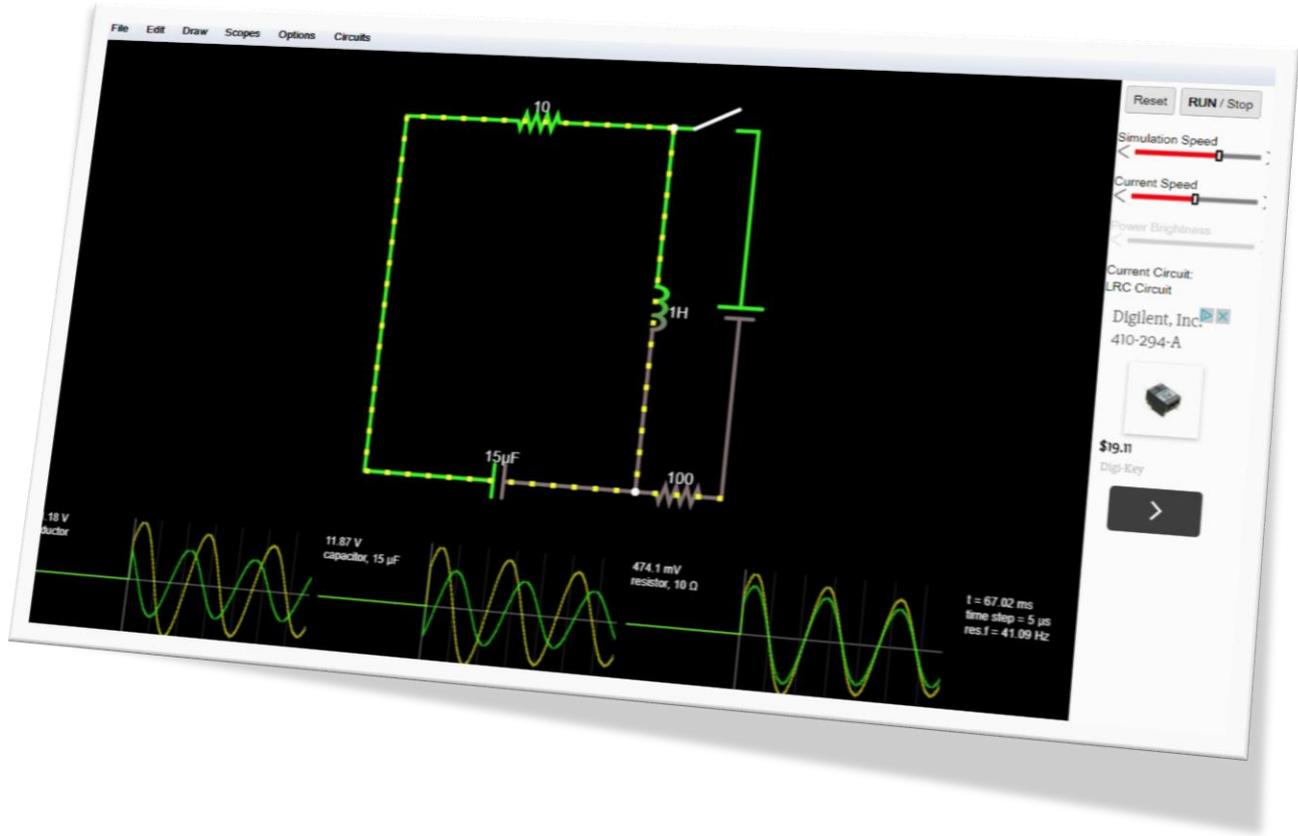


An introduction to the Circuit Simulator Applet

(<http://www.falstad.com/circuit/>)
for use in teaching relays and relay logic



Kevin Berisso, Ph.D.
University of Memphis
Department of Engineering Technology

Winter 2018

Table of Contents

Introduction.....	1
Basic Electronics Knowledge	1
Components	1
Power Sources	2
Switches.....	2
Relays.....	2
Lights (LED and lamps).....	3
Using the Circuit Simulator.....	3
Menus.....	5
Creating your first circuit.....	6
Modifying component settings	11
Power sources	11
Switches.....	11
Relays.....	12
LEDs	13
Troubleshooting	15

Version Control

Version	Date	Description
1.0	18 Jan, 2018	Original version – Relay logic only

Copyright © 2018, Kevin Berisso. All rights reserved.
This manual is available under a Creative Commons license.

Introduction

The Circuit Simulator version 2.1.13js, originally by Paul Falstad, is a free JavaScript based application that works in an enabled Web browser and allows the user to simply and easily simulate various electrical circuits. Providing a level of visual feedback that is ideal for the novice in electronics, the application packs a sufficiently high level of sophistication that an accurate analysis of the designed circuit can be made. Through the available tools and modules within the simulator, the impacts of resistor selection, chip type selection or clock frequency selection can be easily viewed. As an educational tool, the simple fact that the Circuit Simulator is able to visually display what the electrical current is doing can be invaluable when introducing students not only basic electronics and circuit analysis, but to everything from more advanced topics such as relay logic and into some of the more advanced AC topics.

As of this version of the document, only the relay portion of the simulator is covered, as that is the part of the software that is being used in my current class. It is my sincere hope (and plan) to expand this manual out to include segments that cover developing basic DC circuits and beyond as the software is incorporated into more and more classes at the University of Memphis. The format of this manual will be such that a relative novice to electronics should be able to follow along so long as they have a basic understanding of computer usage and more specifically basic Web-based application usage.

Basic Electronics Knowledge

In order to simulate the use of relays, the only electrical understanding that the reader really needs to have are as follows:

- All circuits must be complete, closed and have at least one resistance source for it to do anything
 - While the software will allow for “shorted” circuits, it will warn the user that a circuit exists without a source of resistance
 - A closed circuit has at least one path in which the electricity can flow in a continuous manner from the power source, through a source of resistance and either back to the power source or to ground
 - The software will not allow circuits that are not closed to run correctly unless the break is due to a switch
- All connections between wires must terminate at the point of connection
 - Wires that cross are not considered to be connected
 - Wires are considered to have a negligible level of resistance. If wire resistance is required, a resistor with an appropriately small value can be added

Components

The Circuit Simulator allows the user to arrange a selection of components (switches, chips, power sources, etc.) into a working circuit by connecting the components with wires – much as would be done when physically developing a circuit. While the use of, and description of all of the included components within the software are beyond the scope of this manual at this point, the following are some of the more basic components used in the lower level electronics classes.

Power Sources

There are four basic power sources (see Figure 1); 2-terminal DC, 2-terminal AC, 1-terminal DC and 1-terminal AC. When creating circuits within the simulator, you MUST use one of these four power sources (or one of the more advanced sources) to power your circuit. If using a 1-terminal source, you must also include a ground reference. If, however, you use a 2-terminal source, you can either tie the ground/neutral leg of the power to ground and use a ground reference or you need to close the circuit by having a wire that returns to the ground/neutral side of the power source.

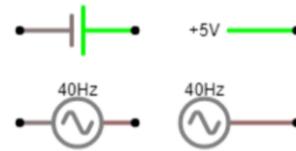
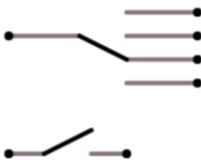


Figure 1. Basic power sources. The right two are 1-terminal sources and the left two are 2-terminal sources.

