

# MODULAR ELECTRONICS LEARNING (MODEL) PROJECT



## PRINTED CIRCUIT BOARDS

© 2022-2025 BY TONY R. KUPHALDT – UNDER THE TERMS AND CONDITIONS OF THE  
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL PUBLIC LICENSE

LAST UPDATE = 13 APRIL 2025

This is a copyrighted work, but licensed under the Creative Commons Attribution 4.0 International Public License. A copy of this license is found in the last Appendix of this document. Alternatively, you may visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons: 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Recommendations for students . . . . .	3
1.2	Challenging concepts related to printed circuit boards . . . . .	5
1.3	Recommendations for instructors . . . . .	6
<b>2</b>	<b>Case Tutorial</b>	<b>7</b>
2.1	Diagnostic visualization tools . . . . .	8
<b>3</b>	<b>Tutorial</b>	<b>13</b>
3.1	Introduction to PCBs . . . . .	14
3.2	PCB construction . . . . .	16
3.3	Layout tools . . . . .	19
3.4	Basic layout elements . . . . .	21
3.5	Thermal design principles . . . . .	26
3.6	Preserving signal integrity . . . . .	27
3.7	High-speed design principles . . . . .	29
3.8	KiCad design example . . . . .	32
3.8.1	Schematic capture using KiCad . . . . .	33
3.8.2	PCB editing using KiCad . . . . .	36
3.8.3	Design rule checking using KiCad . . . . .	41
3.8.4	Plotting files using KiCad . . . . .	45
<b>4</b>	<b>Derivations and Technical References</b>	<b>47</b>
4.1	Trace ampacity ratings . . . . .	48
4.2	Trace voltage ratings . . . . .	49
4.3	Tony's PCB layout recommendations . . . . .	50
4.4	Tony's KiCad recommendations . . . . .	51
4.4.1	Preliminary setup recommendations . . . . .	51
4.4.2	Finding components within KiCad's libraries . . . . .	52
4.4.3	Installing new schematic symbols and layout footprints . . . . .	53
4.5	Electric field quantities . . . . .	57
4.6	Magnetic field quantities . . . . .	59
4.7	Near-field versus far-field regions . . . . .	69

CONTENTS	1
<b>5 Questions</b>	<b>73</b>
5.1 Conceptual reasoning . . . . .	77
5.1.1 Reading outline and reflections . . . . .	78
5.1.2 Foundational concepts . . . . .	79
5.1.3 Applying foundational concepts to a series-parallel circuit . . . . .	81
5.1.4 Parasitic inductance on a PCB . . . . .	82
5.1.5 Heat sink for soldering . . . . .	83
5.1.6 Impedance of lines on a printed circuit board . . . . .	84
5.2 Quantitative reasoning . . . . .	85
5.2.1 Miscellaneous physical constants . . . . .	86
5.2.2 Introduction to spreadsheets . . . . .	87
5.2.3 Current measurement on a PCB . . . . .	90
5.3 Diagnostic reasoning . . . . .	92
5.3.1 Sloppy schematic capture . . . . .	93
5.3.2 Faults in a PCB circuit . . . . .	94
<b>A Problem-Solving Strategies</b>	<b>95</b>
<b>B Instructional philosophy</b>	<b>97</b>
<b>C Tools used</b>	<b>103</b>
<b>D Creative Commons License</b>	<b>107</b>
<b>E References</b>	<b>115</b>
<b>F Version history</b>	<b>117</b>
<b>Index</b>	<b>118</b>



# Chapter 1

## Introduction

### 1.1 Recommendations for students

Printed circuit boards, or PCBs, are *the standard* method of interconnecting electronic components in permanent assemblies. This module explores basic PCB construction and design.

Important concepts related to PCBs include **traces**, **layers**, **copper pours**, **soldermask**, **silkscreen**, component **footprints**, **ratsnest** lines, **through-hole** versus **surface-mount** components, **design-rule checking**, **clearances**, dimensions in **mils** versus **millimeters**, **Gerber files**, **nets**, **vias** versus **pads**, **thermal reliefs**, **crosstalk**, **parasitic** capacitance and inductance, **ampacity**, and **electric fields** versus **magnetic fields**.

Here are some good questions to ask of yourself while studying this subject:

- Summarize the steps you would take to design a PCB layout using modern software.
- What factor(s) dictates the minimum width of a copper trace?
- What factor(s) dictates the minimum spacing between adjacent copper traces?
- What material(s) are PCBs typically made from?
- How do through-hole components differ from surface-mount components?
- What are *standing waves*, and why must we consider them for some PCB layouts?
- Identify some general strategies for minimizing the amount of undesired “signal coupling” between traces on a PCB.
- How may electric fields be minimized on a PCB?
- How may magnetic fields be minimized on a PCB?
- What does it mean to *route* or *track* during the design of a PCB?
- What purpose do *vias* serve, and how do they differ from through-hole *pads*?

- What are some of the common units of measurement used for physical dimensions on a PCB?
- How does a *ratsnest* aid in the layout of a PCB?
- What purpose does a component *footprint* serve when designing a PCB?
- Where would we find dimensional information for a component when creating a custom footprint?
- What purpose does the *soldermask* layer of a PCB serve?
- What purpose does the *silkscreen* layer of a PCB serve?
- What purpose does *conformal coating* serve on a populated PCB?
- What are some different ways components may be soldered to a PCB?
- What is *through-hole plating* on a PCB?
- What does it mean to *tin* a copper trace or a copper wire?
- How do the *schematic capture* and *board layout* steps of the PCB design process differ?
- What are some of the design parameters checked during an automated *DRC*?
- Why might you choose to use a copper *fill* or *pour* rather than a normal-width copper trace in a PCB design?
- What is a copper *plane* on a PCB design?
- What is a *thermal relief* on a PCB design, and why would we wish to utilize one?
- What are some ways in which a high-speed circuit PCB layout differs from a DC or low-frequency AC circuit layout?
- What does the phrase *via-stitching* mean?
- Why is it recommended to space vias no farther apart than  $\frac{\lambda}{20}$  when “stitching” copper planes together on a high-speed design when  $\frac{\lambda}{2}$  is actually the critical distance?
- What do the terms *victim* and *aggressor* mean with reference to circuit performance?

## 1.2 Challenging concepts related to printed circuit boards

The following list cites concepts related to this module's topic that are easily misunderstood, along with suggestions for properly understanding them:

- **Vias versus through-hole pads** – both vias and pads for through-hole devices have copper connecting multiple layers of a PCB, but while through-hole pads are designed to solder to through-hole component leads vias are instead *only* designed to connect copper traces or planes together on different layers. Vias have no exposed copper to solder anything to, and in fact the soldermask layer completely covers vias while it does not cover the pads of a through-hole device.



### 1.3 Recommendations for instructors

This section lists realistic student learning outcomes supported by the content of the module as well as suggested means of assessing (measuring) student learning. The outcomes state what learners should be able to do, and the assessments are specific challenges to prove students have learned.

- **Outcome** – Demonstrate effective technical reading and writing

Assessment – Students present their outlines of this module’s instructional chapters (e.g. Case Tutorial, Tutorial, Historical References, etc.) ideally as an entry to a larger Journal document chronicling their learning. These outlines should exhibit good-faith effort at summarizing major concepts explained in the text.

- **Outcome** – Apply foundational electrical concepts to circuits built on a PCB

Assessment – Qualitatively determine voltages and currents for PCB-mounted components based on their series-parallel interconnections; e.g. pose problems in the form of the “Applying foundational concepts to a series-parallel circuit” Conceptual Reasoning question.

Assessment – Identify effects of faults in a circuit built on a PCB; e.g. pose problems in the form of the “Faults in a PCB circuit” Diagnostic Reasoning question.

- **Outcome** – Apply concepts of electrical diagnosis to PCB circuits where components cannot be removed with ease

Assessment – Infer component currents based on voltage measurements and known resistance values; e.g. pose problems in the form of the “Current measurement on a PCB” Quantitative Reasoning question.

- **Outcome** – Independent research

Assessment – Read and summarize in your own words reliable source documents on the subject of printed circuit board design. Recommended readings include any relevant standards documents published by IPC, including *IPC-2221* and *IPC-2152*.

Assessment – Locate datasheets for thermal bridging components such as Vishay’s ThermoWick and from those datasheets identify such parameters as physical dimensions, soldering recommendations, and thermal resistance values.

## Chapter 2

# Case Tutorial

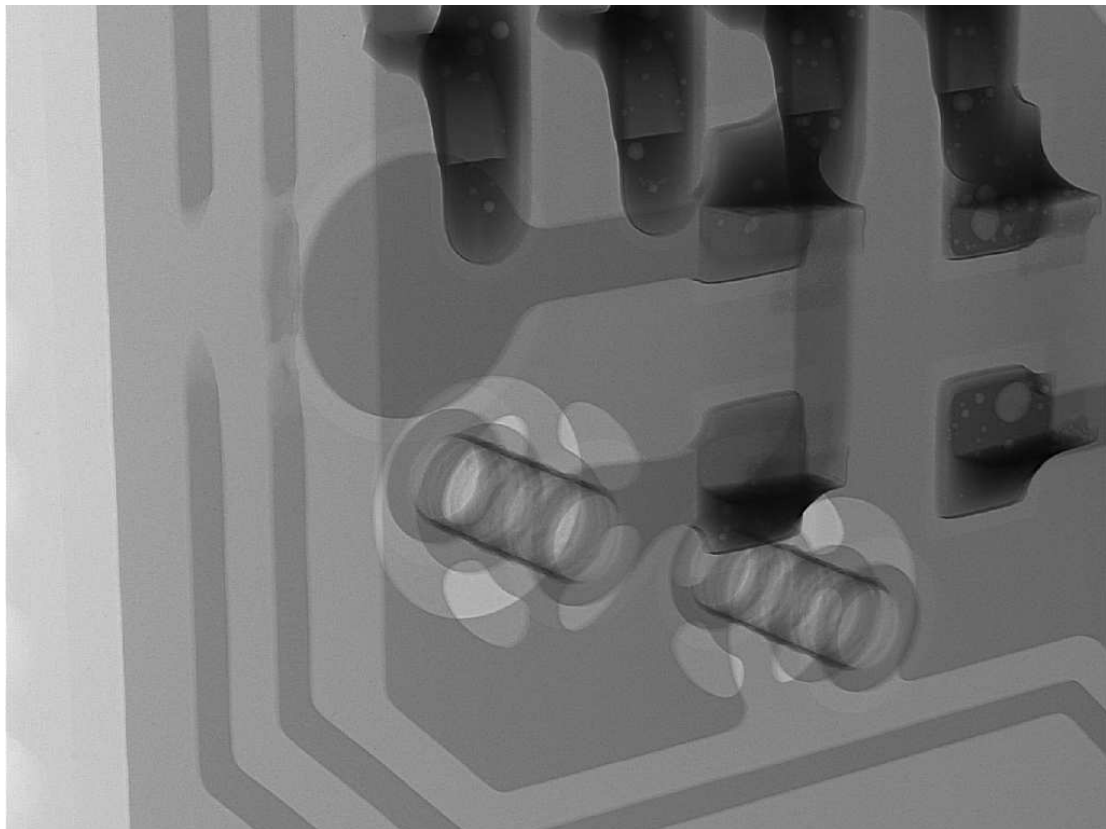
The idea behind a *Case Tutorial* is to explore new concepts by way of example. In this chapter you will read less presentation of theory compared to other Tutorial chapters, but by close observation and comparison of the given examples be able to discern patterns and principles much the same way as a scientific experimenter. Hopefully you will find these cases illuminating, and a good supplement to text-based tutorials.

These examples also serve well as challenges following your reading of the other Tutorial(s) in this module – can you explain *why* the circuits behave as they do?

## 2.1 Diagnostic visualization tools

The following photographs<sup>1</sup> show examples of diagnostic visualization tools such as infra-red thermography and X-ray photography applied to the detection of faults on printed circuit boards (PCBs) and within integrated circuits (ICs).

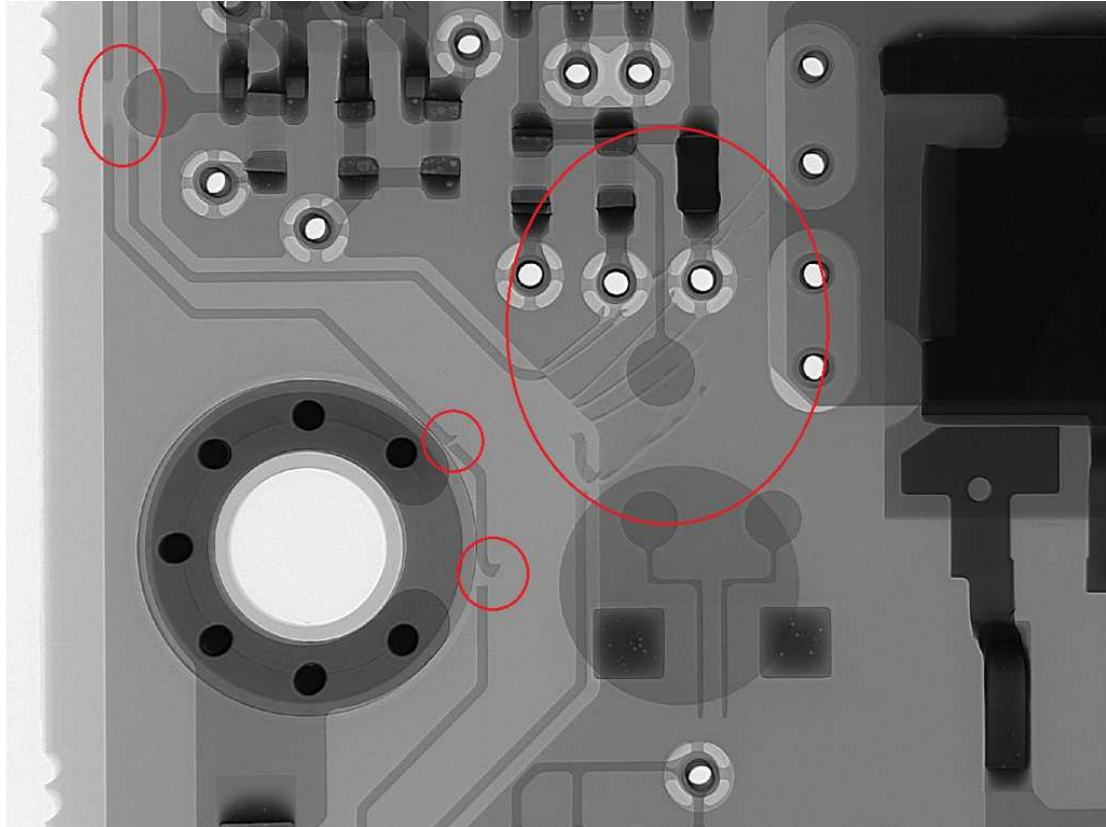
Here we see an X-ray photograph of a PCB showing two broken traces resulting from defective board manufacturing where portions of the two traces were mistakenly etched away:



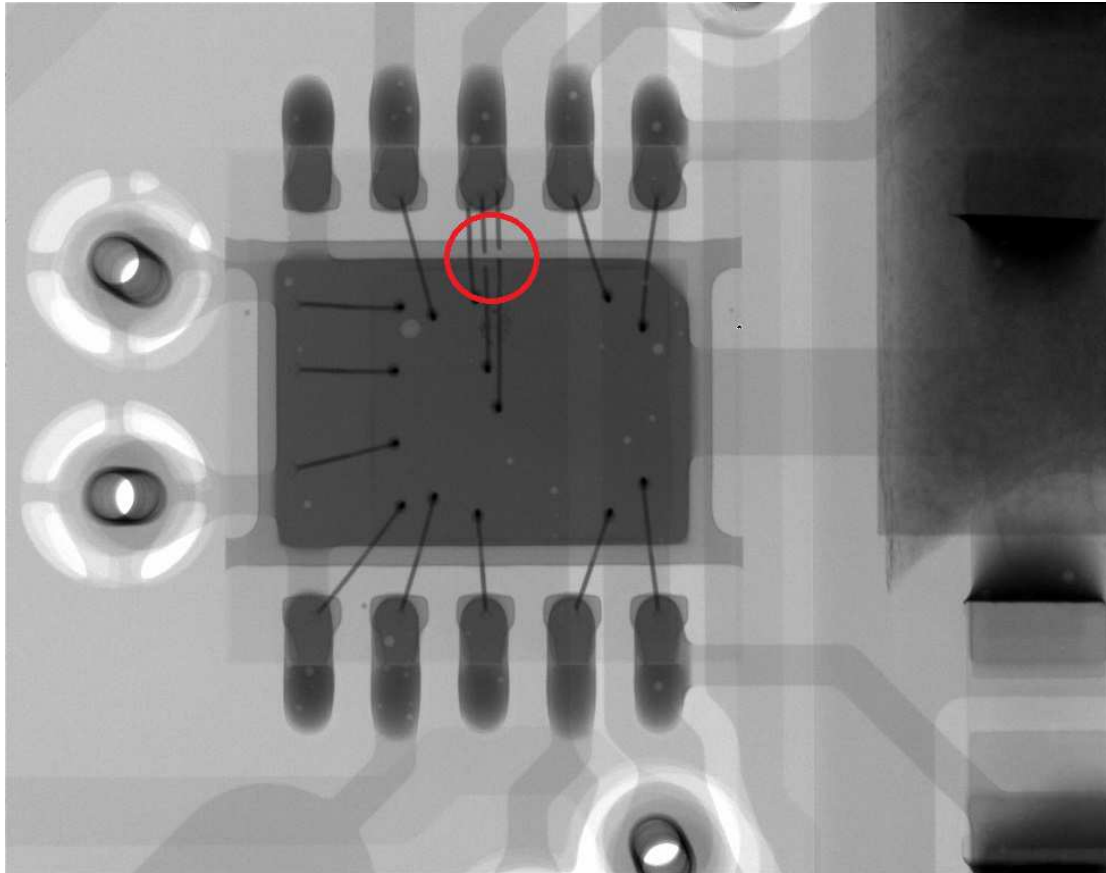
---

<sup>1</sup>I am indebted to the work of Brian Sharpes at Schweitzer Engineering Laboratories (SEL) for these images which he collected over a span of fifteen years.

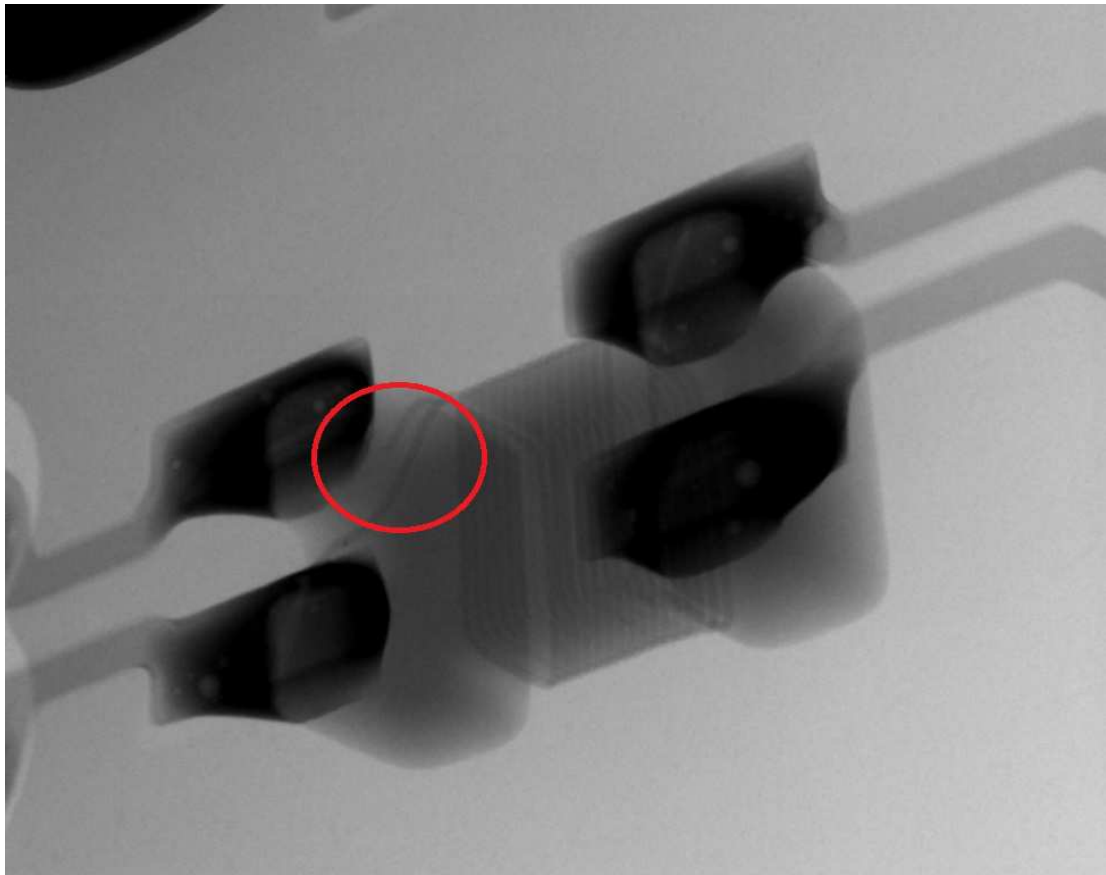
A wider view of the same PCB shows trace failures cause by physical damage to the board in addition to those traces mal-formed at the time of manufacture:



This X-ray photograph shows broken lead wires *inside* of an IC between one of the external terminals and two different points on the IC's internal silicon die:



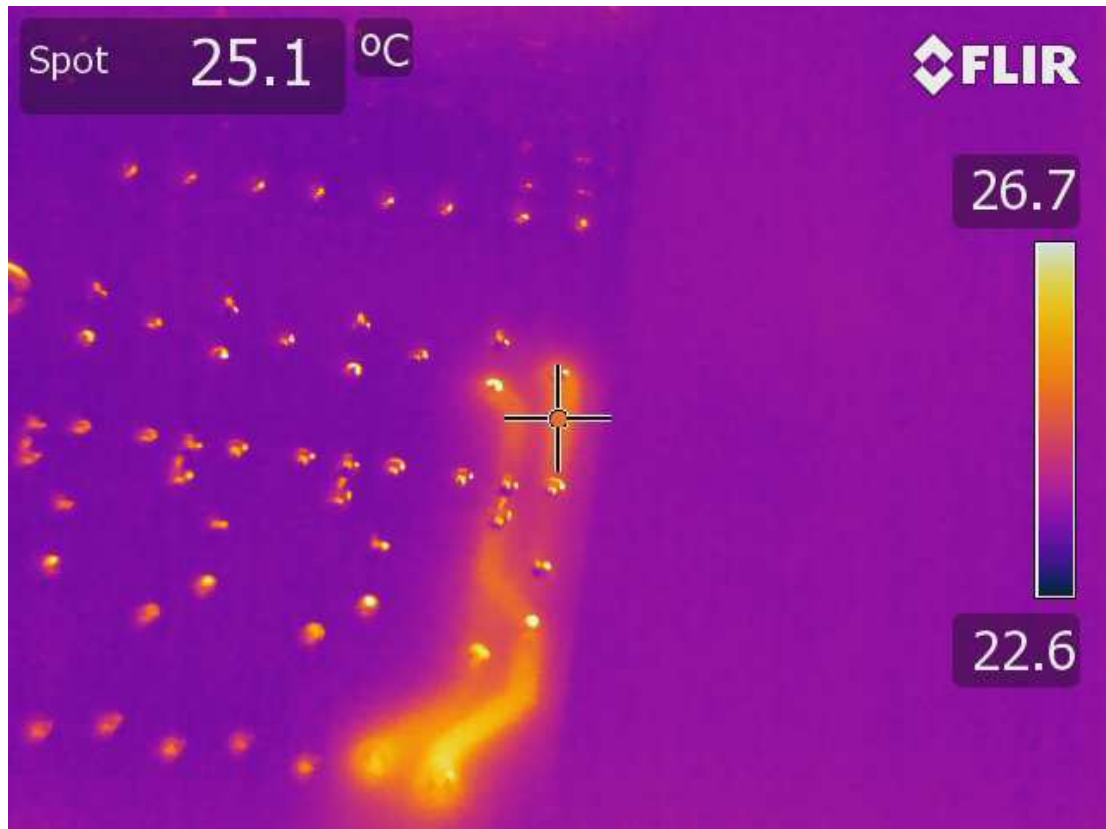
Another example of an internal component fault revealed by X-ray photograph is in this image of a *common-mode balun*<sup>2</sup> soldered to a PCB. These devices consist of two closely-coupled inductors (i.e. a 1:1 transformer) designed to present a high impedance to common-mode currents along a transmission line while presenting negligible impedance to differential currents along the same line. Packaged as a surface-mount device (SMD), we see one broken lead wire at its left end not connecting with the upper-left external terminal as it should:



---

<sup>2</sup>Also known as a *current balun*. The term “balun” refers to Balanced-Unbalanced, referring to a common application of this device: namely, to convert a balanced signal into an unbalanced signal or vice-versa.

In the following infra-red image we see two overheated traces on a powered PCB, clearly distinguished from the others by their brighter coloring:



## Chapter 3

## Tutorial

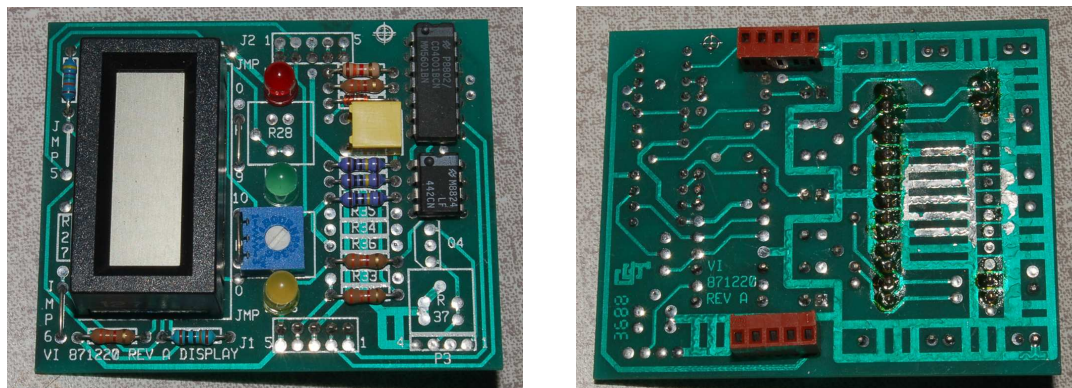


### 3.1 Introduction to PCBs

Electronic circuits consist of multiple components joined together to form complex networks. Many ways exist to join these components together, but one of the most common is to use a structure known as a *printed circuit board* or *PCB*. This is a flat board made of some electrically-insulating material such as fiberglass, onto which strips of conductive metal such as copper are bonded. These metal *traces* (also called *tracks*) form conductive pathways similar to wires, serving to connect the terminals of various components together. Electrical bonds between component terminals and the metal traces of a PCB are generally formed by *soldering*.

Printed circuit board technology lends itself very well to the mass-production of electronic circuit assemblies, as these boards may be cut, drilled, etched, and overlaid by automated processes, with any number of boards readily manufactured to precise dimensional, mechanical, and electrical specifications.

In these photographs we see the front and back sides of a PCB populated with *through-hole* components, where the metal leads of the component penetrate pre-placed holes in the PCB, each hole ringed by a copper *pad* (also called a *land*) where solder is applied to join the component lead to the pad. Through-hole components, sometimes referred to by the abbreviation *THT* (for Through-Hole Technology) are the type commonly used by students and hobbyists to construct prototype and test circuits because they are easy to form connections between, commonly with a tool known as a *solderless breadboard*. For many years through-hole components were exclusively used for commercial PCB-based electronic products.

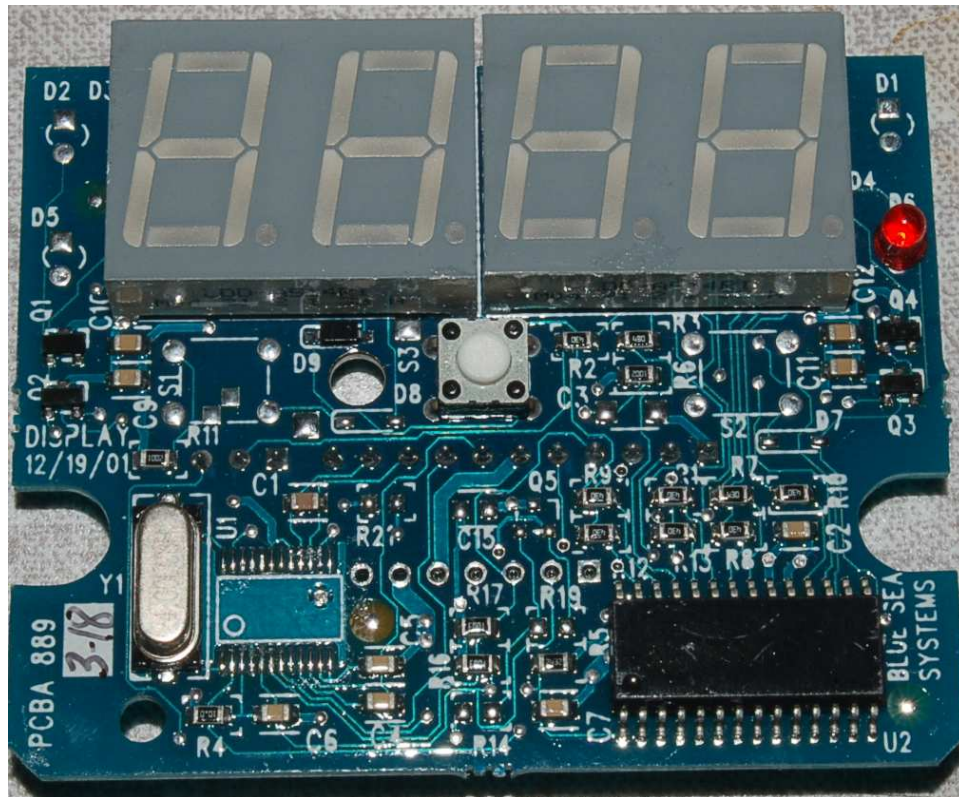


It is easy to identify the green-colored copper traces on both front and back sides of this PCB, as well as the pads where component terminals are soldered to these traces. Additionally, we see writing on both sides of the PCB, the front-side writing applied with white-colored *silkscreen* printing and the back-side writing actually applied as bits of copper.

