MEng Thesis Proposal
Circuit Sensing Breadboard

Josh Gordonson
Advisor: Ed Moriarty

January 1, 2014

1 Motivation

The breadboard has been a staple substrate for electronic construction over the last century. At the dawn of a growing interest in amateur radio, resourceful tinkerers used planks of wood to secure and ruggedize their electrical handiwork. Conductive nodes, such as nails or copper rails, were driven into the non-conductive boards, providing anchors and contact points that were electrically isolated from the rest of the circuit. Components were soldered or wire-wrapped to the nodes, and sometimes secured by non-energized nails or screws. This construction technique provided a lot of artistic freedom in circuit construction, but was relatively time consuming and required relatively heavy hand tools, such as a hammer or drill. The solderless breadboard is the canonical tool given to students taking

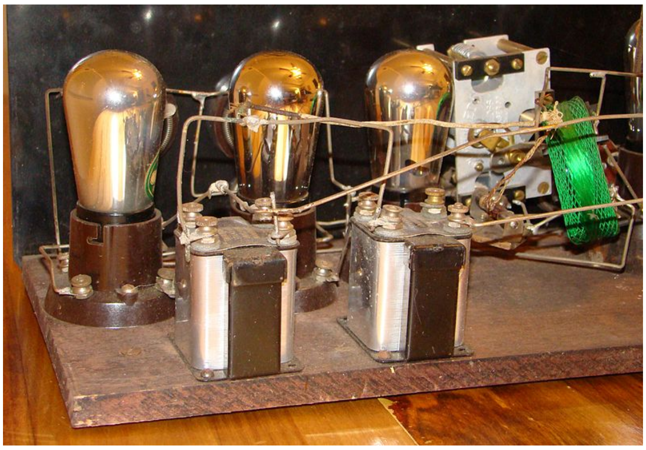

Figure 1: Radio mounted on wooden breadboard

introductory courses in the field. Rather than driving nodes into arbitrary locations, component leads are inserted into contact points that allow rapid semi-rigid construction of circuits with no other tools. The layout of a solderless breadboard is designed to be compatible with a plethora of powerful integrated circuits, enabling complex designs. Modern solderless breadboards have come a long way from their namesake wooden ancestors, but there is still room for improvement.

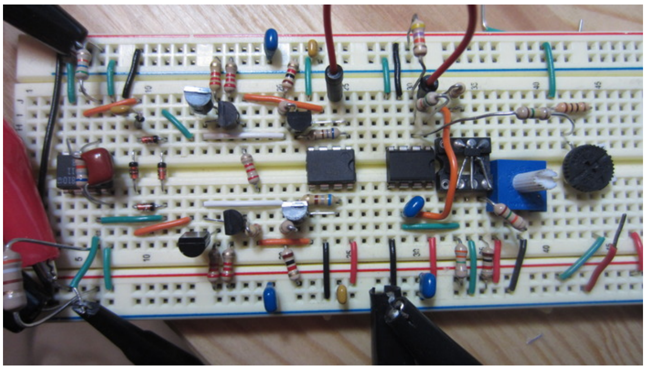

Figure 2: Voltage controlled triangle wave generator on a modern breadboard

The shortcomings of solderless breadboards lie entirely within the art of breadboarding. The intent of breadboarding is to physically realize a circuit. Often, this involves designing or using a reference schematic to guide construction, though circuit improvisation is not uncommon. Inserting components and jumper wires into contact points is straightforward, but poor contacts, broken wires (inside insulation), and mis-inserting leads can plague designers for hours on end and/or destroy components. A meticulous breadboarder can successfully realize a circuit without error, but for the uninitiated it is difficult to justify the additional time and care required to plan and build. Breadboarding is a skill that is learned over time, but small errors can lead to excessive frustration and turn students off to the field. To satisfy the requirements of the Masters in Engineering program, I propose a solution to some of the issues with the solderless breadboard.

2 Overview

A confident linkage between the the electrical and mechanical domains is required to construct a circuit. On larger scales, the mechanical structures (eg. contacts, wires, components) that create electrical circuits are visible in plain sight. It is simple and reliable, then, to determine the circuit representation of a mechanical structure that has no hidden connections. Breadboards often obscure connections due to their construction. The regular nature of a breadboard, a grid of contact points arranged in rails, makes it difficult to keep track of which rail is connected to what circuit node. When combined with the opaque nature of most components and the poor reliability of breadboard contacts, visually verifying complex circuits on a breadboard becomes infeasible. Many of the inherent problems with breadboards stem from this open-loop nature of breadboarding, where visual cues are not enough to determine the electrical circuit from the mechanical structure. A symptom of open-loop construction techniques is a mismatch between the mental model and the physical realization of the system at hand. I seek to close this loop by designing and implementing a circuit-sensing breadboard.