For DC circuits, the voltage can be changed by either double clicking on the power source or by placing the cursor over the power source and clicking with the right mouse button. More about the available options for power sources will be discussed later. For circuits that require a current limited power source, there is a Current Source option.

Switches



Switches are mechanical devices that allow a user to manipulate the state of an electrical circuit. Think of the light switch in your house. When the switch is in the “off” position, the electrical circuit that allows the lights to turn on is interrupted (broken) and when the switch is in the “on” position, the circuit is completed (made), allowing the lights to turn on.

Figure 2. Single throw (bottom) and multiple throw (top) switches.

Within the Circuit Simulator software there are two types of switches (see Figure 2); single throw switches (listed as “switch” and “push switch”) and multiple throw switches (listed as “SPDT switch”).

For the single throw switches, the “switch” option is a normally closed switch (it starts in the closed or “on” position) and the “push switch” is a normally open switch (it starts in the open or “off” position). Additionally, there is an option that can force the switch to act as a momentary contact switch (more will be discussed later). For the “SPDT switch”, the number of throws (the number of positions that the switch can be moved to) can be manually edited to any (reasonable) number. And while the momentary contact option is listed, it doesn’t work at this point in time (a feature request has been made).

Relays

Relays are electrical or electro-mechanical devices that allow the user to switch the state of a “switch” with voltage instead of through the manual manipulation of the switch. Like manual switches, there are multiple configurations for relays that indicate the number of circuits or “lines” that can be switched. In addition to the switched circuits, there is also an actuator (an electromagnet normally referred to as the coil) which is what does the switching.



Figure 3. SPDT relay (left) and DPDT relay (right)

Physical relays come in multiple formats and types (solid state, reed, latching, etc.), but for simplicity, the Circuit Simulator only supports a single type. Additionally, physical relays have varying control voltage options (5 VDC, 24 VDC, 110 VAC, etc.) and limits to the voltage and current that the contactors are rated for. Like the formats, the simulation ignored these differences for the sake of simplicity, which means that any power source can be used for the simulated relays

In order to drive a relay, voltage needs to be applied to the coil. This in turn allows the relay to move the switch from one throw to the other (see Figure 4). If multiple poles have been configured, ALL poles are thrown at the same time, switching each pole from the default position to the alternate position.

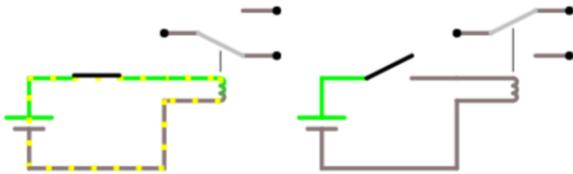


Figure 4. Energized relay (left) and de-energized relay (right)

Most relays (including the ones in the simulator) have two throws, resulting in one throw being normally open (the circuit is open – broken – in the de-energized state) and the other throw being normally closed (the circuit is made – closed – in the de-energized state). Solid state relays are often the exception (in that they are usually Single Pole, Single Throw - SPST). And since the

simulation doesn't directly support SPST relays, you will need to approximate one by just not using the unneeded throw.

Lights (LED and lamps)

One area where the Circuit Simulator has limited support is in outputs. And while it does have output lines, the only two “physical” outputs supported are LEDs and incandescent lamps. Fortunately, it is easy enough for the user to substitute the desired output (e.g. motor, valve, etc.) for either an LED or a lamp in their mind, so this limitation shouldn't be a huge problem. And if the desire is to simulate full blown power distribution solutions, or even full machines, the Circuit Simulator probably isn't what you need anyway.

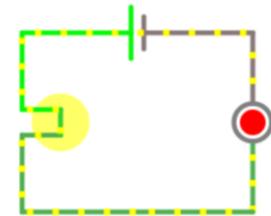


Figure 5. Lamp (left) and LED (right) in a simple

Since both LEDs and lamps are really nothing more than sources of resistance within a circuit, little thought needs to be given as to which is used, and how they go into the circuit (other than the polarity for the LEDs) when dealing with relay logic circuits. In either case, editing the properties of the light will allow for voltage changes, color (LEDs) and other performance controls should they be needed.

Using the Circuit Simulator

To use the software, open up a JavaScript enabled Web browser and navigate to www.falstad.com/circuit/. If you want, you can scroll down the page some and there will be a link to a full screen version, as well as links to some basic instructions and a description of the prefabricated circuits that are available.

There is a default schematic displayed (see Figure 6) when you open the software and you can either erase it all go to *Circuits* → *Blank Circuit* in the menu to clear the simulator.

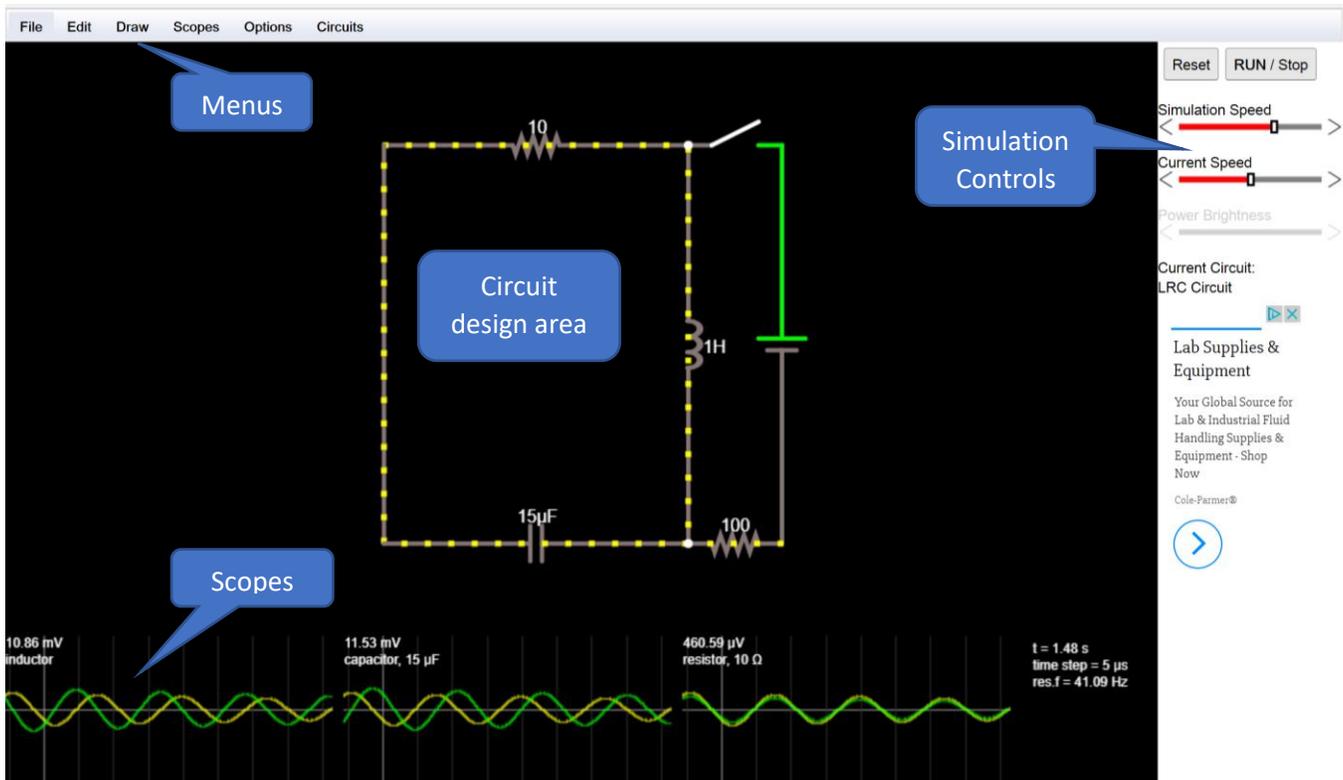


Figure 6. Opening screen for the Circuit Simulation software.