Soldering of mass-produced through-hole PCBs typically uses the *wave* soldering process, where boards carried along a conveyor are dipped into vats of molten solder, just far enough in so that the liquid solder “wicks” up from the bottom surface of the board, through the holes, to join component leads to pads.

As electronic circuits grew more complex, with larger and larger component counts, there was a need to pack more circuitry into smaller PCBs, and with this need came the invention of *surface-mount* components. These are also known as surface-mount devices, or *SMDs*. In contrast to through-hole components with their long wire leads, SMDs have metal terminals designed to be soldered on to hole-less copper pads on printed circuit boards.

This next photograph shows a PCB populated mostly with surface-mount devices, and only a few through-hole devices. The size of this square PCB is approximately 5 centimeters on each edge:



Here as well it is easy to identify the traces serving as wires connecting various components together on the board. Some of these traces terminate at round- or square-shaped pads where through-hole components are soldered in, while most of the traces terminate at tiny rectangular pads onto which surface-mount components are soldered.

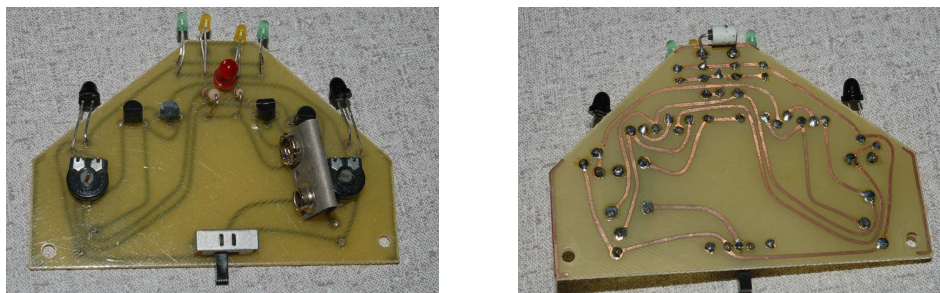
Soldering of mass-produced SMD boards typically use the *reflow* soldering process, where a special paste consisting of a mixture of flux and powdered solder is first applied to all pad areas on the PCB, components placed onto the pasted pads, and then the board passed through an oven where the heat causes the flux to activate and the powdered solder to melt and form bonds between those pads and the component terminals. Precise application of the solder paste to every pad on the PCB is crucial to good reflow solder quality, and so metal *stencils* with cut-out holes exactly matching all the pad placements on the PCB are used to direct the solder paste prior to component

placement. Interestingly, the surface tension of the molten solder paste typically works to pull each component into alignment with its respective pads on the PCB, an important phenomenon because SMD components lack the direct mechanical alignment through-hole components enjoy with the latter's leads being inserted through holes in the board.

## 3.2 PCB construction

At minimum a printed circuit board consists of a thin electrically-insulating board typically made of a fiberglass-resin composite material (*FR4* being the most common material at the time of this writing) with strips of metal bonded to at least one of its sides to form connecting “traces” between component terminals. For through-hole components, holes drilled in the fiberglass board allow the component leads to penetrate as well as to bond with “pads” connecting the lead and the trace by solder.

Such a minimalist PCB is shown in the following photographs (showing front and back sides), using *through-hole* components where the component wire leads are pushed through pre-drilled holes aligned with the copper traces. Solder applied to each of these protruding leads bonds it to the respective copper trace, providing both an electrical connection to the trace and a mechanical connection to the board:

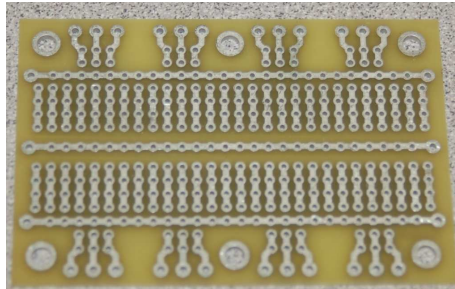


This particular PCB is called *single-sided* because traces exist only on one of its sides. The free-form layout of these metal traces indicates this particular PCB was made by hand rather than by machine, a human being drawing the placement of each trace with a pen. These traces are actually “laid out” in a subtractive manner, the board beginning with a continuous sheet of copper on its back side, and an acid-etching process used to etch away all copper *except* for those traces previously marked. In all likelihood the through-hole component holes on this particular PCB were also drilled by hand!

Modern PCB design dispenses with these manual construction techniques in favor of computer-controlled etching and machining processes. Instead of drawing trace pathways by hand, specialized computer software allows one to sketch the pattern of traces, pads, and holes. The resulting digital file is then sent to a *board shop* where machines convert those instructions into mass-produced PCBs. Such mass production is available even to hobbyists now, with board shops around the globe taking digital orders for custom PCBs, shipping the finished boards to any destination with mail service!



The following photograph shows a PCB designed using PCB layout software, and mass-produced by a board shop. Like the previous (hand-made) PCB, this one is fairly minimalist in that it consists of bare metal traces laid on a fiberglass board:

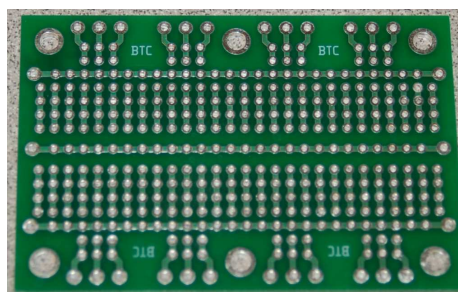


Close comparison between this photograph and the photographs of the hand-made PCB reveal a different color of metal trace. Unlike the bare copper traces of the hand-made board, the traces on the mass-produced board have a silver-like finish. These are *tinned* traces: copper traces covered in a layer of tin for better resistance to corrosion and therefore better solderability.

Another important difference between this PCB and the hand-made board is that this mass-produced board has metal *inside the through-holes*, visible by close inspection of the above photograph. This “through-hole plating” creates greater conductive surface area for soldering between component leads and traces, as well as helping to anchor the fragile metal traces to the fiberglass substrate.

Additional layers of material on the PCB help make it even more robust as a foundation for electronic circuits. One of those layers is called *soldermask*, an epoxy-based material applied over the top of a PCB to cover up all but the pad areas where components will attach. Soldermask helps prevent unintentional “bridging” between bare metal traces that might occur during the soldering process, as well as greatly improving the reliability of reflow soldering for SMD boards by inhibiting the motion of molten solder paste across areas it doesn’t belong. This same soldermask layer adds further corrosion resistance to the metal traces underneath.

The same PCB design shown above is now shown with a green layer of soldermask and lighter-colored silkscreened (white) text applied:



Note how tinned metal surfaces are still visible at all the connection points (pads), whereas all the traces now appear green in color due to the covering of the green-hued soldermask layer.

Incidentally, soldermask is available in a wide array of colors, although green seems to be the most common.

One additional layer found on some, but not all, finished PCBs is *conformal coating*. This layer is applied to a PCB only after all components have been soldered in, its purpose being to provide a protective coating for the components themselves against environmental threats such as moisture, dust, etc. Conformal coating materials vary with the protective purpose (silicone, polyurethane, and acrylic being typical compounds), and since these constitute an additional manufacturing cost are only applied on PCBs which must survive harsh atmospheres.

## 3.3 Layout tools

Laying out PCB traces and pads by hand is thankfully an obsolete practice, especially for volume production. The modern way of PCB design uses specialized computer-aided drafting software to plot the position and interconnections of components on a printed circuit board. This software then outputs digital data files containing all the information a PCB manufacturer would need to cut, etch, drill, and in some cases test, the boards.

PCB design software may be broadly divided into two functions: *schematic capture* and *board layout*. Schematic capture is where the user draws a complete schematic diagram showing all the necessary components and interconnections in abstract form. Board layout takes that information contained in the schematic and translates it into component positioning and trace routing, all with physical dimensions. Some of these tasks are automated, but much of it must be done by the user in order to ensure a final design that is both fully functional and aesthetically pleasing.

All PCB design software provides some form of *Design Rule Checking* or *DRC* to automatically test the layout against a set of rules specifying minimum clearances (e.g. distance between conductive areas to help prevent arcing between traces as well as manufacturing defects such as solder bridges), minimum track widths (to help ensure traces have sufficient ampacity to conduct reasonable amounts of current), and other parameters necessary for a functional circuit.

Both commercial and free PCB layout software exists to serve PCB designers at all levels. A very popular software package for PCB layout at the time of this writing (2022) is KiCad, which strives to be an all-in-one tool for the entire workflow from design to production. KiCad runs on a wide range of operating systems and has a large user base. A more specialized PCB layout tool, developed for personal computers running the Linux operating system, is appropriately named `pcb`, part of the gEDA suite of open-source electronic design automation (EDA) tools. `pcb` is a stand-alone PCB editor with no schematic capture capability, and relies either on other software such as `gschem` for schematic capture or on the user for determining the proper terminal-to-terminal connections. However, the links between these other applications with `pcb` may be automated using scripts<sup>1</sup>

Professional-grade PCB design software applications include Altium Designer, Autodesk Eagle, Proteus, Metro Logic, and OrCad. These sophisticated tools require either single-time payment or ongoing license subscription fees, but in some cases there may be limited-function versions of these programs for trial use.

Once a PCB design is complete enough<sup>2</sup> for production, the user exports a series of data files using the PCB layout software. The most common data file format used for PCB production is the *RS-274X* format commonly referred to as *Gerber*. Gerber files are human-readable ASCII-encoded text files describing the physical locations and dimensions of all PCB features. Typically the layout software creates one Gerber file per layer on the finished board (e.g. bottom copper layer, top copper layer, bottom soldermask layer, etc.) as well as a *drill file* specifying where the computer-controlled drilling machine will place holes in the board. It is customary for PCB creators to combine this set of files into a single compressed “zip” file which is then communicated electronically to the PCB

---

<sup>1</sup>Here, a “script” refers to a simple set of instructions for a computer to follow; i.e. a very simple *program*.

<sup>2</sup>It is a common principle of design that no creative project is every truly *complete*. No matter how much time or effort you expend on the creation of a project, there will always be something you will want to change or improve. One of the traits every creative person develops at some point in their lives is the ability to “let go” of a project and not become mired in an endless cycle of editing and evaluation.

manufacturer for production. The board shop then takes this digital information to produce the requested quantity of PCBs, and in some cases solder paste stencils useful for production of finished boards with surface-mount devices (SMDs).

## 3.4 Basic layout elements

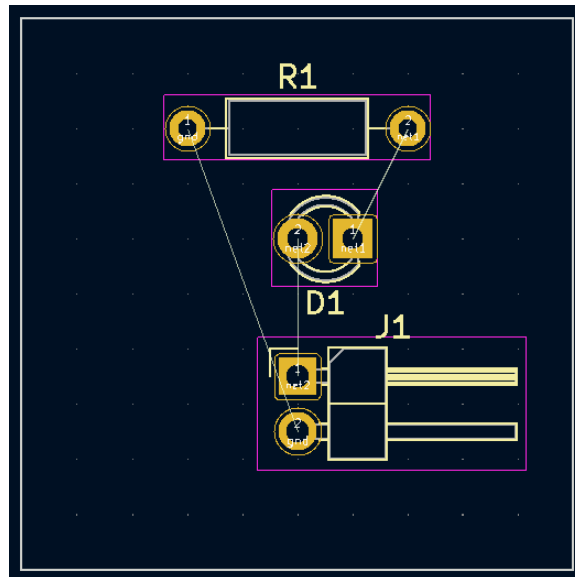
Every PCB design project involves the following elements in one form or another:

- **Schematic capture** – all circuit design begins with a conceptual representation of how the circuit’s components will interconnect to form a functional circuit. At the simplest level, suitable only for the most elementary of circuits, this takes the form of a mental plan. A major step up in sophistication from the mental-only plan is to draft a schematic diagram on paper. Another major step up in sophistication is to use schematic-capture software to draw this diagram.
- **Specifying dimensions and units** – printed circuit boards are physical objects, and as such they possess physical dimensions which must be specified in their design. The unit of the *thou* or the *mil* (i.e. 0.001 inch) is quite common in the PCB design world, and a great many components are constructed in such a way that their terminals are spaced apart some whole-number of mils. For example, integrated circuits following the *DIP* (Dual Inline Package) form factor have a terminal-to-terminal pitch of 100 mils (0.1 inch) along each side. However, many components have *metric* terminal spacing which means whoever designs a PCB containing both metric-pitch and English-pitch terminals must switch units when drafting the layout. Component datasheets typically specify physical dimensions as well as electrical characteristics, and these dimensions are often expressed in both English and metric units written side-by-side on the page with no “mil” or “mm” labels. In order to tell which is which, it is good to know that one inch is precisely equal to 25.4 millimeters!
- **Component footprints** – every component represented on the schematic diagram has a physical representation of its package, its connecting terminals, etc., and in the world of PCB design this is called a component *footprint*. Footprints may be manually created in the board layout software if necessary, but these software packages always provide libraries of common electronic component footprints that may be used instead. Custom footprints created by the user may be added to the library of pre-existing footprints for future PCB design projects. Footprints not only specify the location and size of the copper *pads* to receive the component’s terminals for a soldered joint, but also the hole drill size (for through-hole components), as well as an outline of the component and text labels to be printed on the board’s silkscreen layer.
- **Interconnection nets** – every set of electrically-common points in a circuit form a unique entity known in the PCB design world as a *net*. Associating each component terminal to a net is something done automatically by schematic capture software, but this may be done manually by the user if desired, or even skipped entirely if the circuit design is simple enough for the user to remember which terminals must connect to which. Once a *netlist* exists specifying all the electrical interconnections between component terminals, the board layout editor is able to show these nets as a collection of thin lines called a *ratsnest* joining all the electrically-common terminals together, and these lines in turn show the user where traces (also called tracks) must be placed to form copper connections between those points on the PCB.



