Jul 29th, 11:00 AM - 1:30 PM

A Lite DAQ System for Precision Resistance Measurements

Nathan Schmidt
University of Northern Iowa

Timothy E. Kidd Ph.D.
University of Northern Iowa

A Lite DAQ System
for Precision Resistance Measurements

Nathan Schmidt, Dr. Timothy Kidd
University of Northern Iowa Physics Department

Background

Data acquisition (DAQ) systems are frequently utilized in lab settings. Basic DAQ systems are a common occurrence in lab courses for this reason. Commonly, however, the DAQ systems utilized in such courses are proprietary, and do not allow students to understand how they operate beyond a "plug in and go" nature as a result. A further consequence is that these systems are not capable of being programmed, such as is often done with equipment used in professional labs.

The DAQ systems used in lab courses aren't easily replaced by the test equipment they seek to emulate due to cost. Ideally, student labs have enough equipment such that each student may simultaneously carry out a task requiring it. However, pieces of high–end test equipment can easily cost many thousands of dollars. This is a result of such equipment's robustness and guarantee of extremely precise and repeatable measurements. Much of the cost of such equipment results from the extent of these guarantees. As an example: the common 34420A lab multimeter is designed to measure voltages with a precision of 7.5 digits, has environmental compensations and self checks, and can endure $1kV$ over-voltages. While such features are desirable, they aren't remotely necessary in most student labs. Generally, having subpar equipment is preferable to having no equipment.

Objectives

The main goal of this project is the creation of a DAQ system for student labs which replicates the usability of high-end test equipment. This includes the creation of hardware, firmware, and guidelines for the API which the firmware implements.

Hardware Objectives

Cost per device, and thus scalability, is determined almost entirely by hardware. The resistance meter built for this project is notably cost restricted for this reason. A target of $200 was set for the hardware built within this project. This was done with the expectation that optimizations in further iterations, as well as its production at scale, would bring costs down to a quarter of the initial version.

An emphasis was put on allowing a wide range of resistances to be measured at a high precision, but without environmental compensations or self checks. The initial goal was to be able to measure a range of 100Ω to $100k\Omega$ with five digits of precision. An additional goal was added while making the second iteration of the meter of allowing a range of $100m\Omega$ to $10M\Omega$ with four digits of precision.

Firmware and Usage Objectives

Beside designing a hardware meter, providing a computer interface to it is the main focus of the project. To provide an interface similar to professional test equipment requires the implementation of communication with the device over a standard serial interface, as well as the implementation of a set of standardized commands to control the device.

Requirements for the meter were thus set to provide communication over USB, and for commands to adhere to *IEEE 488.2 Standard Commands for Programmable Instruments* (SCPI)[2].

Hardware Design

The device implements a 4–wire resistance meter, where two leads are used to force a known amount of current through the material being measured, while another two leads are used to measure the potential across the material induced by the current. This allows for small resistances to be measured where the resistance added by test leads would otherwise be significant. Another benefit of this design is that the voltmeter is separated from the current source, allowing for the device to additionally be used as a voltmeter.

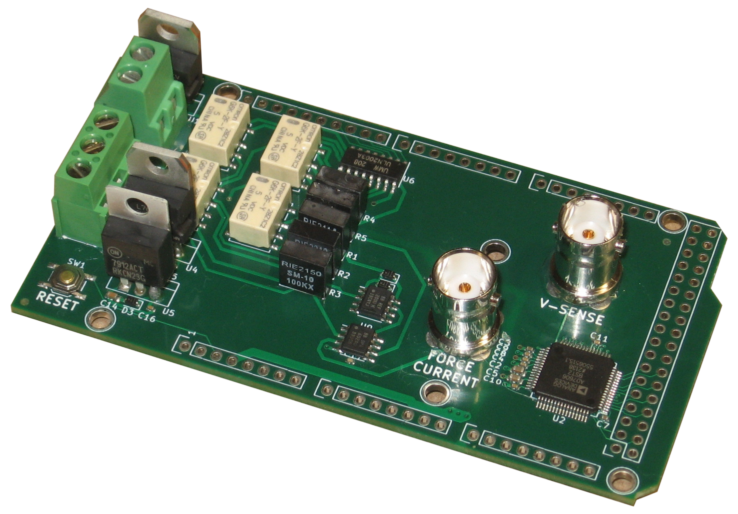

Figure 1: Original design of resistance meter

The circuit can be readily divided into sections for communication and hardware control, current selection and sourcing, voltage measurement, and power input. Communication and general control of the device is implemented by use of an Arduino. Using an Arduino was done to enable simple reprogramming of the device by its users. It also allows for issues with unintentional radio emissions to be avoided, where the FCC would otherwise require radio emission testing if the microcontroller were implemented on–board.