There are four areas on the screen of the software; the menus along the top, the primary area in which the circuit is developed, the scope(s) along the bottom and the right side which has the simulation controls (speed, start/stop, run-time variable component settings, file name). As you move the cursor over components you will see them change to a blue color and a description, along with its current state, will be displayed in the bottom right corner of the design area to the right of the scope(s).

When the system is in run mode, wires that are red negative voltage, green are positive voltage, gray are ground and the moving yellow dots indicate the current flow. It should be noted at this point that the development and modification of the circuit can happen in either state (“on” or “off”). When the Run/Stop button is red (and STOP is capitalized), the simulator is “off” and when it is gray (and RUN is capitalized), the simulator is “on”. Panning around the circuit (when it is larger) can be accomplished via the key and mouse combinations listed in Table 1.

Table 1. Circuit screen navigation

Key stroke(s) and mouse	to get
Alt + left mouse	Pan
Mouse wheel “up”	Zoom out
Mouse wheel “down”	Zoom in

Finally, when a component is hovered over, you can press the delete key or right click on them to delete them. If you want to delete all components, you can press Alt+A and then hit the delete key. To select some (or all) of the components on the screen you can also press and hold the left mouse button and drag a rectangle around the component(s) as shown in Figure 7.

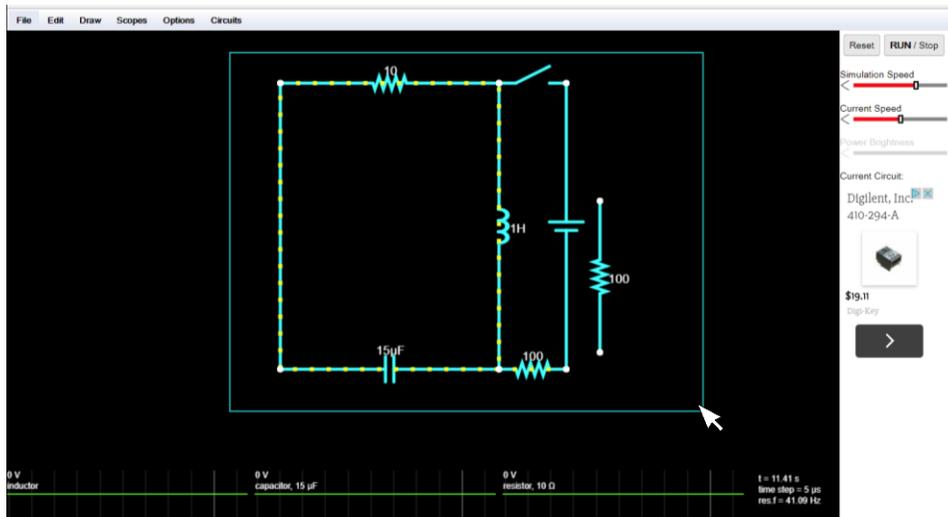


Figure 7. Selecting all the components on a screen.

Menus

The menu structure for the Circuit Simulator follows the traditional Windows logic (as shown below), with ubiquitous “File” and “Edit” menus, along with menus for adding components (“Draw”), configuring the scopes (“Scopes”), configuring the appearance and functionality of the simulation (“Options”) and sample premade circuits (“Circuits”).

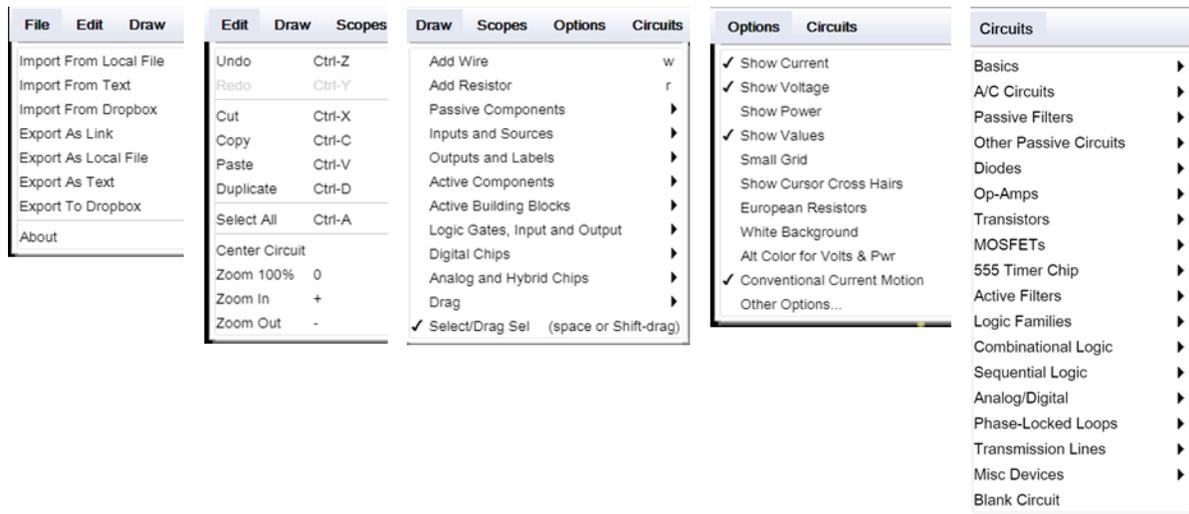


Figure 8. Menu structure.

For the majority of the operations (drawing) that are done within the Circuit Simulator, the user has two options on how to access the menus. Within the primary design area, you can right click and the “Draw” menu will appear as a floating menu, which will show the exact same items as the top menu. Figure 9, shows the “Draw” menu and the three sub-menus (“Passive Components”, “Inputs and Sources” and “Outputs and Labels”) that are needed for creating both basic circuits and specifically relay-related circuits.

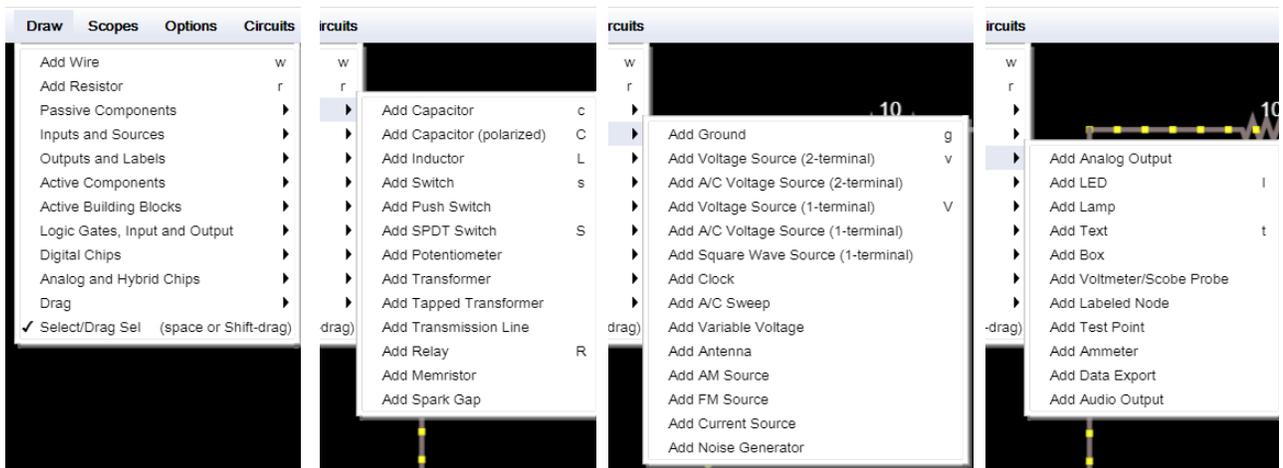


Figure 9. Basic component menu items. Letters next to the components are hotkey shortcuts.