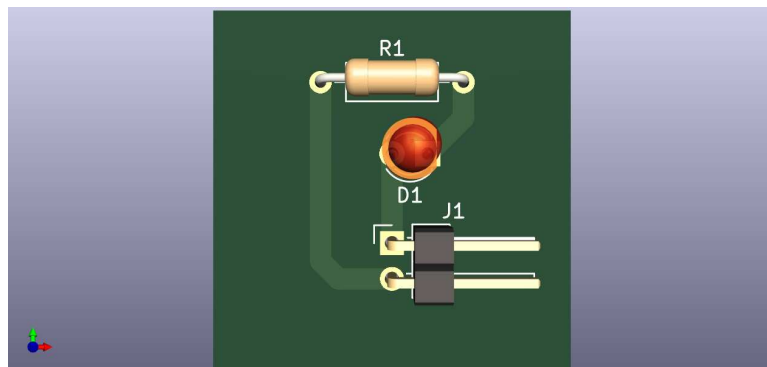
- **Routing traces/tracks** – this is where actual traces are drawn between component terminals in the board layout software to define where copper will remain on the board. Again, this layout may be done in a completely manual way by the user if the circuit is simple enough, but for any non-trivial circuit design a ratsnest is very helpful to ensure all terminals are connected properly and that no traces are accidentally omitted.
- **Layers** – most computer-aided drafting (CAD) software applications provide a feature called *layers* to break the visual design into separate superimposed entities. In a general sense, layering makes complex drafting projects more manageable. In PCB design, however, layering takes a more literal meaning because the final printed circuit board itself is actually composed of distinct layers of material. For example, most PCBs have a separate layer of copper traces on the back (bottom) side of the board as well as on the front (top) side. Covering those copper-trace layers is another layer of *soldermask* to help prevent solder from bridging between terminals and/or traces. Covering the soldermask is another layer of printed ink called *silkscreen* for component identification and other notes. For PCB designs using surface-mount components, another critical layer is the *solder paste* layer useful for creating a metal template for accurately applying solder paste to all the SMD device pads. Whenever the user routes a trace or adds text to a PCB design, it is imperative they first select the desired layer in the editing software so that the added element goes where it belongs. Forgetting to first select the proper layer when adding elements to a board layout is a common beginner mistake.
- **Adding vias** – a *via* is a copper connection extending from one layer of the board to another, usually added when it becomes necessary for a trace to “jump” to another layer in order to reach a component terminal without intersecting an electrically distinct trace. Vias find common application in board layouts where many surface-mount devices exist. One advantage of through-hole devices is that every device terminal offered an opportunity for copper traces on all conductive layers of the board to join, but surface-mount devices of course have no need for holes drilled through the entire board and so do not naturally present these same inter-layer connection opportunities.
- **Additional artwork** – “artwork” is a term used in PCB design to refer to *any* elements added to a board layout, including traces, component outlines, and component labels. However, “artwork” may take a more literal meaning in the form of silkscreen-layer text and/or images to personalize the final design.
- **Design Rule Checking** – often abbreviated as *DRC*, this automated feature of PCB design checks your completed board layout against several pre-defined rules for such things as adequate clearances, minimum widths and hole sizes, etc. Some board manufacturers provide rudimentary DRC at the time of file submission, prior to manufacturing.

Consider the following circuit, so elementary it needs no schematic diagram to describe its connections. It simply consists of a light-emitting diode (LED) labeled **D1**, a series-connected resistor **R1**, and a two-terminal connector **J1** for DC power:

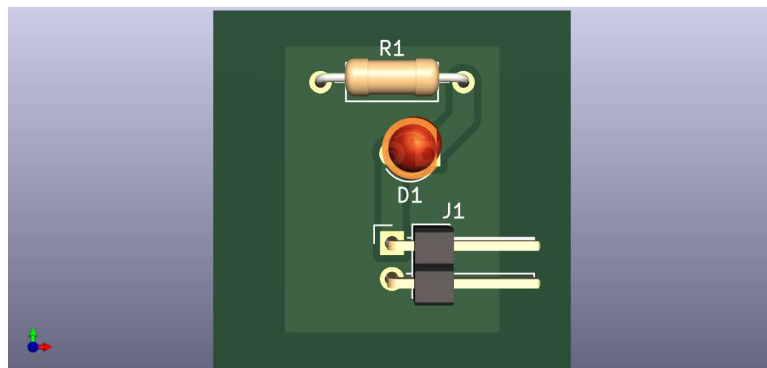


Note the thin, straight, white-colored lines joining certain component terminals together. This collection of lines is the *ratsnest*, and if you look closely at each pair of terminals joined by one of these lines you will see those joined terminals bear a common *net* name. In this particular circuit we see terminal 1 of resistor R1 and terminal 2 of connector J1 both have the net name **gnd**; terminal 1 of the light-emitting diode D1 and terminal 2 of resistor R1 both share the net name **net1**; finally, terminal 2 of the light-emitting diode D1 and terminal 1 of connector J1 share the net name **net2**. These net names instruct the board layout software where to draw the interconnecting ratsnest lines, which in turn serves to guide the user where to place copper traces to finalize those connections.

One simple routing scheme links these three components together with top-layer copper traces 60 mils (0.060 inch) in width, the result shown in a visual rendering by the board layout software:



An alternative routing strategy makes use of a rectangular *copper fill* area rather than a trace as the “ground” (**gnd**) conductor:



Note how the board layout software automatically places clearance gaps between the other traces and the new copper fill to ensure non-contact. Those points intentionally connected to the fill were specified by net name: when the copper fill was added to the board layout, the dialogue window prompted the user for a net name, and when **gnd** was selected as that name the software joined both terminal 1 of the resistor and terminal 2 of the connector to the new copper fill while excluding the other traces.

Using a filled area of copper rather than a relatively thin trace obviously lends more current-carrying capability to that portion of the circuit, but usually this is not the main reason to use copper fills (also called copper *pours*) rather than traces. Other reasons include minimization of parasitic inductance and minimization of electric-field coupling between other traces. Entire layers of a multi-layer PCB consisting of solid copper-fill are called copper *planes*, and when these are used for DC power rails they lend the additional advantage of natural decoupling capacitance<sup>3</sup> to help

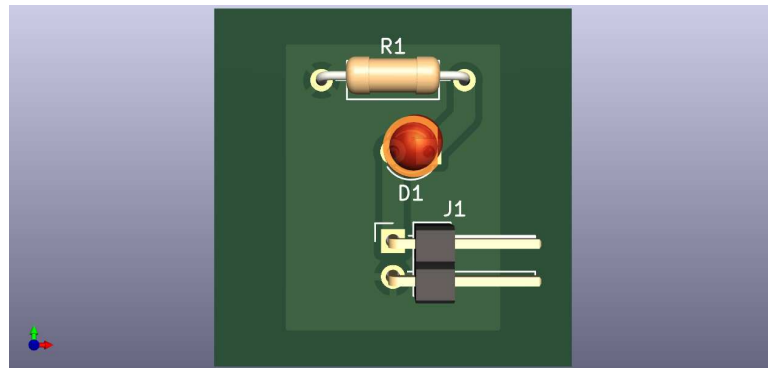
<sup>3</sup>Capacitance exists wherever an electric field spans a space between two conductive surfaces. This electric field can store energy, and that stored energy makes the electric field (and hence the voltage associated with that field) resistant to rapid changes. Thus, where there is capacitance, voltage cannot change without energy being transferred either into or out of that electric field; *i.e.* *voltage cannot vary without work*, helping to stabilize DC voltage over

stabilize DC supply voltage.

Copper fills are not without their disadvantages, though. One consideration, especially for large PCBs, arises from the fact that copper metal has a vastly greater temperature coefficient of expansion than FR4 fiberglass, which means if the PCB is soldered in any automated process where the entire board is heated to soldering temperature, the copper fill will physically expand more than the fiberglass substrate which may warp the board. A mitigation for this effect is to *balance* the distribution of copper-fill areas across the PCB’s multiple copper layers so that physical stresses caused by the copper’s expansion with heat will be approximately symmetrical on either side of the PCB’s center plane. This is known as *copper balancing*.

Another caveat to the use of copper fills and planes is that their relatively large surface areas function as a *heat sink* during the soldering process, drawing heat away any localized heat source applied to a solder joint. This makes manual soldering more difficult. This is not a problem in automated “wave” or “reflow” soldering where the entire board’s temperature briefly raises to the melting point of the solder, but any board thus designed will be more difficult<sup>4</sup> to manually rework or repair if needed.

The typical mitigation for this undesired “heat-sink” effect is to apply *thermal reliefs* to all component pads electrically common to the copper fill, as seen in this visual rendering:



Note how the two “ground” terminals now connect to the copper fill using thin radial spokes between the circular copper pads and the fill, with a 20 mil clearance otherwise between the two. The board layout software places these thermal reliefs automatically when the relief option is selected.

---

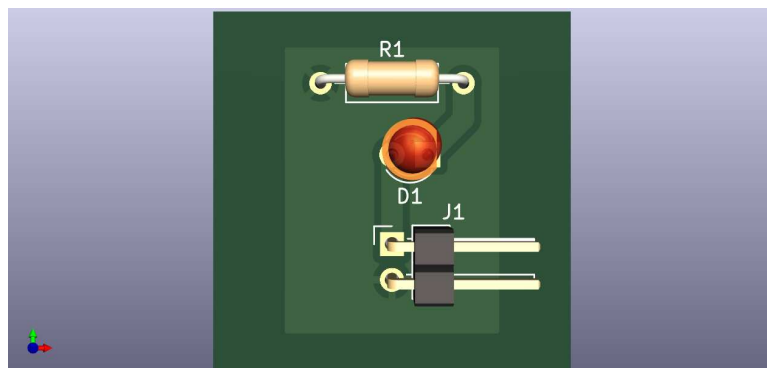
time, as the capacitance serves as an energy reservoir for all parallel-connected components.

<sup>4</sup>A common technique to remove and re-apply solder to a copper fill is to use a temperature-controlled hot-plate surface to pre-heat the board to a temperature less than the melting point of solder, then use a manual soldering tool to raise the temperature of the component terminal and the pad area just enough for the solder to liquefy.

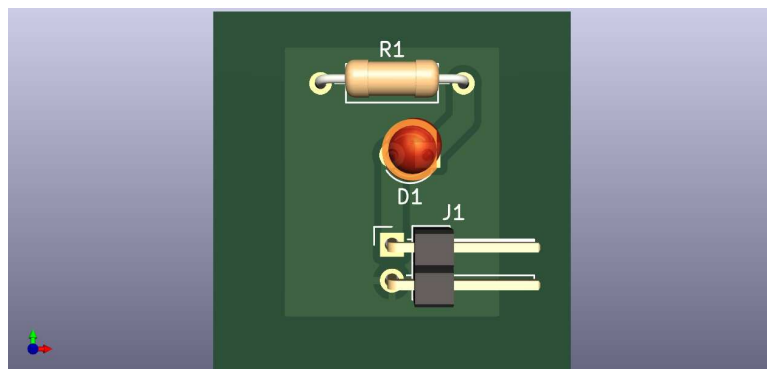
### 3.5 Thermal design principles

Printed circuit boards not only provide necessary electrical connections between components, but they may also provide *thermal* connections between heat-dissipating components and “heat sinks” in order to keep those heat-dissipating components cool. Copper happens to be an excellent conductor of heat in addition to being an excellent conductor of electrical charge carriers, which means we may judiciously apply copper traces and fills to aid heat transfer when necessary.

For example, the following visual rendering of a board layout shows the use of a front-layer copper “fill” connecting the left-hand terminal of resistor R1 to the bottom terminal of connector J1. As per standard practice, the junctions between those terminal pads and the copper fill are *thermally relieved* by a clearance gap at each to minimize heat transfer and radial copper “spokes” to ensure electrical connectivity, so that the copper fill will not act as a gigantic heat sink which would complicate manual soldering of those two component terminals to their respective pads:



However, if resistor R1 was known to be a large dissipator of heat in this circuit application, it would actually be a good thing for the copper fill to fully connect with the resistor’s pad. Thus, we may edit that pad’s properties in the board layout editor to have a “solid” connection rather than a “thermally-relieved” connection which will make it a *heatsink pad*:



Of course, using a copper fill in this manner is practical only if we also need that fill to be an electrical connection to the hot component’s terminal(s). For applications where we might wish to use a nearby copper fill as a heat sink but cannot connect to that fill electrically, there exist

special *thermal jumper* components designed specifically for this purpose. These components come in standard surface-mount device sizes, with metal end-caps suitable for soldering to the PCB's copper pads and an aluminum-nitride core with high thermal conductivity but low electrical conductivity for conducting heat away from the dissipative component and to the cooler copper fill.

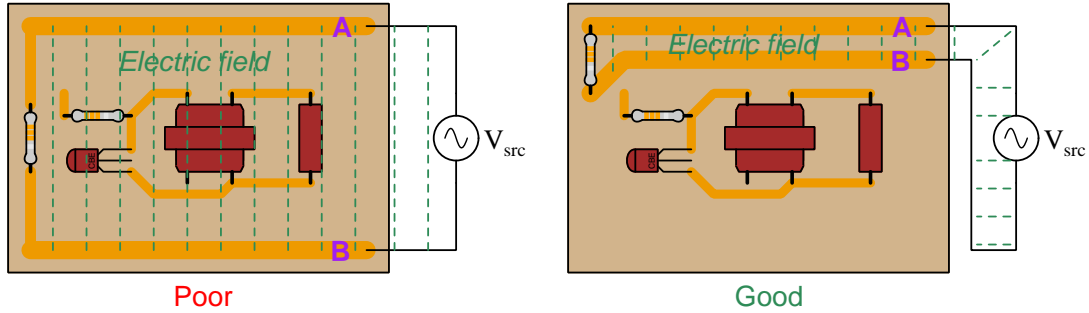
## 3.6 Preserving signal integrity

Electrical signals that vary over time may unintentionally “couple” from one trace on a PCB to another, either by the parasitic capacitance naturally existing between any isolated conductors and/or by the parasitic inductance naturally existing along any conductor magnetically coupling to an adjacent conductor. Sometimes known as *crossstalk*, this coupling may corrupt one or both of the signals. Best practices for avoiding such coupling to maintain the integrity of all signals on a PCB include the following:

- Separate offending (“aggressor”) traces as far apart from susceptible (“victim”) traces as possible. This decreases parasitic capacitance between the two traces as well as decreasing magnetic coupling between them.
- If traces must cross paths, do so perpendicular to one another and avoid running them parallel to one another. Perpendicular crossing may be done on different layers of the PCB, or by routing one trace beneath a component bridging a gap between two other traces on that layer, so that the one trace is perpendicular to the current-axis of the component. Perpendicularity of traces ensures minimal overlapping area (thus decreasing parasitic capacitance) as well as ensuring minimal mutual inductance between the two traces.
- Adjacent inductors should be mounted with their magnetic centerlines perpendicular to each other for the same reason, to minimize magnetic coupling between the two (i.e. mutual inductance).
- Make liberal use of ground and DC power “planes” made of wide copper “pour” areas rather than thin copper traces to deliver DC power to components. Not only will these wide regions of copper result in lower current density, but they also provide an electrostatic shielding effect to surrounded traces much like electrostatic shielding on a cable.
- Traces carrying high-speed signals (i.e. signals with fast rise/fall times) should never pass over a gap between adjacent copper pours, as this will introduce an impedance discontinuity causing signal reflections.
- Any pathways for high AC current should encompass the least surface area possible on the PCB. Minimizing current loop area has the effect of minimizing the amount of magnetic flux created by any given amount of current through that loop.
- Traces between which high AC voltage exists should be kept as close together as safely possible in order to minimize the area over which their electric field will spread.