The current source can provide eight different current levels, spanning from $10nA$ to $100mA$, with each level being ten times the magnitude of the previous. The polarity of the output current can also be controlled, which is implemented by switching the polarity of the sources current reference.

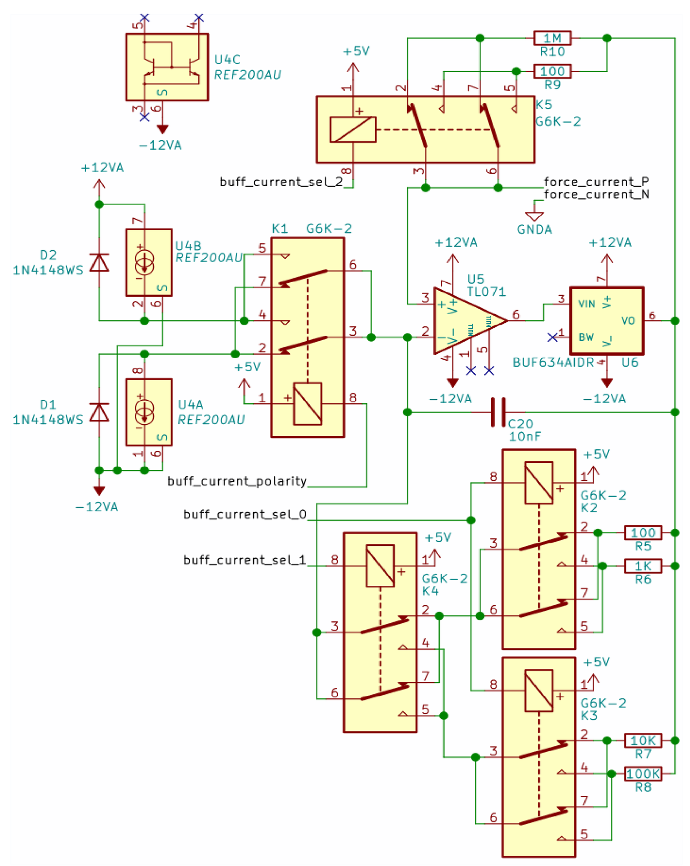

Figure 2: Schematic of the current source

Voltage is measured by a 16–bit analog to digital converter (ADC) with an instrumentation amplifier used to buffer input. The instrumentation amplifier provides a high impedance input which prevents high resistance measurements from being skewed. The first implementation of the meter lacked input buffering, instead relying on the input impedance of the ADC ($4M\Omega$). As a result, resistance measurements approaching $4M\Omega$ had a substantial error which could not be compensated for due to the non-linearity of current draw by the ADC's input circuitry.

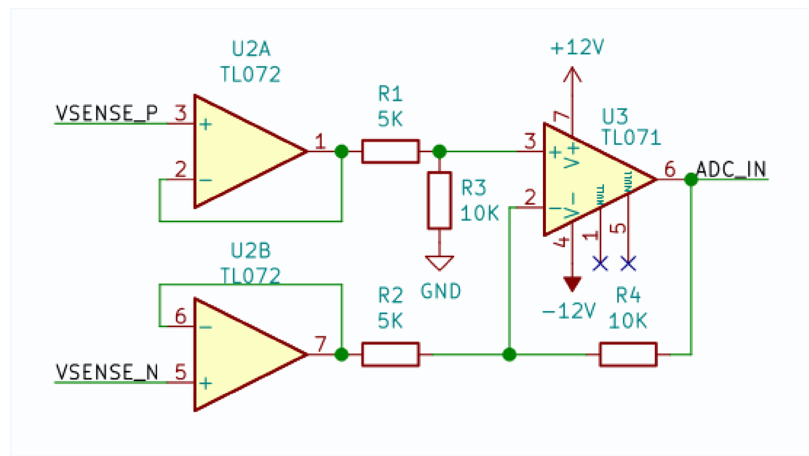

Figure 3: Schematic of the instrumentation amplifier

The initial device relied on an external power supply and a separate battery pack to operate. A custom power supply has since been designed to replace both with a single unit. The new supply conforms to the Eurocard standard, allowing it to fit alongside the resistance meter PCB in a generic enclosure.