Creating your first circuit

The first relay circuit we are going to generate is a simple single SPDT 24 VDC relay that turns on a 5 VDC light with a SPST push button switch. And while there are multiple ways to represent this circuit, Figure 10, shows one such method which was chosen due to the fact that the circuit has two differing voltages.

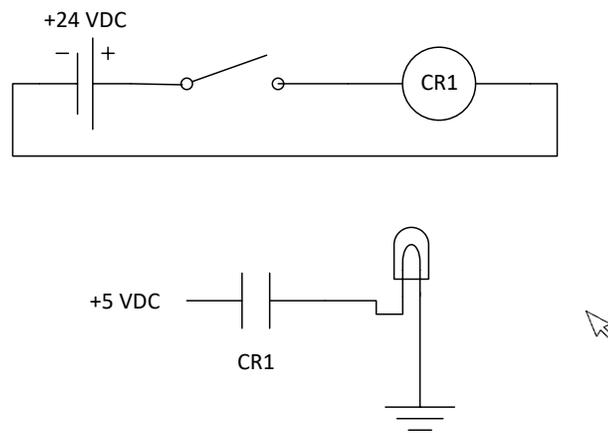


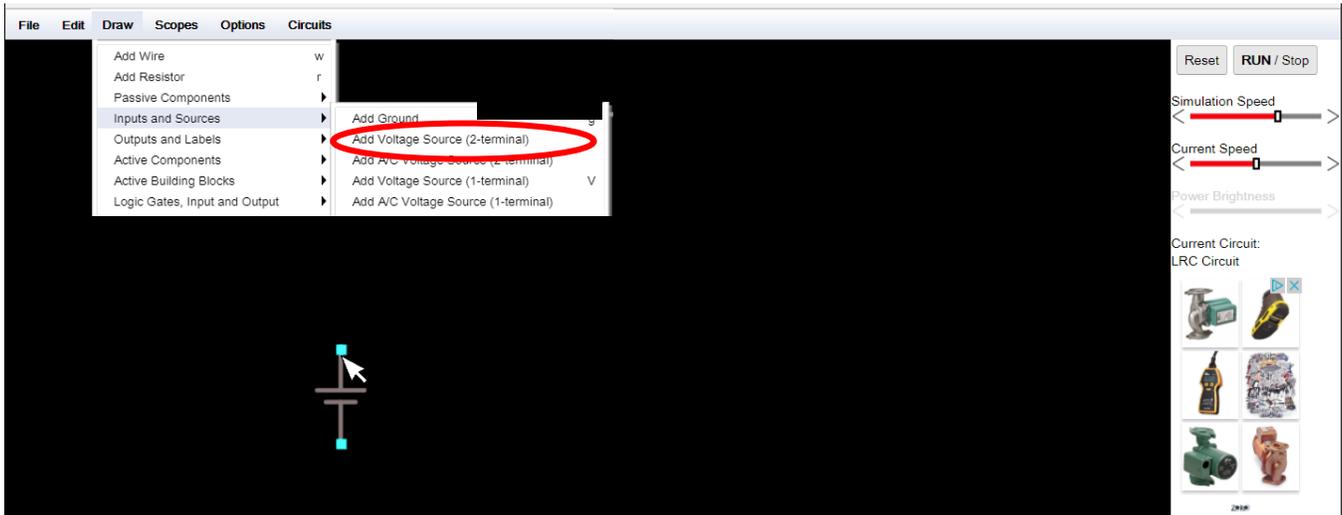
Figure 10. Sample circuit schematic.

The general process in which circuits are developed is as follows:

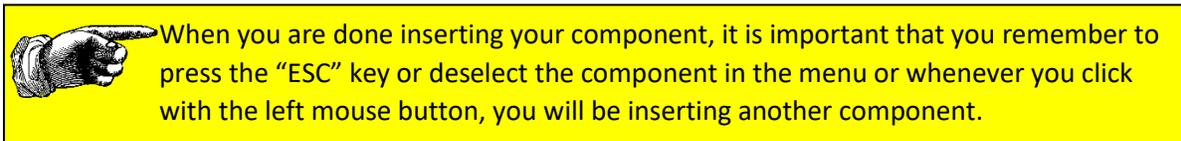
1. Place the necessary power sources in the circuit
2. Add ground references (if needed)
3. Add any switches and outputs
4. Configure the components as needed
5. Connect the individual components together with the necessary wires

The following screen captures show the process in which our circuit is generated. Below the individual pictures are any hints, suggestions or items of note related to the step. It should be noted that screens have been cropped to save on space.

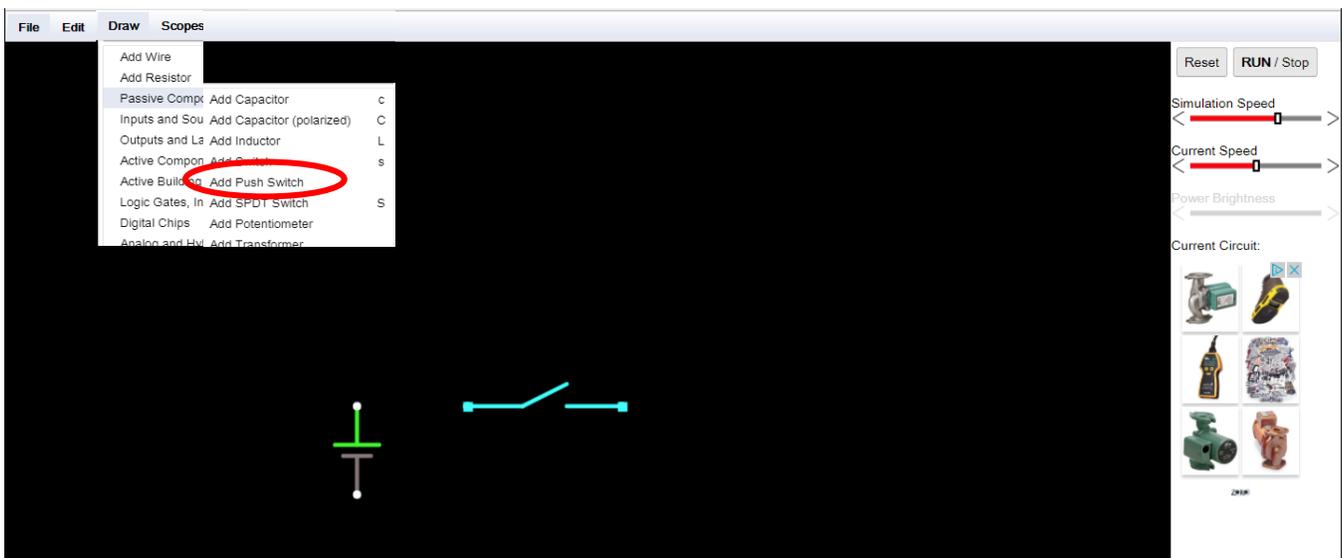
1. Add the power source for the relay portion of the circuit



When adding voltage sources, you will be functionally drawing the source from the minus (negative) side to the positive side. So, to draw our voltage source with the positive terminal on the top, we would click the left mouse button and then drag UP.



