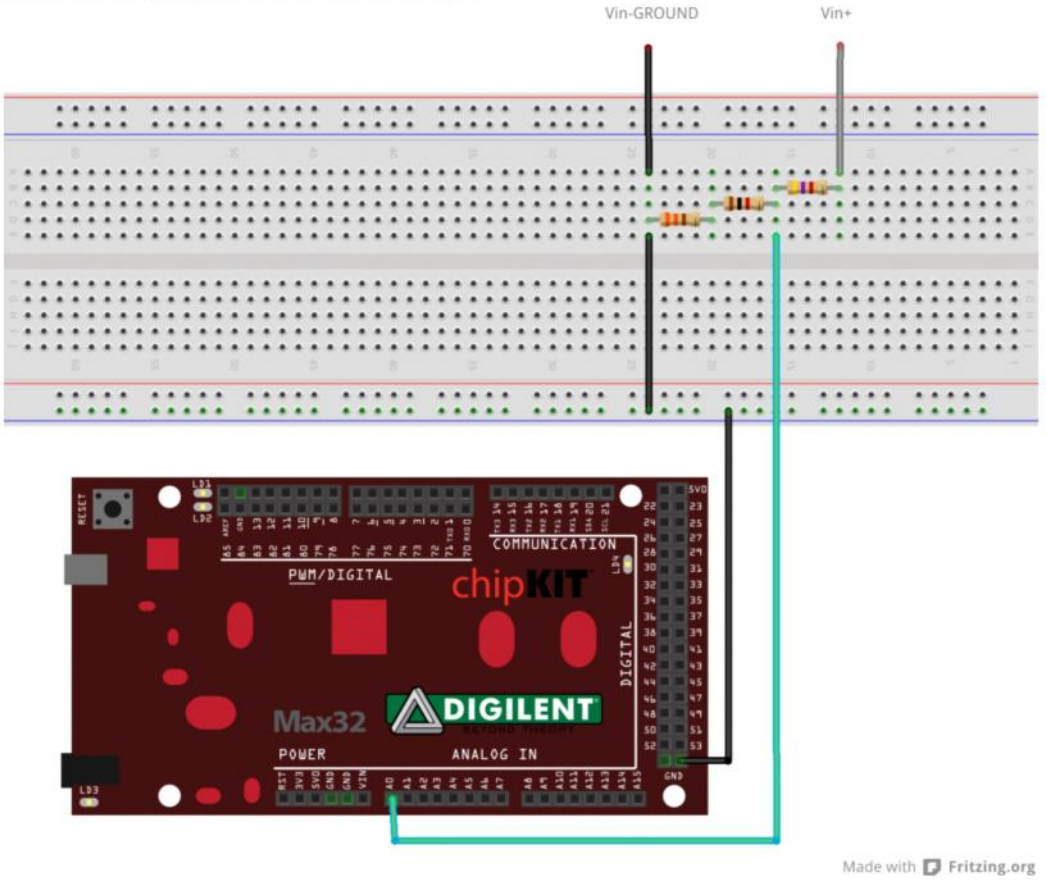
Hardware:

- Digilent chipKIT Max32

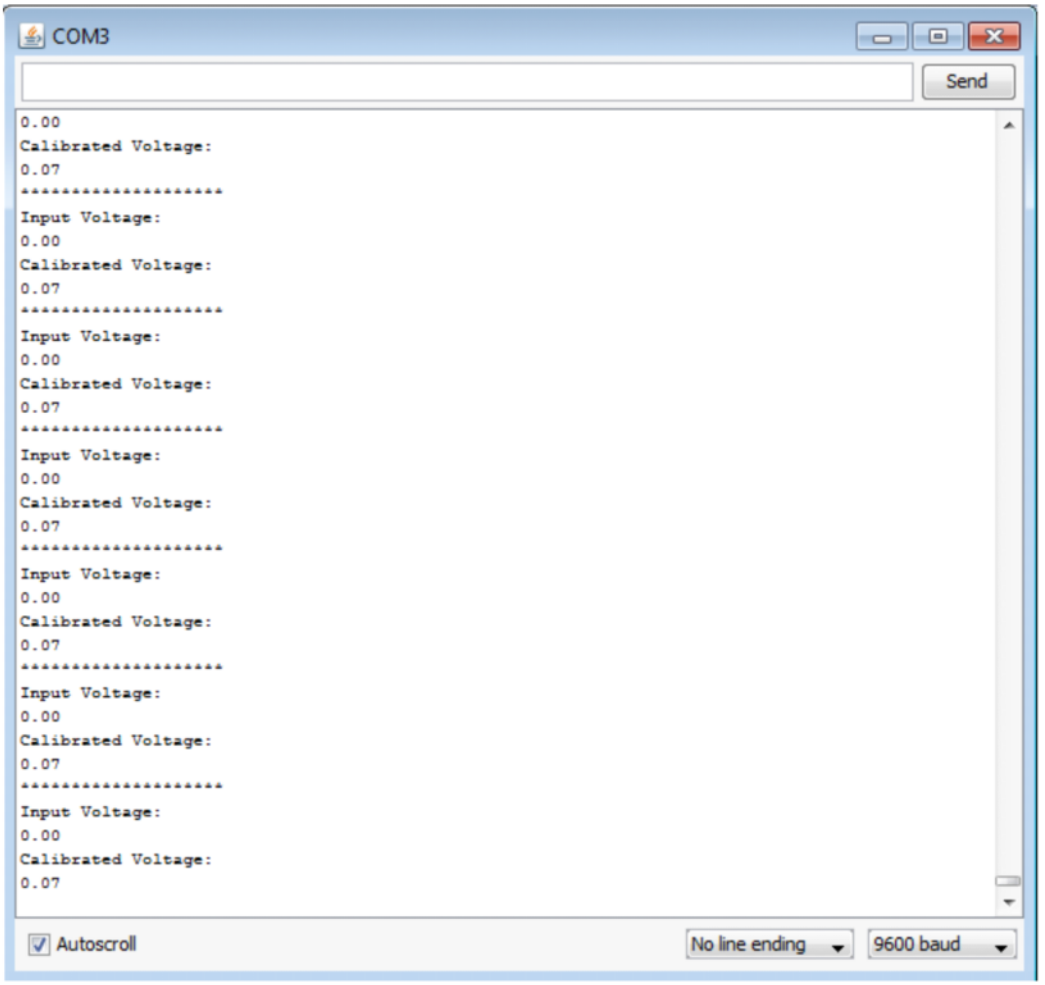
- Mini USB Connector
- Jumper Wires
- Breadboard
- Various resistors
- DMM of choice

Process:

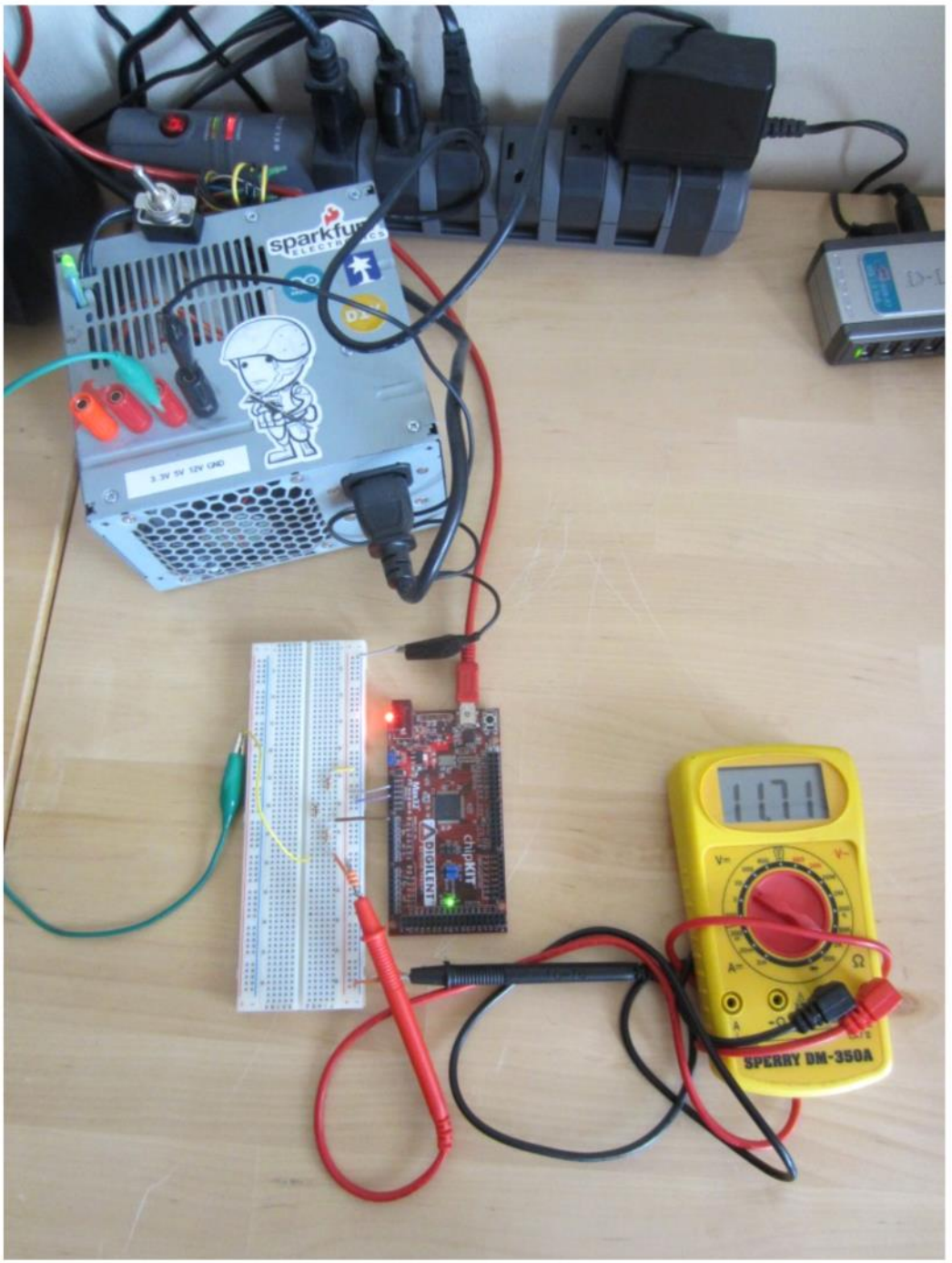
1. Install all software for the MPIDE and the Max32. Make sure you can program 'Blink'.
2. Make the circuit as shown in the schematic:



3. Load the source code in https://github.com/samueltoepke/Voltage_Calibration/tree/master/src/Voltage_Example. Verify, upload, and open the serial port at 9600 baud. You should see a screen crawl like so:



4. At this point, the 'slope_modifier' and 'offset_modifier' in "Voltage_Example.ino" are based off my calculations. Set them to 1 and 0 respectively. Verify, upload, open serial port.
5. Once the code is running, attach your Digital Multimeter (DMM) to GND and Vin on your circuit, and set to voltage reading.
6. You can now attach various voltages to the Vin. Annotate the voltage output from the Max32, as well as the output from the DMM. Full hookup example with a bench DC power supply:



7. The above hookup uses 3 DC power samples, 3.3V, 5v, and 12V. If you have a variable power supply and can get more readings, all the better.
8. In the calibration script, edit the 'dmm' and 'chipkit' variables to have their respective values for each reading. Once complete, run the script with Octave or MATLAB. The result should show the 'slope_modifier' and the 'offset_modifier' along with some error information.

9. Copy/paste the two modifiers into the .ino file in their respective places. Verify, compile, and open the serial port.
10. Running the same test through the different DC values should show more precise readings from the Max32.

Final:

Using simple mathematical methods, this example shows a process to increase the precision of an analog sensor reading. This method will work for any analog sensor, e.g. analog pressure differential or temperature.

This entry was posted in Electronics, Robotics and tagged Electronics, Robotics on 04/19/2015 [http://samueltoepke.com/?p=317] .