

MASSACHUSETTS INSTITUTE OF  
TECHNOLOGY

MASTERS OF ENGINEERING THESIS  
PROPOSAL

---

**Circuit Simulation with  
Automatic Protoboard  
Layout**

---

*Author:*

Michael MEKONNEN

*Supervisor:*

Dennis FREEMAN

April 20, 2013

# **CIRCUIT SIMULATION WITH AUTOMATIC PROTOBOARD LAYOUT**

## **1. INTRODUCTION**

In this paper, I will describe the project I plan to complete for my Masters of Engineering Thesis. I will design and implement a new teaching tool for 6.01, Introduction to Electrical Engineering and Computer Science I. The tool will allow students to draw schematics of circuits and analyze them. Furthermore, it will automatically generate the corresponding protoboard layouts for the schematic drawings.

In Section 2 I will discuss a comparable tool, Circuits Maximus (CMax), that is already available for 6.01 students. In Section 3 I will discuss the motivation for building a new simulation tool and its potential improvements on CMax. In Section 4 I will discuss previous work (or lack there of) done related to the simulation tool I plan to build. In Section 5 I will go into a little bit of detail regarding the implementation of the tool. Finally, in section 6 I will discuss some of the possible risks I anticipate while completing this project.

## **2. BACKGROUND**

My focus in this project is the circuits module of 6.01. In this module, students get an introduction to the world of circuits and get hands-on experience through various labs. In a typical circuits lab, students first design a circuit by drawing a schematic diagram of the circuit and

discuss their design with a staff member. After they iteratively amend their design and are happy with it, they build the circuit on a simulation tool called Circuits Maximus (CMax)[1]. With this tool, students can layout their circuits on a simulated protoboard as if they were laying it out on a physical one. CMax allows students to test the circuit to make sure that it behaves as desired. Circuit layout is much easier on CMax than on a physical protoboard. Hence, CMax provides a very fast and safe way of debugging circuit layouts. Once the students are satisfied with their observations from CMax, they build their circuits on physical protoboards and carry out the appropriate experiments.

CMax has been a fantastic resource for 6.01 students. Its introduction has made learning circuits significantly easier for many students, especially those that have little or no prior experience with circuits. In addition to making the lab exercises much more manageable, it provides students with a very handy way to build, analyze, and experiment with circuits at their own leisure.

### 3. MOTIVATION

While CMax is a fantastic tool, we can imagine a tool that can be even more useful. The most instructive part of the labs that students do in the circuits module of 6.01 is really designing the circuits in the first place, which they currently do by drawing schematic diagrams on paper. Once they are happy with their schematic diagrams, they proceed to laying out the corresponding circuits with CMax. The process of laying out a schematic drawing of a circuit on a protoboard (be it a physical one or on CMax) does not really have very much instructive

substance. This process is essentially solving a puzzle, and has almost nothing to do with the subject matter – designing circuits. In fact, when the circuits get complicated and involve several pieces, translating a schematic diagram into a protoboard layout gets to be quite challenging and time consuming. In these situations, students often end up with convoluted and unpleasant layouts that are very difficult to debug in the likely case of the circuit not behaving as expected.

Hence, in the best case scenario, students should not have to produce protoboard layouts for their schematic diagrams. Indeed they should work out the right schematic diagram of the circuit of interest, but the layout generation should not be part of the learning process. My MEng Thesis project will, therefore, be a tool that lets students draw and analyze schematic drawings of circuits and produces the corresponding protoboard layouts automatically. Given the protoboard layouts output by this tool, students can proceed to building the circuits on physical protoboards and carrying out the appropriate experiments.

With this tool, a typical circuits lab would go as follows. First, as before, the students draw schematic diagrams of their circuits on paper. Once they have schematic drawings they are happy with, they can directly draw their schematic drawings on the simulation tool. In fact, students may go straight to building the schematic drawings on the simulation tool, bypassing the experimentation on paper. Once they have a schematic drawn, they can analyze it with the tool, discuss it with staff members, and amend it easily and quickly with the user-friendly graphical user interface of the simulation tool. When they

are satisfied with the behaviors of their schematic circuit, they can produce the corresponding protoboard layout simply at the click of a button – this would be the most important advantage of this tool. They can then build the layout on a physical protoboard and carryout experiments with it.

#### 4. PREVIOUS WORK

As discussed so far, the simulation tool I will build is meant to be an improvement on CMax, and it will be tailored specifically for 6.01. As such, the only directly related previous piece of work is CMax. I have already discussed the motivation behind creating this new simulation tool despite the availability of CMax.

There are several very general circuit simulators available, but none that directly fit the material presented in 6.01. In addition, I was not able to find any tools that automate protoboard layout. On that front, the new simulation tool I plan to build will provide something novel.

#### 5. IMPLEMENTATION

In keeping with the implementation of the current tools and infrastructure in 6.01, I plan to build my circuit simulation tool entirely in the Python programming language. This suits me very well because I have significant experience developing software in Python. The language is flexible enough to allow any of the things I plan to do in this project.

There are three major pieces to this simulation tool.

**5.1. A user-friendly GUI.** It is important that the GUI be very easy and intuitive to use to allow for quick building of schematics as well as quick amending of schematics already drawn. The GUI will essentially be composed of two parts. The first is a large (initially empty) area in which the schematics will be drawn. The second is a palette that provides all the available circuit components. The components will be precisely the components necessary for 6.01 labs, i.e. the ones currently available in CMax. Students will be able to select circuit components from the palette to put in the drawing area, and they will also be able to connect different circuit components using wires. The GUI will also allow easy saving and opening of files, undoing and redoing actions, etc.

I will use the Tkinter module, the standard Python module to write GUIs, to implement my GUI[2].

**5.2. Circuit Analysis.** The tool will provide the ability to analyze circuits by allowing students to probe voltages and currents in the schematics they draw as well as by producing the responses of some of the circuit components (motors, for example) to the particular circuit. The circuit analysis capabilities of my circuit simulator will essentially be identical to that which is currently provided by CMax.

**5.3. Automatic Protoboard Layout.** This is of course the most novel and important part of the project. The way I will tackle the automatic layout problem is by formulating it as a search problem. The very general steps to this process are given below, and these steps

will be elaborated upon in much further detail in my final Thesis write-up:

- (1) Extract all the circuit pieces from the schematic drawing.
- (2) Figure out how the pieces are to be interconnected from the schematic drawing.
- (3) Figure out a good way to place the pieces on a protoboard so as to require as little wiring as possible.
- (4) Figure out a way to wire up the pieces. This step can be formulated as a search problem, in fact 6.01 provides a great infrastructure for this using the  $A^*$  search algorithm[3].

## 6. RISKS

TODO

## REFERENCES

- [1] Circuits Maximus, [http://mit.edu/6.01/www/cmax\\_docs/cmax\\_docs.html](http://mit.edu/6.01/www/cmax_docs/cmax_docs.html).
- [2] Python Tkinter module, <http://docs.python.org/2/library/tkinter.html>.
- [3] 6.01 Course Notes, Chapter 7: Search, <http://mit.edu/6.01/www/handouts/readings.pdf>.