2. Add your switch



3. Add the relay

When connecting physical relays, either the relay body or the relay base (if the relay is modular) will have something to indicate how the relay is wired. In the case below (upper left image below), terminals “A1” and “A2” are connected to the coil. Terminals 21 and 11 are the Common connections and 24 and 11 are the normally open contacts while 22 and 12 are the normally closed contacts. As can be seen, 21, 22 and 24 are one single pole, double throw (SPDT) switch and 11, 12 and 14 are the second SPDT switch within the relay. Additionally, the physical terminals on the base (lower left image below) will be numbered. These numbers relate to the schematic numbers on the relay or base.

The same logic holds true for the relays in the software (right image below). The coil will have two connection points as well as the normally open, normally closed and common parts of the switch.

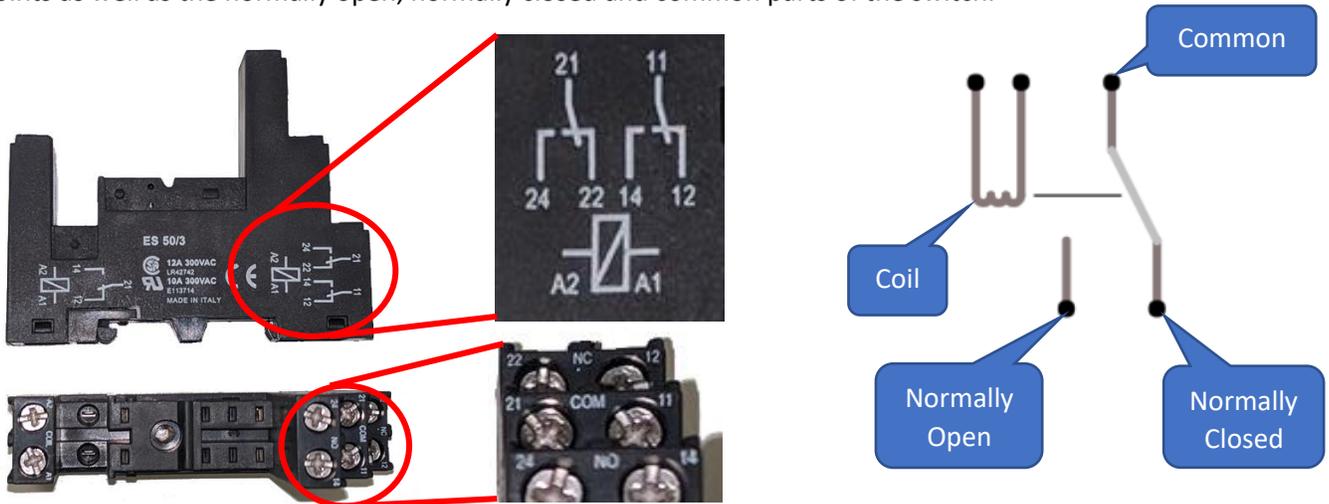
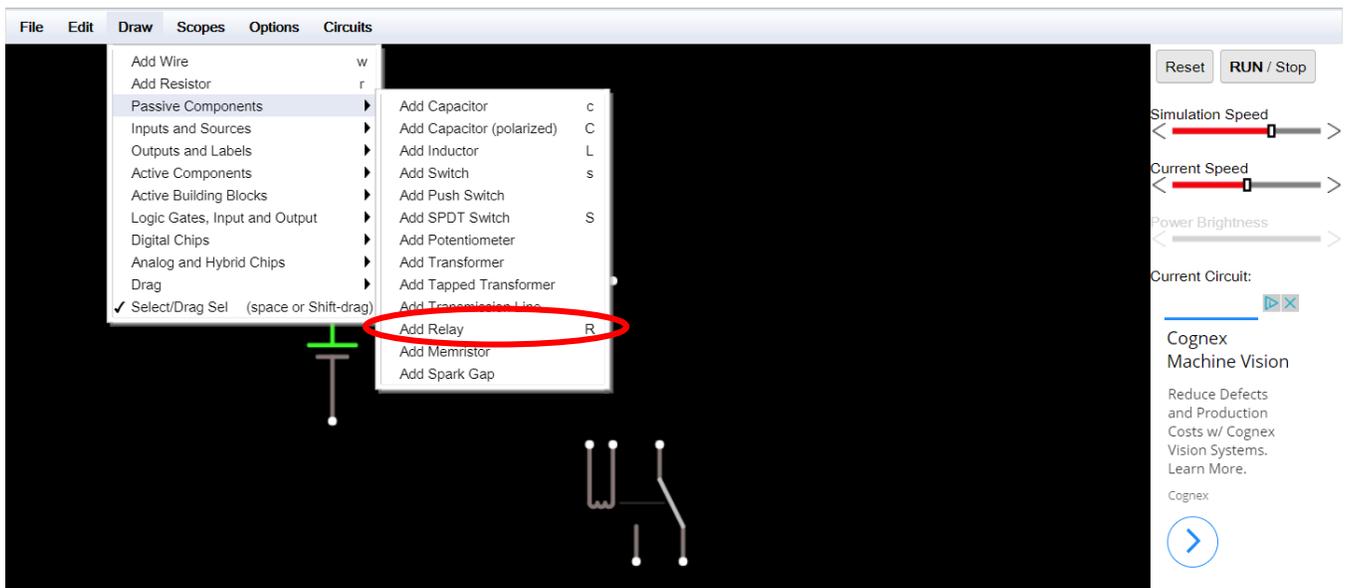
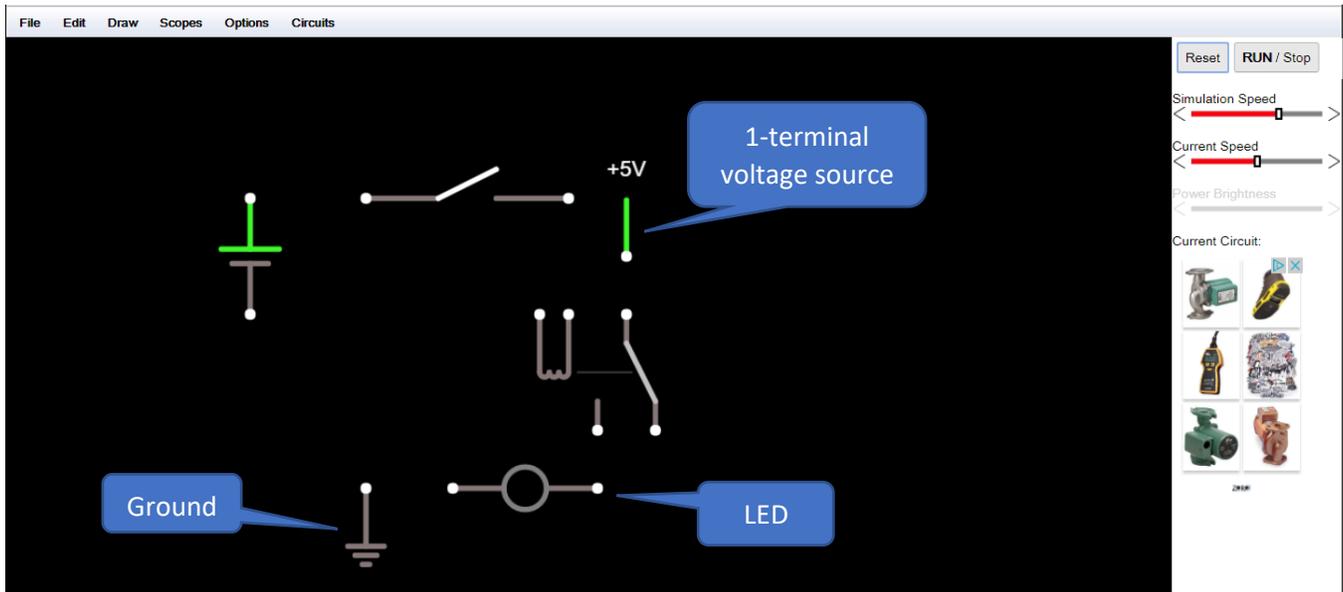