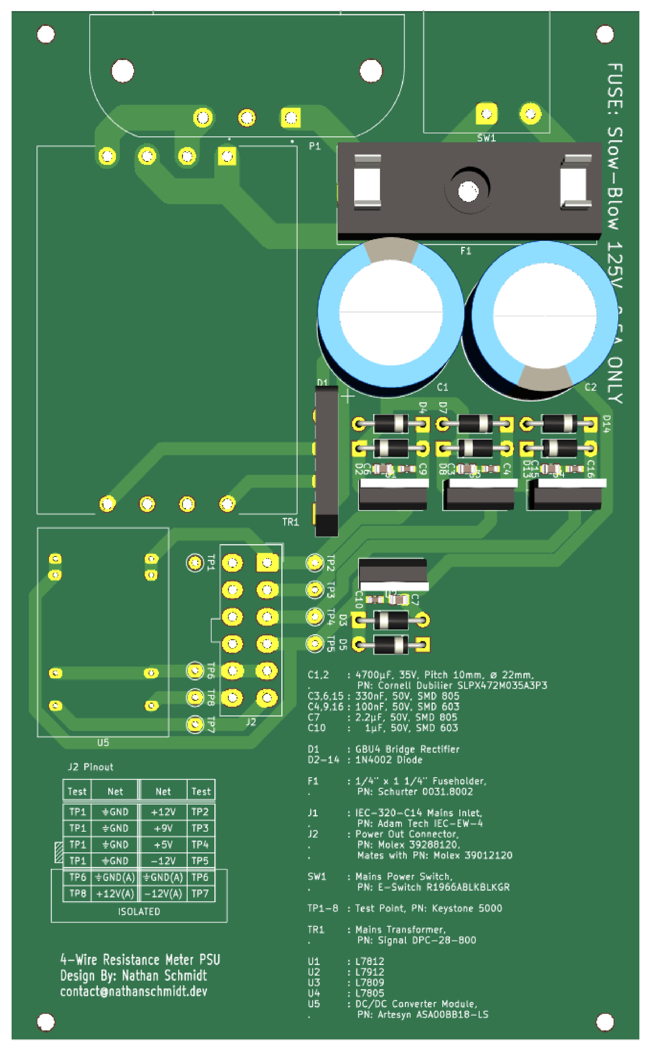

Figure 4: Rendered image of the power supply for the next device version

Firmware Design

Firmware is written in Arduino C and utilizes the freely licensed *Vrekrer SCPI Parser Library*[1]. The SCPI standard[2] specifies a base set of commands and additional commands for specialized instruments. Currently only SCPI status and error commands still need to be added for full conformance to the required commands. Specialized commands have also been implemented, including for manual range adjustments, auto-ranging, and for the measurement of voltage in addition to resistance.

Results

The initial version of the device fell short of the initial goals set out for precision, dropping off to two digits of precision when measurements approached $100k\Omega$ due to its lack of input buffering. Precision to five digits was seen from 100Ω to $1k\Omega$, however, which gives promise that the next hardware version will fix this just by its inclusion of input buffering.

Regardless of the precision of its measurements, the first version did allow for a platform to develop firmware on, and further allowed for a subsequent design which is expected to fix its shortcomings. It can be considered quite successful for these reasons.

Future Work

A second version of the device has already been designed in order to resolve the limitations of the first. The construction of the device is already underway, after which testing is required.

Firmware should be compatible between the two versions, needing only some pin numbers to be redefined. This only leaves the implementation of the SCPI status and error commands before the device can be considered complete.

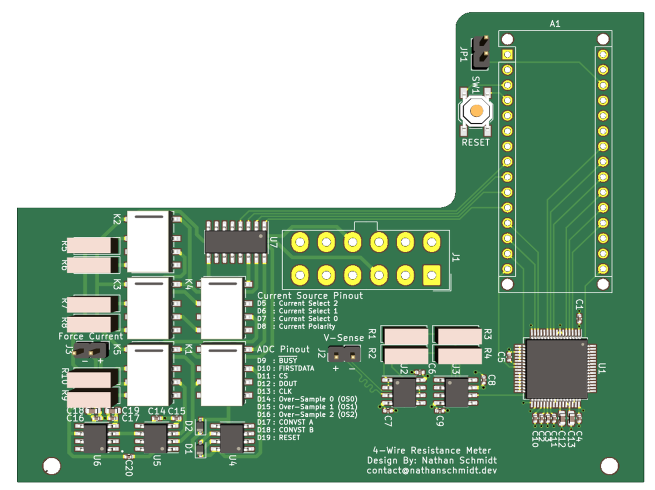

Figure 5: Rendered image of the next device version

References

[1] Diego González Chávez. Vrekrer scpi parser library.
[2] SCPI Consortium. Scpi–1999 specification, volume 1–4.

Acknowledgments

The work was supported by the grant *Grant NO: DE-SC0020334* funded by the U.S. Department of Energy, Office of Science and a NASA EPSCoR Research Building Seed Grant.