At first glance, the proposed problem is straightforward. A standard-issue breadboard has 130 conductive rails. Each of these rails can be connected to an array of ADCs (analog to digital converters) and DACs (digital to analog converters) controlled by a microcontroller. With access to every node in the circuit, by iteratively stimulating and probing the circuit it is possible[1] to determine the circuit topology constructed on the breadboard. Once the topology has been determined, a schematic representation of the constructed circuit, from the perspective of the breadboard, can be displayed. This visual display will provide an accurate depiction of the electrical circuit at hand, effectively closing the loop on breadboard construction.

Designing a circuit-sensing breadboard will require a substantial amount of hardware and software infrastructure. Fortunately, the majority of this infrastructure is useful in its own right. A hardware test-bench will have to be constructed to interface each of the 130 rails on the breadboard with a hand full of ADCs, DACs, and the microcontroller. The test-bench should include a pcb-mounted breadboard, an ADC and DAC multiplexing board, firmware to control sampling and stimulating the breadboard, and software to stream the sampling data to a computer for display. Effectively, the test-bench is a 130-channel digital breadboard-oscilloscope/function generator. Since software is faster to prototype with, I will be doing most of the theoretical circuit topology work in software. A test-bench for a circuit simulator will have to be built. Ideally, this test-bench will allow a circuit-sensing program to probe and stimulate node voltages of an automatically generated circuit, as if it were running on a microcontroller connected to the hardware test-bench with an arbitrary circuit built up.

After the test-benches have been constructed, they need to be integrated and the circuit-sensing program will be tested on physical circuits. Finally, a schematic drawing program will be written to visualize the circuit-sensed topology.

3 Details

3.1 Breadboard Interfacing

In order to sense a circuit constructed on a breadboard, we must be able to connect to every node on that breadboard. There are many ways to accomplish this, of which a few are listed below. For instance, insert ribbon cable terminated in 0.1" headers into every rail on the breadboard, then wire those headers to an auxiliary board. This method has the drawback of consuming one of the 5 tie-points on that rail, leaving only four for the end user. Another solution is to remove the double-sided

[1] I have not yet shown this.

foam tape that insulates the back of the breadboard and use spring-loaded [pogo] pins to provide contact with the rails. The pogo pins will require a mechanical jig to keep them vertical, and a snap-lock flexure to hold the breadboard against the pins. Most pogo pins are tall ($\sim$.5" or taller) and expensive relative to standard 0.1" pin headers. A better solution is to surface mount each rail to the PCB, then press on the plastic breadboard enclosure.

I conducted some experiments on soldering PCB finger springs. The lower-quality breadboard springs were easily bent and difficult to solder to, while higher-quality breadboard springs were more robust and easier to solder to. A PCB with a breadboard rail footprint was etched and stuffed with six finger springs, using nail polish as soldermask. The 6-rail breakout board was designed with a row of six 0.1" header pins to fit into the ADC header on an arduino. A quick arduino sketch was written to sample each of the rails and transmit them to a processing sketch that plotted them in real time on a computer. The prototype breadboard breakout functioned as expected.

3.2 Microcontroller

An STM32F407vg 32-bit ARM Cortex-M4 microcontroller was selected for its 168MHz clock rate, 3 ADCs (24 channels), 2 DACs, USB On The Go (OTG), 1 MB of flash, 192 KB of RAM, $15 price point, and the open-source community that surrounds it. The microcontroller also has DMA (direct memory access) to support asynchronous data flow from the ADCs and to the DACs. This should be more than enough computing power to handle the tasks at hand. Although I am not fluent in STM32 C, I am in the process of learning using their STM32F4Discovery board and libopencm3. I have successfully written firmware that accesses GPIOs (general purpose input/output), retrieves data from the ADCs, and transfers that data over a serial port. Additional work will include writing an ADC scheduler to determine when each ADC needs to sample, a trigger to determine when to start sampling a waveform, and accessing and configuring USB OTG, to name a few.

The Discovery board form factor is close to ideal for this application. A daughter board will be made to connect the Discovery board to the PCB-mounted breadboard for development. Eventually, a standalone board may be designed and fabricated to minimize the size of the circuit-sensing breadboard.