Figure 11. Relay connections.

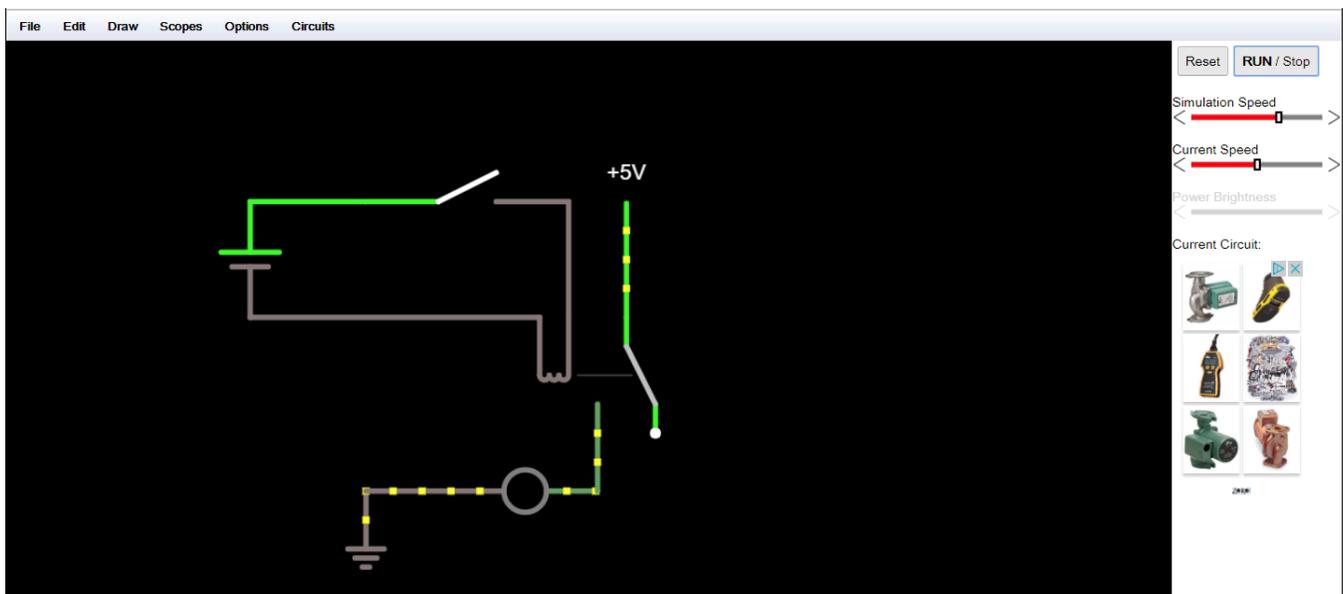


4. Add the remaining components



Remember that because your schematic shows the LED with a 5 VDC power source, that you need to add that power source as a secondary voltage source and that you cannot just tie it into the primary power source.

5. Add wires to connect the components together to create complete circuits.



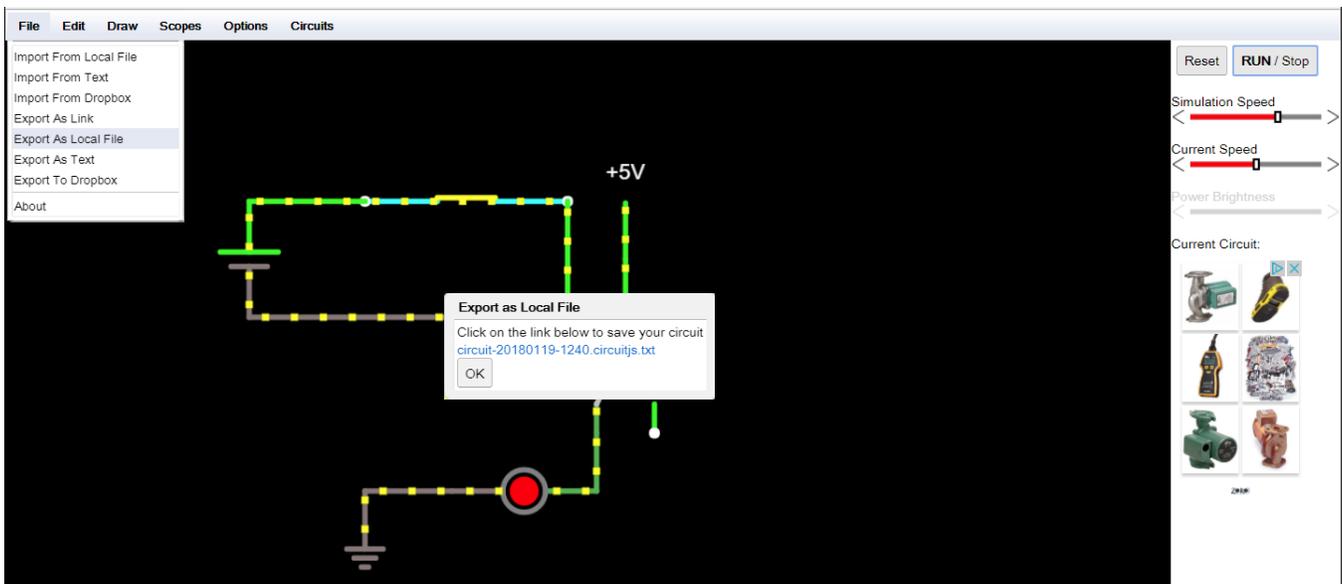
Make sure you are starting and ending the wires on the white dots (nodes) on each component that is being connected. A failure to start and stop on a node will result in a broken circuit.

6. If the system is not in the “on” or RUN mode, click on “Run/Stop” to turn on the simulation. Clicking on the switch will change its state from open to closed.



Changing the state of the switch will cause the LED to turn on and off (the circle will be black or red unless you have changed the LED properties). In addition to the LED lighting up, you will see a current being simulated via the yellow dots moving around the complete(d) circuit(s).

7. To save your circuit, select File → Export as Local File and then click on the link in the resulting dialog box. This will save the file to the Downloads directory on your local computer.