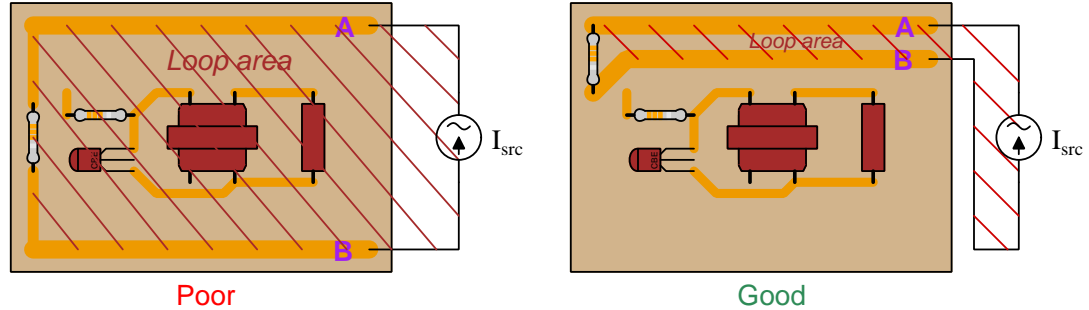
Those last two points, about minimizing high-current loop areas and minimizing separation distance between high-voltage conductors, are illustrated below. Electric field containment is improved when conductors bearing high AC voltage between them do *not* have any other components or traces between, and are as close together as safety will allow:

#### Containment of electric fields



Magnetic field containment is improved when conductors bearing high AC currents encompass the least amount of area:

#### Reducing loop area to mitigate magnetic fields



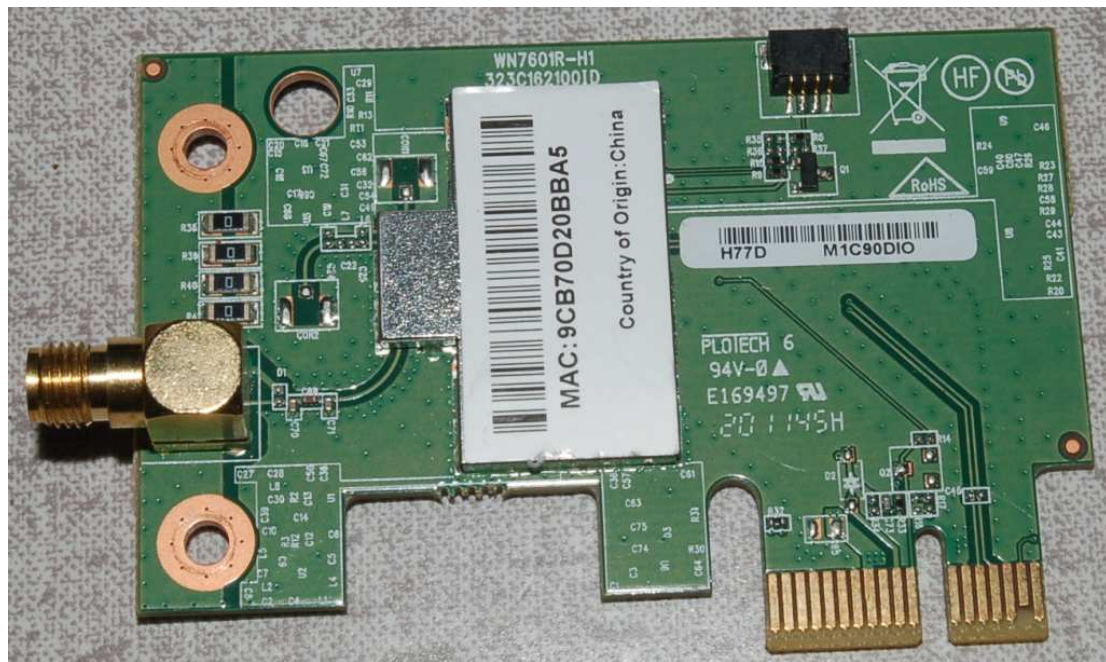
Such concerns are not generally present if the high voltages and/or currents are *DC* or have very low frequency, since capacitive coupling works on the principle<sup>5</sup> of  $I = C \frac{dV}{dt}$  and inductive coupling on the principle<sup>6</sup> of  $V = L \frac{dI}{dt}$ . With DC there is no change in voltage or in current, making both  $\frac{dV}{dt}$  and  $\frac{dI}{dt}$  equal to zero. Low frequencies naturally have lower  $\frac{dV}{dt}$  and  $\frac{dI}{dt}$  values than high frequencies for the same peak values of Volts and Amperes because the waves do not oscillate back and forth between those peak values as often over time; i.e. their rates of rise and fall are less at lower frequencies.

<sup>5</sup>In this case,  $I$  represents the current induced in the victim trace and  $V$  the voltage between the victim and aggressor traces.

<sup>6</sup>In this case,  $V$  represents the voltage induced along the length of the victim trace,  $I$  the current flowing through the aggressor trace, and  $L$  the (mutual) inductance between the two traces.

### 3.7 High-speed design principles

Traces laid out on printed circuit boards (PCBs) act like transmission lines just like cables and individual wires at radio frequencies. That being the case, careful layout is important for good circuit performance. Below we see a PCB for a personal computer Wi-Fi radio network adapter. Note the female SMA connector on the left-hand edge of this PCB, awaiting connection to an antenna:



Several features of this PCB are worth noting for their high-frequency qualities. First, the curved trace connecting the center-pin of the SMA connector to the RF amplifier (housed under the metal cover) is a good example of careful trace routing intended to maintain a consistent characteristic impedance along its length. Not only does the trace avoid any sharp turns, but the consistent clearance between the trace and the surrounding copper “pours”<sup>7</sup> ensures the traveling signals experience consistent parasitic capacitance and inductance between those conductors. A sharply-angled trace here would have presented a discontinuity to any signals traveling along it, and thereby caused signal reflections; inconsistent clearance between the trace and surrounding ground plans would have also given the line different characteristic impedance values at different locations. Second, we see a differential (balanced) signal pathway in the form of two traces extending from two of the card-edge connector pads, ending in a pair of *vias*<sup>8</sup>. Third, it is impossible not to notice

<sup>7</sup>A *copper pour* on a PCB is simply an otherwise unused area of a board layer that is filled with copper instead of being left bare. This stands in contrast to a *copper plane* which is an unbroken layer of copper covering the entire area of one board layer.

<sup>8</sup>In PCB design, a *via* is a metal-plated hole passing between two or more copper layers on the board, intended to connect a trace on one layer to a trace on another layer. These differ from through-hole component *pads* in their lack

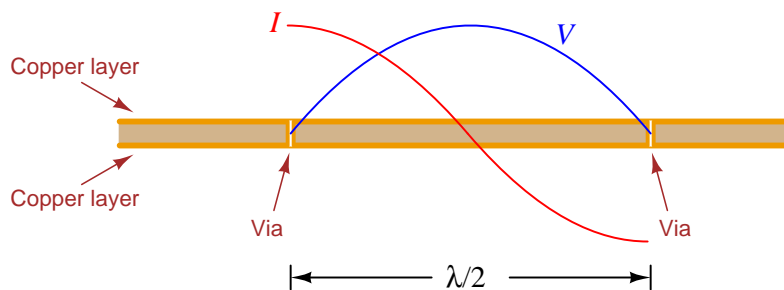


the abundance of vias connecting large copper areas on the top side of the board to corresponding copper areas on the bottom side. This next photograph shows the underside of this same Wi-Fi PCB, where the vias connecting top and bottom copper planes appear as tiny holes:



In DC and low-frequency AC applications, such an abundance of vias would serve no purpose other than for high-current applications. Ordinarily, just a single via would be necessary to make two copper surfaces electrically common to each other, but here we must “stitch together” those opposing copper layers far more frequently. A good practice is to space these “stitching” vias no farther apart from each other than one-twentieth of a wavelength ( $\frac{\lambda}{20}$ ) for the highest anticipated signal frequency, with even closer spacing being better.

Concern over via spacing is rooted in transmission line theory, particularly in the phenomenon of *standing waves*. When two conductive surfaces are electrically joined at two locations, the area between those joints has the ability to resonate like a transmission line shorted at both ends. The short-circuit (via) at either end ensures voltage *nodes* (zero) and current *anti-nodes* (maximum), with the lowest-frequency standing wave being a half-cycle. Therefore, vias placed at  $\frac{\lambda}{2}$  intervals invite resonance, which in turn will degrade the performance of adjacent RF circuitry:



A maximum spacing recommendation of  $\frac{\lambda}{20}$  comes from the time-honored engineering practice of designing things at least one order of magnitude ( $10^1$ ) more conservative than some critical limit in order to render that critical effect inconsequential. Placing vias closer together means the resonant frequency for any standing wave must increase (i.e. higher  $f = \text{shorter } \lambda$ ), and so choosing to

---

of bare copper areas to solder, since vias are nothing more than inter-layer connectors and not intended as termination points for components.

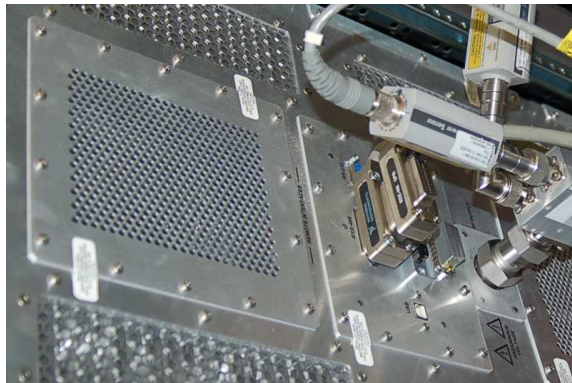
space vias closer together means resonance is possible only at frequencies greater than the circuit is designed to operate.

Below are top- and bottom-layer photographs of another RF board, this one being a low-noise amplifier for boosting the strength of RF signals prior to the input of a radio receiver. The top side of this PCB (left) hosts all the components and is rather simple in layout, having three distinct copper “pour” areas surrounding traces. This board’s bottom side (right) is nothing but a solid copper ground plane, with vias “stitching” the bottom-layer ground plane to all three of the top-layer pours:



Rows of vias “stitching” top-layer ground areas to the bottom-layer ground plane are plentiful on this PCB, and as you can see they are necessary around the circumference of each pour.

The reason for the close spacing of “stitching” vias on a PCB is exactly the same as the reason why metal cover plates on RF devices are fastened by a seemingly over-abundance of screws: to ensure the distance between connection points is significantly shorter than a half-wavelength of the RF signal in question. The following photograph is a good example of this, there being far more screws fastening each cover plate to the metal chassis than mechanical considerations alone would require:

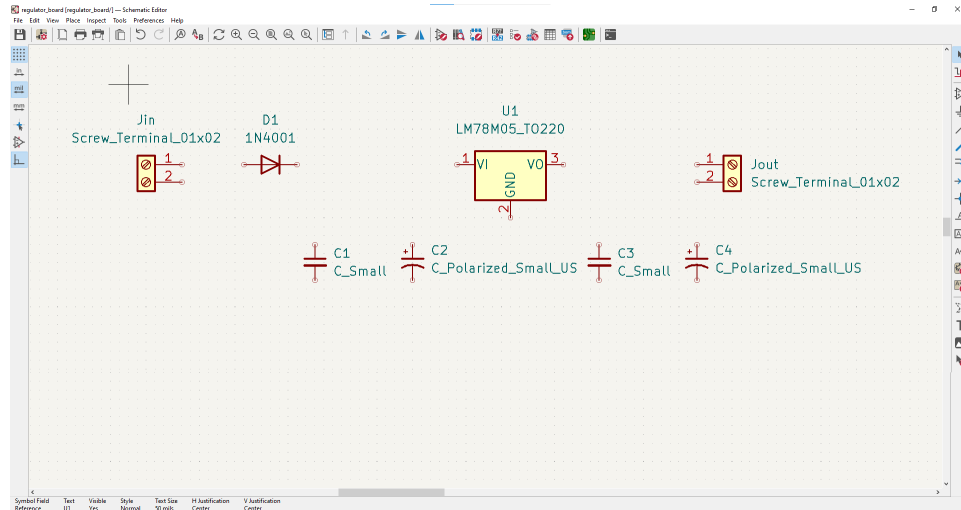


### 3.8 KiCad design example

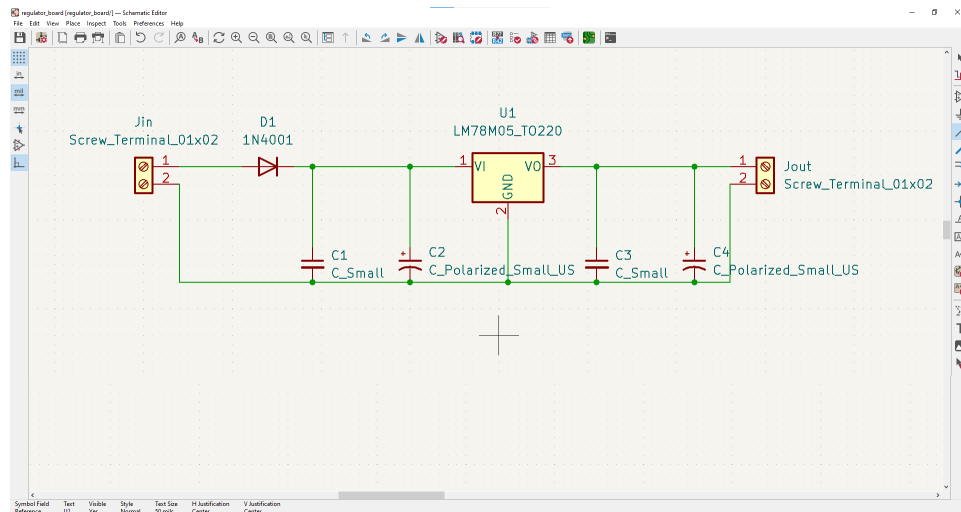
In this section we will explore the use of the popular KiCad software to design and lay out a simple voltage regulator circuit using the LM7805 three-terminal voltage regulator IC. The version of KiCad used here is 6.0.7 running on Microsoft Windows operating system.