3.3 Node Sensing and Stimulation

A linear, time invariant circuit's topology can be extracted from the state variables associated with the voltage at each node and the current through each element. [needs citation] Given a self-imposed constraint that the sensing device may not break the circuit that it senses, there is no way to measure the current through each element independently. This is a reasonable constraint to wish to work with, as allowing the sensing device to break each rail into its five tie-points increases the overall complexity by adding more than an order of magnitude of node combinations. Assuming that we are only sensing the voltages at each node, we will need a way to connect at least one ADC to every node.

Starting from the 130 finger spring rails, care must be taken to not disturb the sensed circuit's behavior. In many cases, nodes with high-impedances are extremely sensitive to noise and current injection, so extremely high-impedance voltage buffers will be used to drive the rest of the sensing circuitry. The 130 buffers could subsequently feed directly into 130 ADCs, but the cost of high performance ADCs makes this prohibitively expensive. Instead, analog multiplexers will fan-down the 130 buffered nodes to 8 high performance ADCs. The on resistance of the multiplexers won't be an issue, since the input impedance of the ADCs is relatively high and the output impedance of the buffers will be relatively low. The three ADCs on the STM32F407 are relatively high-spec'd ADCs, but their error and noise margin is sub par. Higher performance 12-bit 1-MSPS ADCs will be needed, and are available in small form factor with a high-speed serial interface.

Node stimulation will also be necessary to sense the circuit topology, since breaking the circuit is infeasible. There are two ways to go about node stimulation - voltage stimulation or current stimulation. Voltage stimulation would involve connecting buffers to multiplexed DACs, fanned-out to all of the nodes. Current stimulation would require an additional block at each node that converted the voltage stimulation signal into a perturbation current. Each has its advantages, but I have not completed exploring these methods analytically.

3.4 Breadboard Emulation

While the hardware design remains unfinished or on lead time, work can be done on the circuit analysis portion of the problem. A breadboard emulator was written using python and NGSpice. The emulator is used to provide a circuit-sensing program (to be written) with the raw data that node sensing and stimulation board would collect for a given circuit topology. It works by feeding NGSpice (an open-source circuit simulator) a given netlist and voltage probing each available node. It then returns the list of nodes with their measured voltages to the circuit-sensing program, when can then edit the netlist to change the node to stimulate and run a simulation again. Eventually, the netlist may be automatically generated to test the circuit-sensing program.

To test the breadboard emulator, a resistor divider network solver was written. The solver was able to predict the two values of resistors in the divider with .1% accuracy.

3.5 Circuit Sensing

A circuit sensing algorithm (CSA) that determines circuit topology and element values by probing node voltages will need to be designed. There are two ways to go about determining the circuit topology constructed on a breadboard - analyzing the final constructed topology, or analyzing the incremental topology as the circuit is constructed from a bare breadboard. The scope of this work should only require the latter to be implemented, as incrementally building a circuit model is much easier than attempting to analyze an entire network at once only using voltage probes and sources.

To begin, the CSA will be designed for resistor networks. This slightly reduces the complexity from networks that include capacitors and inductors, since AC or transient analysis is not required. It's likely that the algorithm will be composed of a stimulation loop that iterates over all of the nodes, collecting DC operating point changes. To determine absolute values, some reference will be required. This means that either the source current for the voltage stimulator will need to be measured, the voltage output of the current stimulator, or an in-line calibrated resistor will have to be used. If the resistor network problem is solvable, RLC networks should also be solvable by stimulating the circuit over a swept frequency range and measuring phase lag between nodes.

Adding support for nonlinear elements and integrated circuits is also desirable, but more difficult. It would require pre-programmed models, pinouts, and likely a custom testing procedure to identify each device. An alternate method may involve a dedicated device characterizer with one-off device detecting circuitry that a user would insert a component into before placing it on the breadboard. Another alternative could involve detecting ICs by their characteristic power rail locations, ground on the bottom left and VCC on the top right, then prompting the user to select the component they are placing from a menu. Although the circuit-sensing breadboard may seem unpractical without the ability to detect integrated circuits and nonlinear components, implementing this functionality may is beyond the scope of this work.

3.6 Schematic Drawing

Once a topology is determined, a schematic generator will be used to display the circuit topology. There exists many libraries that support force directed graph rendering. One such library is D3, a javascript library written by an open source community. In addition to a schematic drawing, a netlist will also be generated, allowing for comparison between simulated and experimental results.