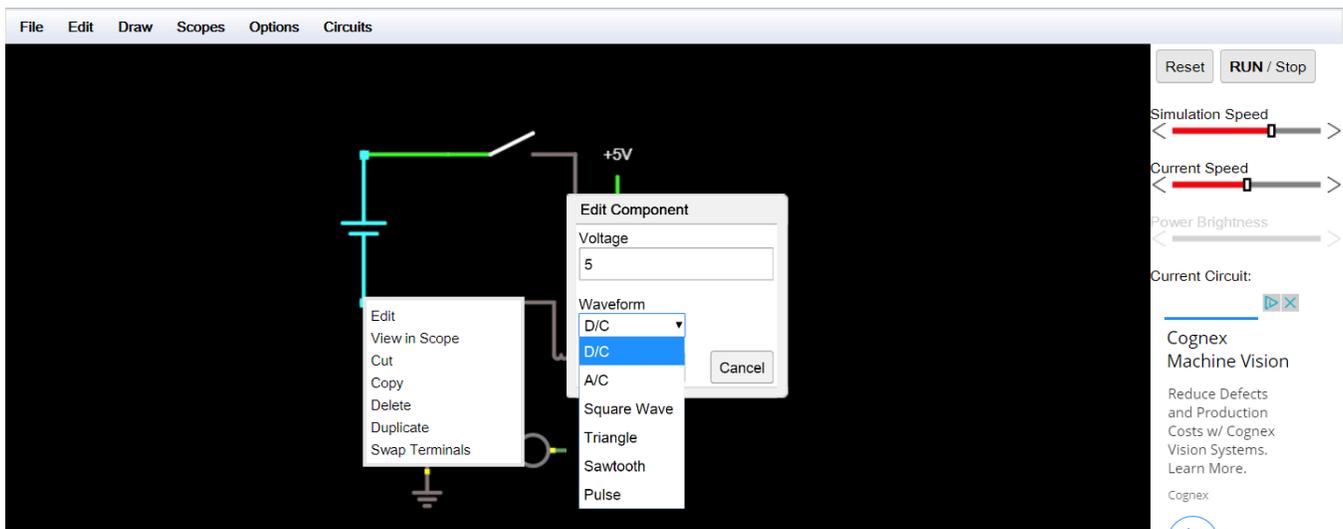


Modifying component settings

While most of the default settings for the components will be sufficient, there are times when you may need to update the settings for the specific circuit being simulated. This is easily accomplished within the software; just place the cursor over the component and click the right mouse button.

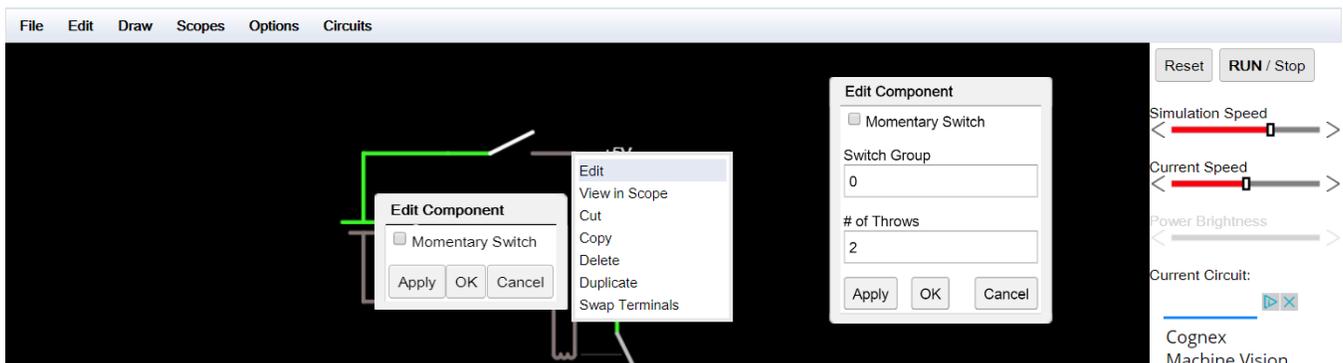
Power sources

In addition to the voltage, power sources can have varying waveforms. It should be noted that the software WILL allow the combining of both AC and DC voltages without any sort of negative results (unlike in the real world where you would be converting a power supply into a door stop).



Switches

As was mentioned earlier on, there are two functional types of switches in the Circuit Simulation software; single throw switches and multiple throw switches. Single throw switches only have a “Momentary Switch” setting (often called “momentary contact” in industrial settings). When changing this setting, whatever state the switch is currently in will be the de-asserted (not pressed) state. This will allow you to be a normally open (N.O.) momentary contact or a normally closed (N.C.) momentary contact. If the “Momentary Switch” setting is not turned on, then the switch does not have a N.O. or N.C. state – it is just considered to have an open or closed state at the time the switch is being discussed.



For multiple throw switches, there are two additional settings; "Group" and "# of Throws". It should also be noted that for multiple throw switches, the "Momentary Switch" function, which is displayed, is currently disabled in the software (although a request has been made to enable it). A multiple throw switch can be thought of as a rotary selector switch (such as showing in Figure 12, or even the ignition of your car). As the switch is moved between settings, each of the throws is connected to in turn (see Figure 13). To increase the number of throws, enter any reasonable number. The software will automatically add the necessary number of addition connection points.



Figure 12. An example of a 3 position (multiple throw) selector switch.

The "Group" setting allows the user to create multi-pole switches (see Figure 13, right image). These are switches where multiple common lines (think "input" lines) are tied together to allow separate circuits to be driven by a single switch. This setting will allow for functionality that is functionally a manually controlled relay.

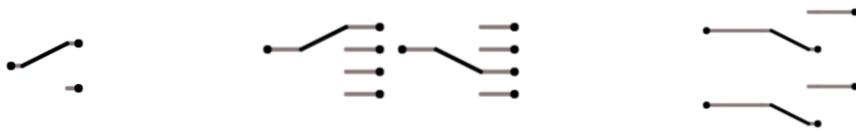
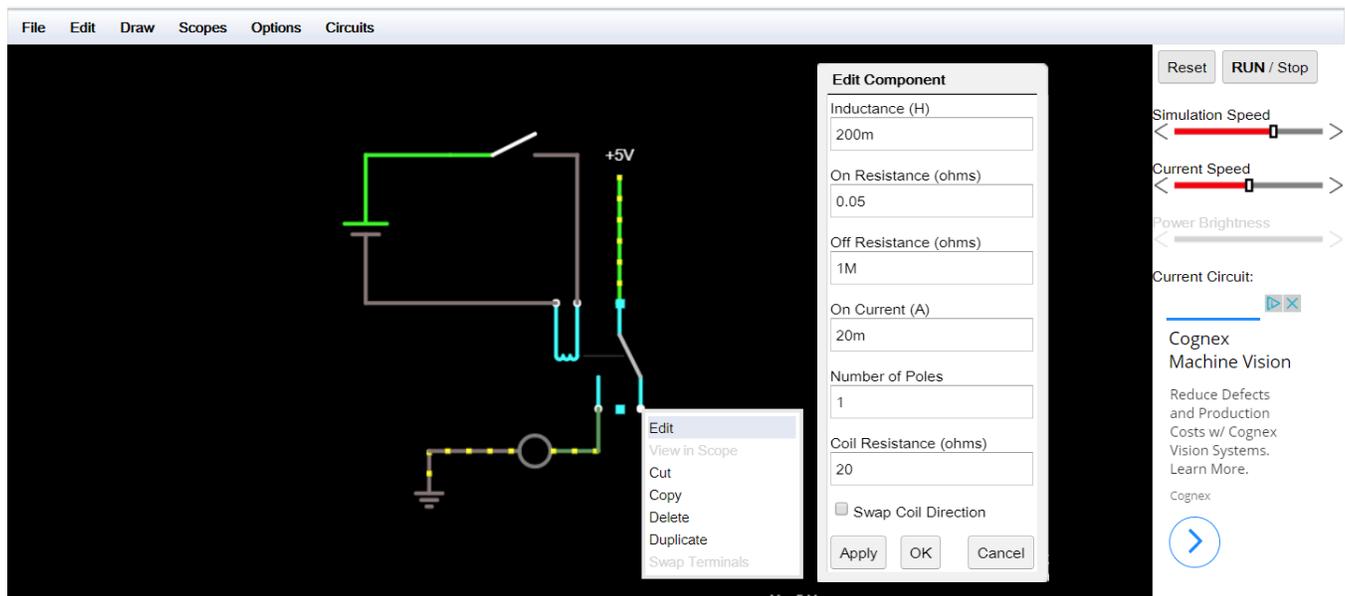