### 3.8.1 Schematic capture using KiCad

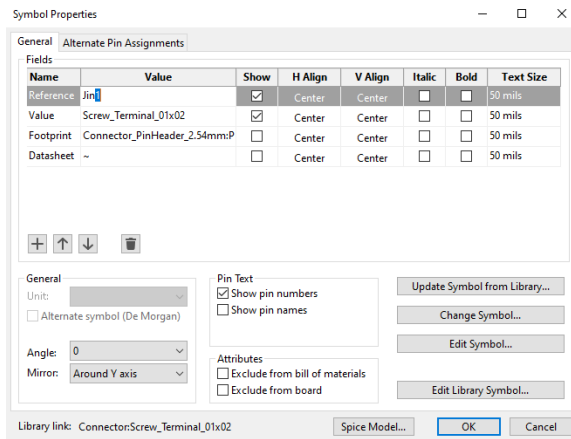
First we must create a schematic diagram using the Schematic Editor. Within the “Place” menu we find options to place components (called “symbols”):



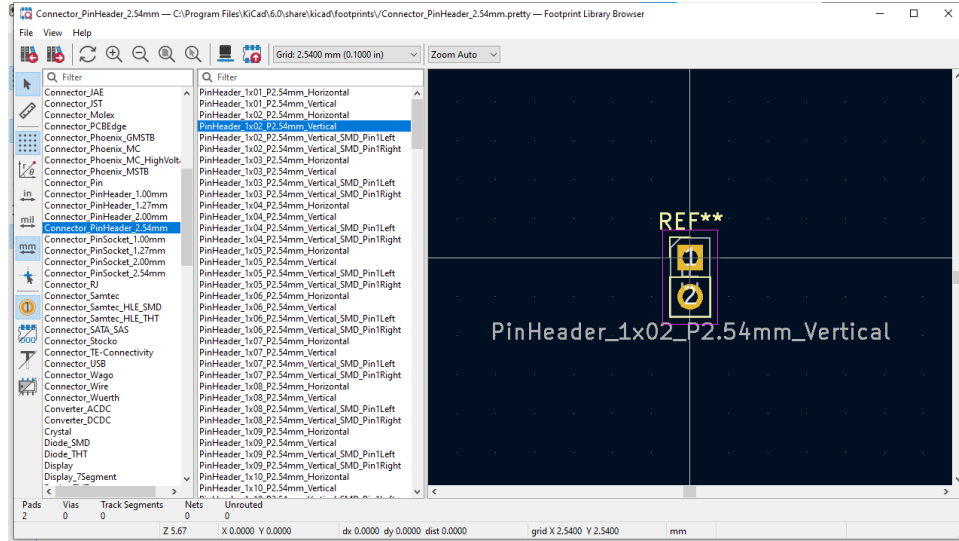
After arranging the components where we want them in the schematic, we also “place” connecting wires to form a complete diagram:



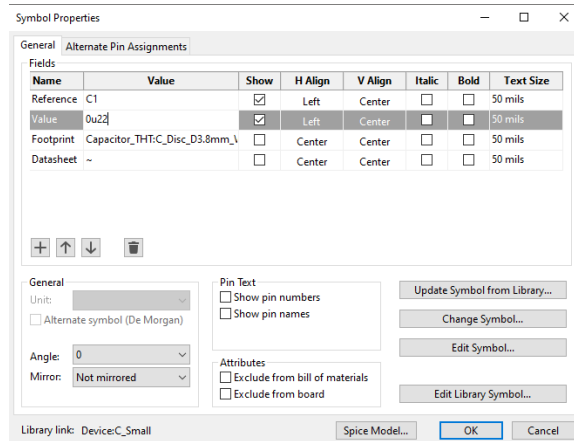
When we place each component on the schematic diagram, its label will have a question-mark after the component designation letter. For example, each of the four capacitors in this diagram started with labels of **C?** when initially placed. To change this, we double-click on each component in turn to open a “Symbol Properties” window and assign to it a component number. Below we see the Symbol Properties window for the input connector, which we have labeled **Jin1**:



Some of these components come with pre-designated PCB *footprints* describing their terminal locations and dimensions, ready for layout in the PCB Editor. Other components, however, are generic and have no pre-assigned footprints. For those generic components, we must assign a realistic footprint in the “Footprint” field found in the same “Symbol Properties” window. For this we will need to know the terminal spacing of our actual components to be soldered into the final PCB. Each footprint specifies where the pads will be located in the final PCB layout. A typical list of component footprints is shown below, this particular selection being for one of the dual screw terminal assemblies used in this circuit for input and output power connections:

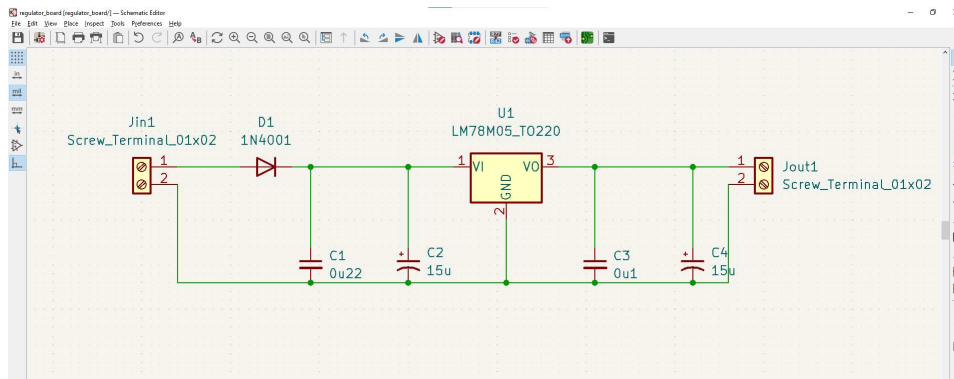


The Symbol Properties window is also where we assign values to components. Here, we are assigning a value of  $0.22 \mu\text{F}$  to capacitor  $C_1$ :



Component values are relevant for PCB layout only if we wish those values to be printed on the PCB (usually on the silkscreen layer).

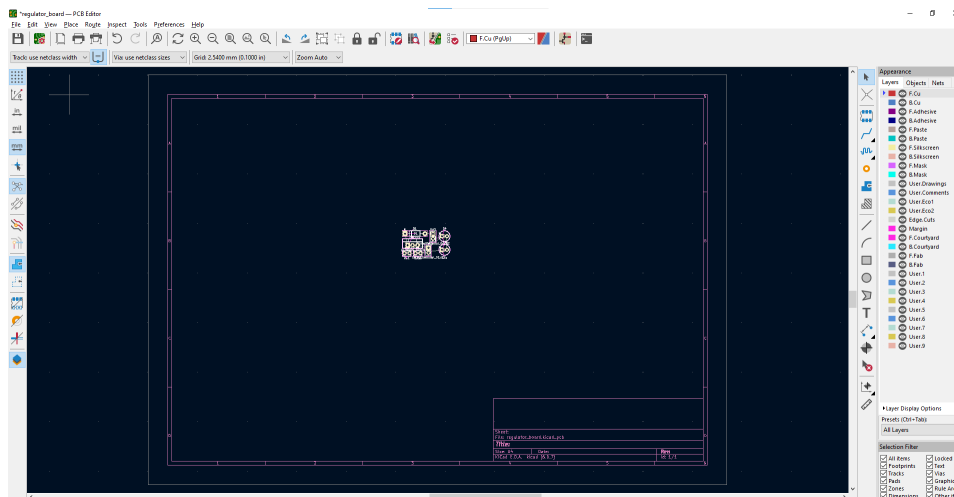
Finally we have a complete schematic diagram, ready to translate into a PCB layout:



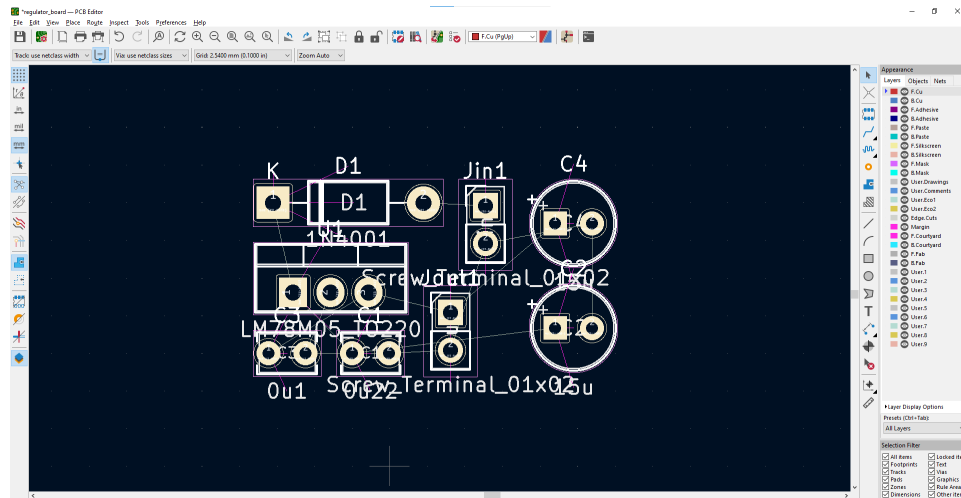
A major reason many PCB layout applications such as KiCad have us start by drawing a schematic diagram is that the schematic defines which component terminals connect to one another. This logical association between component terminals becomes very helpful during the PCB layout process.

### 3.8.2 PCB editing using KiCad

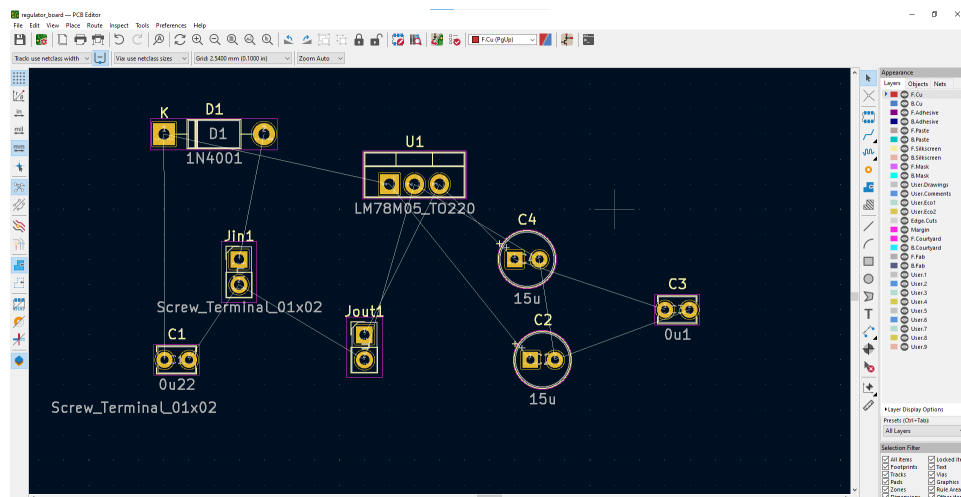
Under the “Tools” menu lies the option to *Update PCB from Schematic* which transitions to the next phase of the workflow, opening up the PCB Editor window. When this window first opens, we are greeted by a dense collection of component footprints at the center of the space, representing all of our circuit’s parts crammed together in no particular order:



If we zoom into this compact collection of footprints, we see a jumbled mess:



Spreading these footprints apart from each other reveals thin lines connecting certain terminals together, those thin lines “stretching” as far as needed as we move the footprints around:



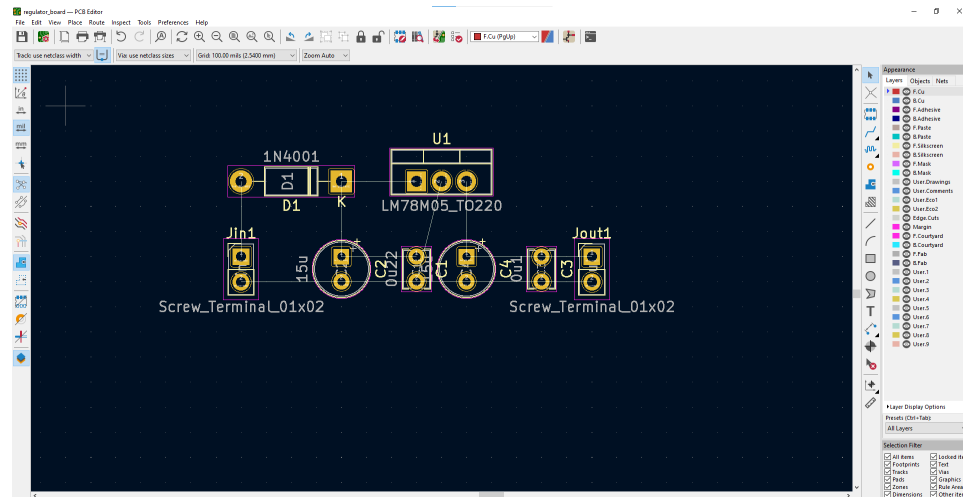
Each of these lines is called a *net*<sup>9</sup> in KiCad. All together, the collection of nets is referred to as a *ratsnest* due to its typically jumbled and messy appearance. Our major task in arranging all the component footprints is to place them in such positions that the metal traces are able to route as needed from component terminal to component terminal with a minimum of crossing (to avoid trace-to-trace signal interference) as well as abiding by other electrical and physical design requirements. A recommended strategy is to move the footprints around, rotating them as necessary, until the

<sup>9</sup>Some PCB editing software applications refer to each line as a *rat*, for reasons that will become clear very shortly.