4 Project Risks and Difficulties

When embarking on a design project of this depth, there are always risks, uncertainties, and difficulties to consider. Writing a circuit sensing algorithm will not be an easy feat. As I have not yet explored the mathematics behind voltage-based topology reconstruction, I cannot comment on how likely I will be to succeed. However, I have worked on the incremental modeling technique for the past semester and am confident that it can be written.

Converting the circuit sensing algorithm to firmware and hardware will likely prove difficult. The crux of the problem stems from not being able to probe the current in each node. However, we can approximate measuring the current in each node by placing a known resistor in series with the source voltage. By measuring the voltage across the known source resistance, we can find the source current, which should help to determine the current in the added element.

5 Related Work

There has been a substantial amount of work related to designing better learning tools over the last decade. In particular, the open source hardware movement has produced some amazing educational tools, such as little bits, lightup.IO, and a range of portable test equipment. Many of these projects approach the design goals that I have set for this thesis, but fall short in one way or another.

Little Bits is a media lab spinoff that encourages play with electronic modules by making it easy to connect them. Little Bits uses proprietary magnetic headers that ensure good contact between 'bits' and form a digital communication bus. Although there is no schematic representation of what users are constructing, the 'bits' themselves are portrayed like blocks with inputs and outputs. The 'bits' are rather complicated pieces of hardware, often combining a microcontroller or multiple pieces of digital logic to perform simple functions. This enables the creation of more complex circuits with fewer components. Little bits are excellent for introducing people to circuit play, but restrictive in the sense that there is a limited amount of modules, the form factor is bulky, and they are expensive.

Lightup.IO also attempts to encourage play by mounting raw passive components on substrates with magnetic headers. Unlike Little Bits, each piece is composed of one electronic component, no digital communication bus. The pieces are all tagged with visual codes similar to QR codes that allows image processing to determine the circuit topology. A virtual reality mobile app was written that overlays a circuit simulation on top of the physical circuit. The design is excellent, but again each component must be attached to magnets, resulting in a bulky, expensive learning tool. The educational technique of providing multiple representations of systems under examination is called

Bimodal Modeling. It provides students with the opportunity to draw parallels between different methods of understanding, allowing them to traverse the space of reasoning.

Seeed Studio is a Chinese open source hardware design and manufacturing firm. One of their flagship products is the DSO nano, a digital sampling oscilloscope smaller than a smart phone. Although its specifications aren't all that impressive (200KHz analog bandwidth, 1MSPS sampling, 2 channels), the $90 price point is right for an introductory tool.

Integreight, and electronics design firm in Egypt, has spent the last few years designing 'smart breadboard' - a breadboard that removes the need to connect components together with wires. They do this with analog multiplexer ICs and software that allows the user to draw the wires on a digital breadboard, while the 'smart breadboard' configures its analog switches to match the digital representation. To aide with placing components, LEDs surrounding the breadboard indicate where ICs should be placed. The 'smart breadboard' successfully reduces the time it takes to breadboard a circuit by removing the need to cut, strip, and bend wires into place. Unfortunately, it leverages expensive components in large quantities to perform this function. Likewise, the analog switches and routing are not conducive to high-frequency design techniques. Nonetheless, 'smart breadboard' is an interesting prototyping tool and is on the right track.

The tool that I propose to build is in some sorts an amalgamation of all of the aforementioned projects. The goal is to present a Bimodal Modeling perspective on circuit design and construction while providing a comprehensive set of tools to explore with.

6 Projected Timeline

It's likely this work will extend into the summer of 2014

November 2013 - Conduct component search for ADCs and multiplexers

December 2013 - Develop software test-bench

January 2014 - Research breadboard manufacturing and surface mount soldering in Shenzhen, China

February 2014 - Develop incremental resistor network sensing algorithm

March 2014 - Continue work on resistor network sensing algorithm

April 2014 - Design and fabricate hardware test-bench

May 2014 - Integrate hardware test-bench with incremental resistor network sensing algorithm

June 2014 - Begin writing thesis, Develop RLC network sensing algorithm

July 2014 - Continue writing thesis, Develop IC sensing algorithm for a handfull of ICs

August 2014 - Finish thesis

7 Facilities and Equipment Requirements

The majority of this work can be done on a personal computer and a small electronics lab setup. I own all of the pieces of equipment I need to do the hardware and software work. However, access to the undergraduate electronics lab in 38-500 and MITERS in N52 is available if need be.