Figure 13. SPDT switch (left), 4 throw switch in multiple positions (center) and a pair of SPDT switches grouped (left)

Groups are defined by any number other than zero (0). Whenever a one switch within a group is changed, ALL of the switches within the group are changed.

Relays

Since the primary goal of the Circuit Simulation software is to simulate and analyze circuits, there are several settings in the relay component that are not needed if simple relay logic is being addressed. For the purposes of this version of the manual, only the number of "Poles" and the "Swap Coil Direction" options will be discussed (future versions of this manual will address the other settings as needed).



The number of poles (switched circuits) that relay will support is set with the “Number of Poles” setting. And while the software will allow larger numbers (greater than 100), it should be realized that a) you probably couldn’t find a physical relay with that many poles and b) the software will quickly come to a halt as it tries to process that many poles. Most conventional relays max out at between four and six poles, with two and three being at the higher end of easily purchased relays. Should a large number of poles or addition throws (beyond two) be needed for a circuit, normal convention dictates the combining of multiple “simpler” relays into a larger complex relay “block”.

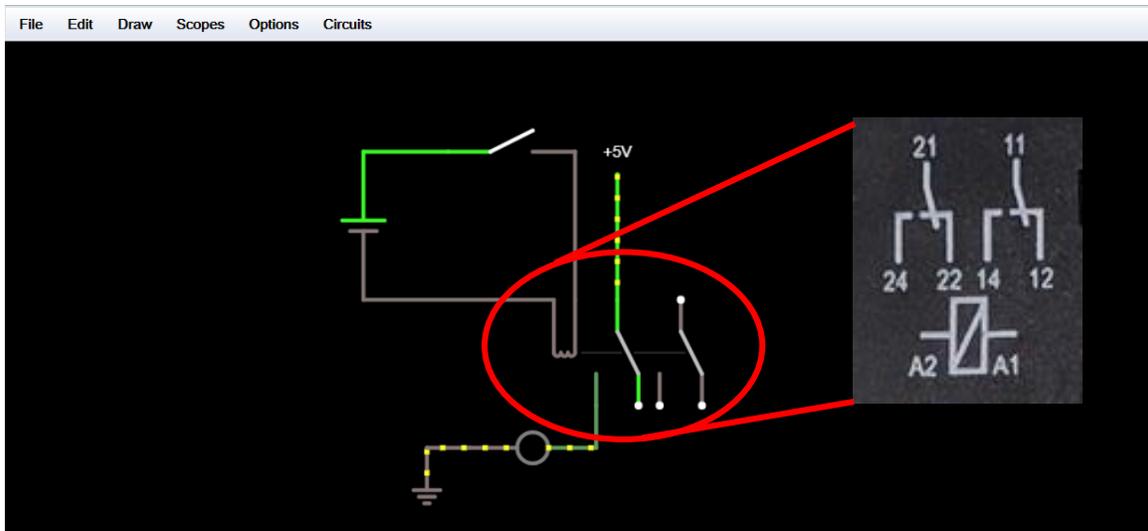


Figure 14. Example of a DPDT relay like what was shown in Figure 11.

LEDs

The LED settings, much like most of the settings for the relays, is outside of the scope of this version of the manual. The relevant settings for now are the colors. The three colors listed allow for the generation of pretty much any Red/Green/Blue (RGB) color combination desired. The table below shows some of the common color combinations available.

Table 2. A sampling of RGB Color combinations. Derived from https://www.rapidtables.com/web/color/RGB_Color.html¹

Red	Green	Blue	Result
0%	0%	0%	Black
0%	0%	100%	Blue
0%	100%	100%	Cyan/Aqua
50%	50%	50%	Gray
0%	50%	0%	Green
0%	100%	0%	Lime Green
100%	0%	100%	Magenta
100%	0%	0%	Red
75%	75%	75%	Silver
100%	100%	100%	White
100%	100%	0%	Yellow

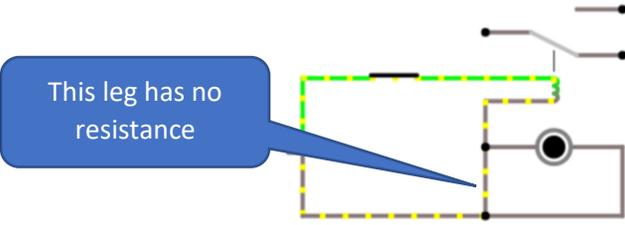
One thing that may have to be addressed in some of the circuits will be a current limiting resistor. If you receive a “max current exceeded” in your circuit, try adding a 100 Ω resistor. To add the resistor, delete the wire after the LED and go to *Draw*→*Add Resistor* and add that into the circuit in place of the plain wire. Right click on the resistor and change the “Resistance” to 100 (see Figure 15).



Figure 15. Circuit without (left) a resistor and one with (right) a resistor.

¹ The author has no affiliation with this webpage. It was picked for the simple reason that it was the first one to show the names of mixed colors and to have an interactive display of the resulting colors for the search “rgb color”.

Troubleshooting

<i>Problem</i>	<i>Issue(s)</i>
Voltage source loop with no resistance	<ul style="list-style-type: none"> One or more of your circuits are shorted (have a path from the positive terminal of the power source that doesn't go through any sources of resistance – e.g. resistor, LED, coil, etc.). <p>Add a source of resistance to your circuit</p>
Max current exceeded	<ul style="list-style-type: none"> Your circuit goes through a LED with without a current-limiting resistor. <p>Add a 100 Ω resistor to your circuit</p>
X bad connection(s)	<ul style="list-style-type: none"> One (or more) component nodes do not connect to your circuit. <p>Make sure each wire starts and stops at each node to be used in your circuit. Having wires cross a node is not sufficient.</p>
One or more parts of your circuit do not have current (yellow dots) flowing through it	<ul style="list-style-type: none"> One (or more) component nodes do not connect to your circuit. <p>Make sure each wire starts and stops at each node to be used in your circuit. Having wires cross a node is not sufficient.</p> <ul style="list-style-type: none"> You have created a parallel circuit that has one leg which is functionally a short (or has such a low resistance) resulting in no current flowing to the second leg <div style="text-align: center;">  <p>The diagram shows a parallel circuit with two legs. The top leg contains a resistor and an LED. The bottom leg is a direct short circuit. A blue callout box points to the bottom leg with the text "This leg has no resistance". Yellow dots on the wires indicate no current flow through the shorted leg.</p> </div> <p>Alter your circuit so that the zero resistance leg has some resistance.</p>