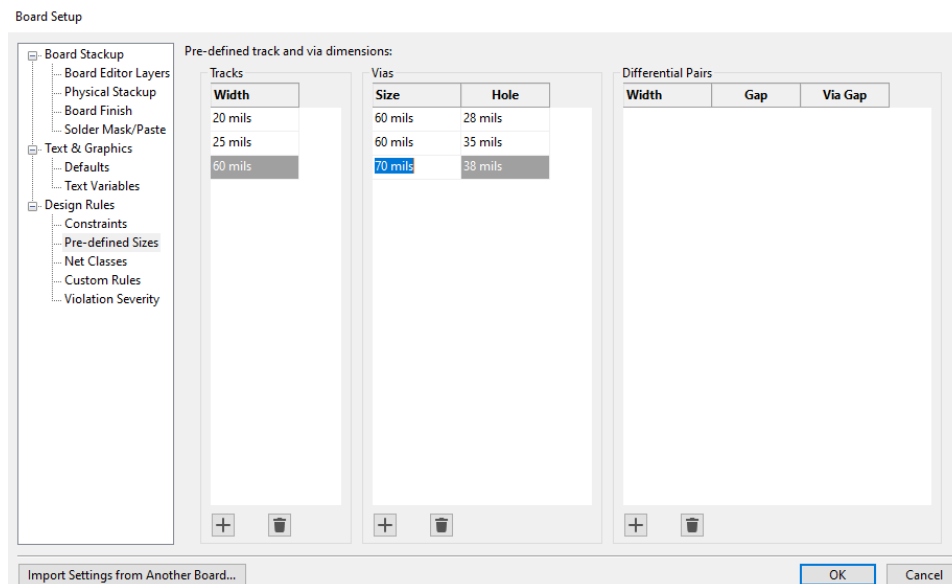


ratsnest looks “clean” and orderly. Once we have a sensible ratsnest, converting each net to a metal trace will be much easier.

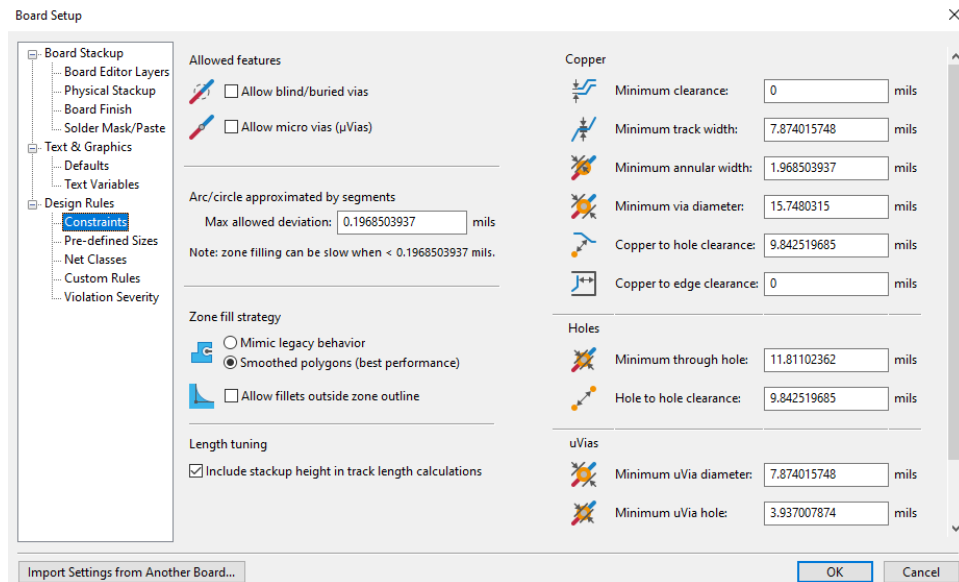
Here we see a much more orderly ratsnest than before, ready for conversion into traces:



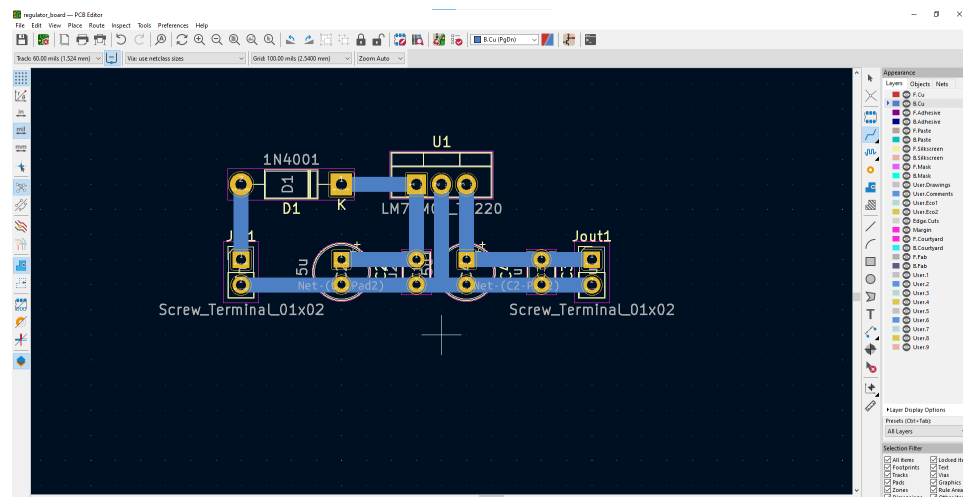
Before we start *routing* each net line into a trace, though, there are some important parameters we need to set. One of these parameters is the width of the traces we will later route. Also, we will specify the outside and inside diameters for all vias to be placed on the board. In KiCad these parameters are located in the “Board Setup” window, in the Design Rules section, under Pre-defined Sizes. Shown here are some sizes I’ve find useful in my own PCB layout work:



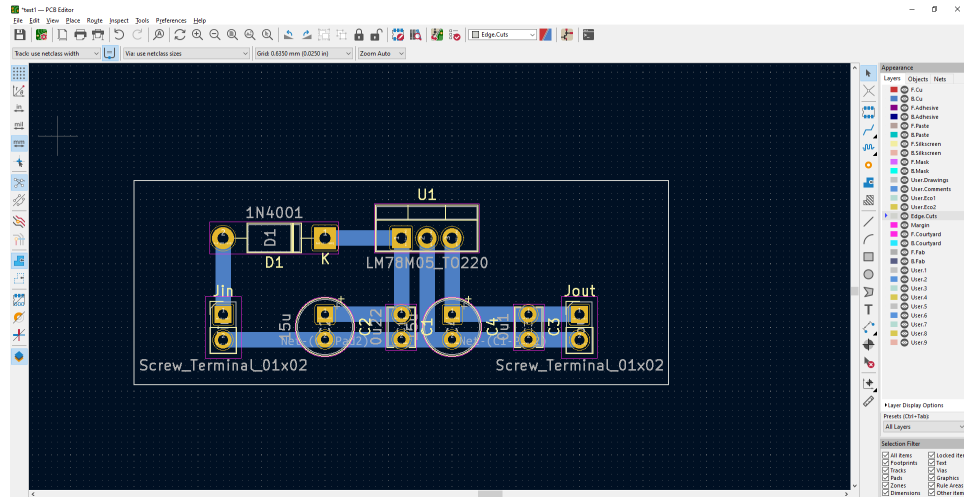
Also important prior to routing is the setting of design rule constraints, such as minimum track widths, minimum copper-to-hole clearance, minimum clearance between tracks, etc. These are also set within the “Board Setup” window, the values shown here being the default when I started my first session of KiCad:



With these parameters verified, we can proceed to routing traces. As traces are drawn between component terminals, the net lines formerly connecting those points together will disappear. The software still tracks all net connections, though, so that net lines will re-appear if any tracks are deleted. For this PCB layout I chose my widest pre-defined track width of 60 mils, selecting the bottom copper layer (B.Cu) on the “Layers” list:

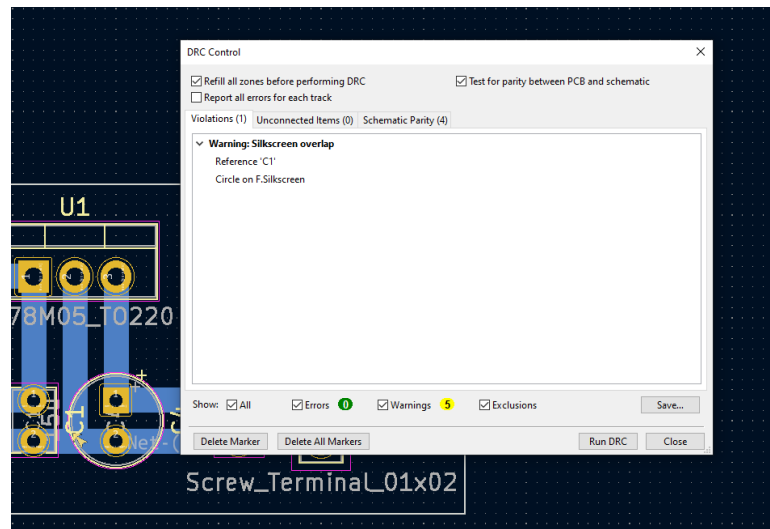


We also need to add an outline for the printed circuit board itself, and this is done on the **Edge.Cuts** layer in the PCB editor. Simply switch the layer to **Edge.Cuts** and draw a rectangle around the layout to define the PCB edges:

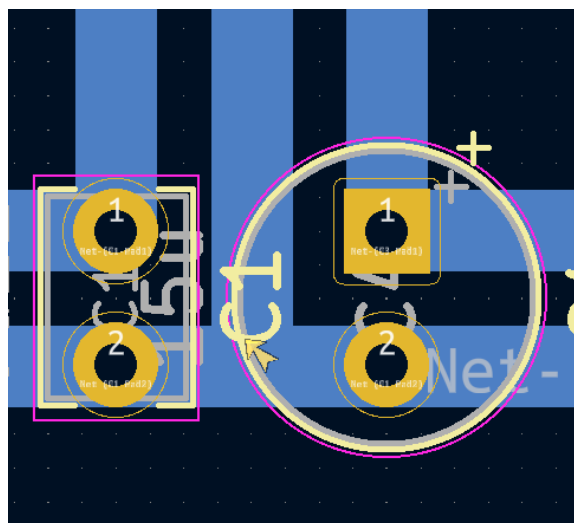


### 3.8.3 Design rule checking using KiCad

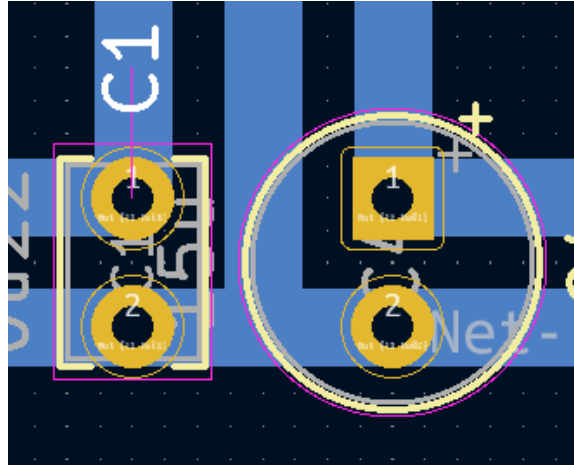
An important step in any PCB layout design process is to check the layout against the pre-defined design rules to ensure the design is compliant with those rules. When we run a DRC on this layout, the computer reports one rule-check violation and four schematic parity warnings:



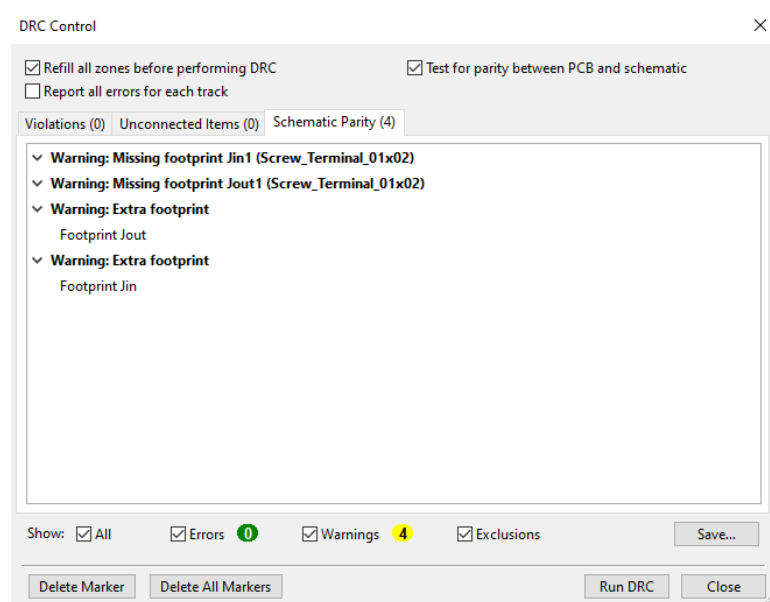
Zooming in to the violation, conveniently highlighted by KiCad using an arrow symbol, shows us that the silkscreen-layer C1 label for that capacitor overlaps with its silkscreen-layer circle outline. This, of course, poses no electrical liability for the layout, which is why KiCad categorizes it as a “warning” rather than an “error”, but it is still something we ought to fix for aesthetic reasons:



Simply moving the C1 label fixes this problem:

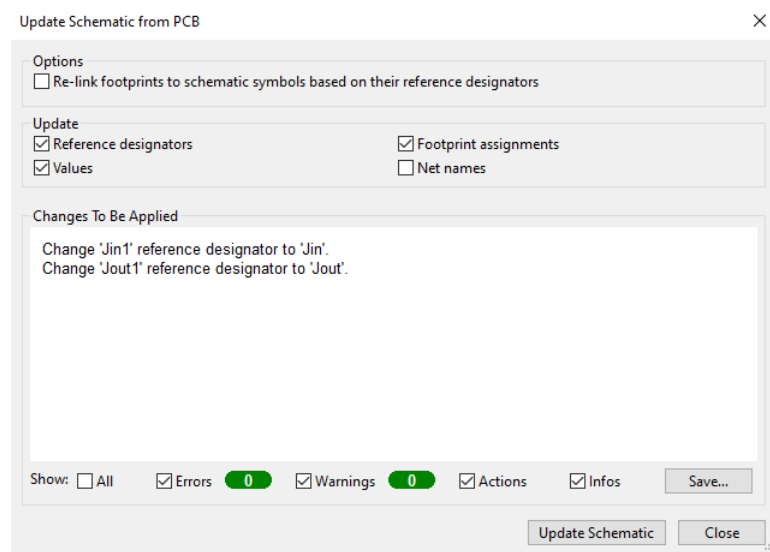


Now we turn our attention to the four schematic parity warnings, which all have to do with PCB footprints:

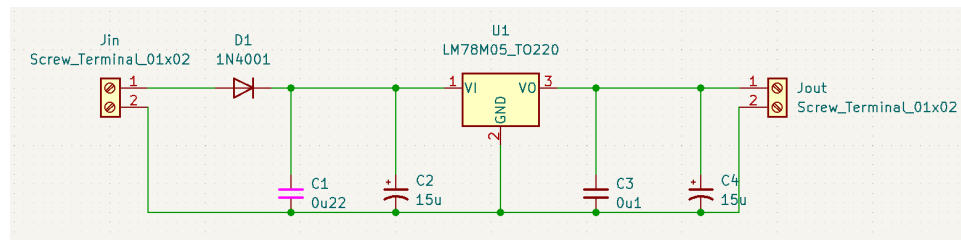


Here, KiCad's DRC report tells us there is a mis-match between the naming of the footprints for the input and output connectors. In the schematic editor these connectors bear the labels `Jin1` and `Jout1`, but in the PCB editor the labels are `Jin` and `Jout`. We need to harmonize these two labels to eliminate the parity warnings.

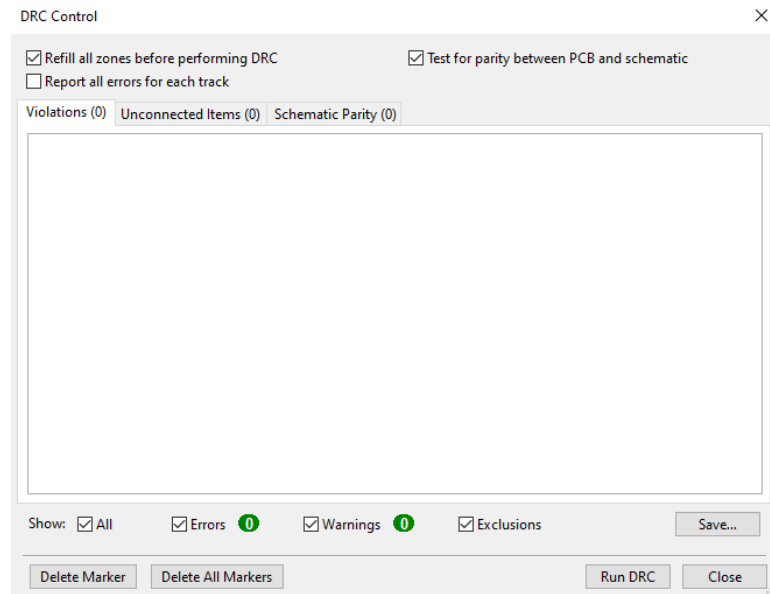
Fortunately, KiCad provides utilities for doing just that. We may choose either “Update PCB from Schematic” or “Update Schematic from PCB” depending on which set of labels we would prefer the layout to have. In this case, I chose the latter:



After this harmonization, we check the schematic editor again and see that the connectors have been re-labeled to **Jin** and **Jout**, respectively:



Running one more DRC report shows us that all warnings now are gone:



### 3.8.4 Plotting files using KiCad

Here are some special instructions requested by one board shop (JLCPCB) in December of 2022 regarding the use of KiCad version 6:

- In the “Plot” dialogue window
  - Include all copper layers, paste layers, silkscreen layers, and soldermask layers, as well as the edge cuts layer
  - Plot reference designators
  - Use Protel filename extensions
  - Subtract soldermask from silkscreen
  - Check zone fills before plotting
- In the “Generate Drill Files” dialogue window
  - Select Excellon drill file format
  - Select Gerber map file format
  - Drill origin should be absolute
  - Drill units should be in millimeters
  - Zeros format should be Decimal
- It is also recommended to generate a human-readable Map file as well to include in the final .zip compressed archive

Lesson from personal use: I neglected to follow all these steps on a PCB build request I submitted to JLCPCB on 1 December 2022, and one of the problems I encountered is that none of my mounting holes on the PCB had complete instructions to drill holes through the board at those locations! This was despite the fact that KiCad showed those mounting holes properly finished in its own 3D viewer. However, viewing the drill files (later) using separate Gerber-viewer software showed those mounting holes to actually lack drill marks. I was using KiCad version 6.0.7 on a personal computer running the Microsoft Windows operating system when this happened.





## Chapter 4

# Derivations and Technical References

This chapter is where you will find mathematical derivations too detailed to include in the tutorial, and/or tables and other technical reference material.

## 4.1 Trace ampacity ratings

These DC ampacity values for copper metal traces come from the February 1998 version of ANSI/IPC-2221 standard for generic printed circuit board design, assuming the standard copper thickness of 1 ounce (i.e. 1 oz. of copper per square foot of un-etched board surface, approximately 1.38 mils in thickness), for traces that are on the outside layer of a board exposed to ambient air, resulting in no more than a 10° Celsius temperature rise.

Trace width	Maximum current
15 mils = 0.38 mm	1 Ampere
25 mils = 0.64 mm	1.5 Amperes
40 mils = 1.0 mm	2.2 Amperes
55 mils = 1.4 mm	3 Amperes
85 mils = 2.2 mm	4 Amperes
125 mils = 3.2 mm	5 Amperes
150 mils = 3.8 mm	6 Amperes
190 mils = 4.8 mm	7 Amperes
225 mils = 5.7 mm	8 Amperes
260 mils = 6.6 mm	9 Amperes
300 mils = 7.6 mm	10 Amperes

Please note that this table shows conservative estimates only. For professional PCB design you should always consult the latest standard(s)! Alternatively, you may use this formula derived from later versions of ANSI/IPC-2221:

$$I = k(\Delta T)^{0.44}(1.38wz)^{0.725}$$

Where,

$I$  = Maximum allowable current (DC) in Amperes

$k$  = Coefficient related to trace exposure, 0.048 for external traces and 0.024 for internal

$\Delta T$  = Temperature rise above ambient in degrees Celsius

$w$  = Width of copper trace in mils

$z$  = Weight of PCB copper layer in ounces per square foot

Techniques for increasing ampacity for high-current applications include using parallel traces on multiple layers of the board, using copper planes rather than conventional traces, using thicker copper (e.g. 2 ounce per square foot rather than the customary 1 ounce per square foot), and eliminating soldermask over the high-current trace (causing solder to adhere to the bare-copper trace and thereby increase the thickness of metal available to conduct current).

## 4.2 Trace voltage ratings

These minimum spacing values between conductors come from the February 1998 version of ANSI/IPC-2221 standard for generic printed circuit board design for traces covered by polymer soldermask, and are shown here as an approximate guide only. For professional PCB design you should always consult the latest standard(s)!

Voltage	Minimum spacing
0 to 30 Volts	0.05 mm $\approx$ 2 mils
31 to 100 Volts	0.13 mm $\approx$ 5 mils
101 to 300 Volts	0.4 mm $\approx$ 16 mils
301 to 800 Volts	0.8 mm $\approx$ 31.5 mils
Over 500 Volts	Additional 0.00305 mm per Volt over 500 V

For example, if two soldermask-coated parallel traces on a PCB must withstand peak voltages of 800 Volts, the minimum spacing between those two traces must be 0.8 millimeters *plus*  $(800 - 500)(0.00305)$  millimeters for a total minimum spacing of 1.715 millimeters.

Uncoated (bare metal) traces require much greater spacing due to the lack of any electrically-insulating polymer coating to withstand the electric field posed by a potential difference between them. Also, for bare metal traces there is an *altitude* correction: the higher the altitude the PCB will operate at, the less dense the atmosphere will be, and air at lower densities has a lower breakdown voltage rating than air at sea level. This is important to consider for PCBs used in aircraft or spacecraft. Also, consider that even with soldermask-coated traces, there will be bare-metal conductor surfaces in the form of component terminals where this altitude correction applies. For additional voltage tolerance, soldered PCB assemblies may be *conformally coated*, that final layer serving as additional electrical insulation.

### 4.3 Tony's PCB layout recommendations

The following are recommendations for sizing of PCB traces, holes, etc. based on personal practice:

- Signal trace width = 10 to 20 mils
- Power trace width = 25 mils
- “Fat” trace width = 60 mils
- Through-hole resistors (e.g. 1/4 Watt carbon-composition and metal-film resistors)
  - Pad-to-pad spacing = 400 mils
  - Pad diameter = 60 mils
  - Hole drill size for through-plated holes = 28 mils or 0.8 millimeters
- Through-hole power components (e.g. 1N400x rectifying diodes, LEDs, TO-220 device terminals)
  - Pad diameter = 60 mils
  - Hole drill size for through-plated holes = 35 mils or 0.9 millimeters
- Through-hole headers
  - Pad diameter = 70 mils
  - Hole drill size for through-plated holes = 38 mils or 1 millimeter
- #4-40 machine screw holes for mechanical mounting
  - Pad diameter = 200 mils
  - Hole drill size for through-plated holes = 130 mils
- #6-32 machine screw holes for mechanical mounting
  - Pad diameter = 250 mils
  - Hole drill size for through-plated holes = 150 mils
- #6-32 machine screw holes for electrical terminals (e.g. ring lug connections to PCB)
  - Pad diameter = 300 mils
  - Hole drill size for through-plated holes = 150 mils

## 4.4 Tony's KiCad recommendations

One of the most popular schematic capture and PCB layout programs at the time of this writing (2025) is KiCad, and for good reason: it's both free and open-source, and it is widely supported with a large user base. What follows is a set of tips and recommendations I've found helpful especially for new users of KiCad. Note that these recommendations apply to version 8.0.4 of KiCad and *may* not be 100% accurate for newer versions.

### 4.4.1 Preliminary setup recommendations

These are steps to take prior to sketching schematic diagrams and laying out the actual PCB.

- Open the **Preferences** dialogue from any of the KiCad windows (Jain, Schematic Editor, or PCB Editor) and set up preferred grid spacings, colors, hotkeys, etc.
- Open the **Schematic Setup** dialogue from the File menu of the Schematic Editor and set up the preferred General parameters, Electrical Rules, and Project parameters (e.g. Netclasses for things like power rails and signal paths with special considerations such as heavy-current traces). Note that these settings may be imported from other KiCad projects!
- Open the **Board Setup** dialogue from the File menu of the PCB Editor and set up the intended Board Stackup (e.g. layers), Text & Graphics preferences, and Design Rule parameters. Note that these settings may be imported from other KiCad projects!
- An easy way to maintain all your personal design and editor preferences in one place is to create an “empty” project with a name like `my_defaults` where you set all these parameters there, and then for all new KiCad projects just import your default settings into the new project as a first step!

### 4.4.2 Finding components within KiCad’s libraries

KiCad offers a fairly large library of components that can be challenging to navigate. This list is intended to be a quick reference to some of the more common library components.

- The **Device** library item seems to be a catch-all location for a wide range of generic components including passive components (e.g. resistors, capacitors, inductors), electrical sources (e.g. batteries), meters, antennas, active components (e.g. transistors, thyristors), etc. Note that many similar components are located under other library items, but by part number. The “Device” item is for generic components.
- The **Power** library item contains power rail and ground symbols.
- The **Connector** library item contains signal test points, screw terminals, header pins, sockets, etc.
- **Digital logic ICs** are found under the library item corresponding to the device family, e.g. *4xxx*, *74xx*, etc. Note the existence of *IEEE* versions of these library items using the box-shaped component symbols that nobody ever uses.

Furthermore, the Symbol Editor contains a *Filter* field useful for searching. Note that one need not press “Enter” to complete a search query, but only begin typing for the search to activate. Pressing Enter prematurely will select the first result of the search.

### 4.4.3 Installing new schematic symbols and layout footprints

Like all modern PCB layout software packages, KiCad sports a fairly large library of components. This makes schematic capture and PCB layout much easier than if you had to custom-design each and every component prior to using it in a KiCad design! However, KiCad's library is not exhaustive, and so from time to time you will find yourself having to install new components or even create one from scratch.

An important distinction in the realm of KiCad is that a component's abstraction suited for use in KiCad's Schematic Editor is called a *Symbol*, whereas a component's abstraction suited for use in KiCad's PCB Editor is called a *Footprint*. Symbols appear in the Schematic Editor with pins numbers and functions, making it easy when sketching the diagram. Footprints appear in the PCB Editor with correct dimensions of the component package as well as terminals. It is critically important that these symbols are correct and properly mapped to suitable footprints, especially given the fact that one symbol may many possible footprints. Consider a component available in both through-hole and surface-mount packages, where a single KiCad symbol properly defines the number of pins, the designator (e.g. R for resistor), etc. while each possible footprint for that one component defines pad dimensions, spacing, silkscreen outline, etc.

Symbols and footprints within KiCad's library take the form of two different file types: KiCad symbol files ending with the filename extension `kicad_sym` and KiCad footprint files ending with the filename extension `kicad_mod`. An individual symbol file actually contains code describing multiple component symbols, which is why symbol files are typically named after a *class* or *group* of different components (e.g. `Connector.kicad_sym` containing code for *all* connector symbols). Footprint data is organized a bit differently, with classes or groups of footprints each having their own folder (directory) containing the word "pretty" (e.g. `Connector.pretty`) and then individual footprint files stored within the appropriate folder (e.g. `Banana_Jack_1pin.kicad_mod`, `CalTest_CT3151.kicad_mod`, and other connector footprints stored in the `Connector.pretty` folder).

Many manufacturers (e.g. Texas Instruments) and distributors (e.g. Digi-Key) provide symbols and footprints for download, often as compressed archive files you must extract the individual symbol files and footprint files from. For example, shopping for a component on Digi-Key might end with downloading a single compressed archive file containing one symbol file containing that component's symbol data, plus one footprint file containing that component's footprint data. Getting this third-party data into KiCad, however, is not as straight-forward as simply moving these files into their proper locations, especially in the case of symbols where each of KiCad's symbol files contains data for multiple components! The following steps will guide you through the symbol and footprint installation process.



### Installing a third-party component symbol

- Download the archive file from the manufacturer or from the distributor (usually a compressed zip file) then extract it to find the `kicad_sym` file within that archive. It is recommended you move the extracted files to a stable location on your computer's filesystem, but not directly within the nested directory structure used by KiCad's stock symbols and footprints so that nothing will be overwritten when you upgrade to a later version of KiCad. The next steps describe how to use KiCad to import this new symbol data into its library structure.
- If you haven't already, create a New Library for your imported symbols using the Symbol Editor within KiCad's Schematic Editor window (under the Tools menu). Make this new library "Global" so that it will be accessible for all projects (rather than being limited to just the project you happen to have open now) and then instruct KiCad to save this symbol file to the stable symbol folder (directory) you've made for yourself in the previous step. You can name this symbol file whatever you wish, and because KiCad symbol files (ending in `kicad_sym`) actually store multiple symbols, you should name it according to the type or class of symbol you intend to install. For example, if you are installing a third-party symbol for a model 74HC163 digital counter IC, you might want to rename this symbol file something like `MyCounters.kicad_sym` so as to hold future counter IC symbols you might import in the future.
- Again using the Symbol Editor tool, right-click the personal library you created that should now appear in the Libraries list, and select the Import Symbol option. This will open a file navigation window prompting you to find the symbol file you downloaded from the internet. KiCad will extract the component's symbol data from that downloaded file and append it to the contents of your personal library (which, again, is a single file in itself).
- Be sure to "Save" before exiting the Symbol Editor window. It will prompt you to save if you forget this step and tell it to close.
- When finished, the Symbol Editor's libraries list should contain an item of your naming (e.g. `MyCounters`) and under this will be the manufacturer's symbol with the appropriate part number. As you import more digital counter symbol data from various manufacturers and use KiCad to append this new data to your library item, that `MyCounters.kicad_sym` file you created will grow in size, and within the Symbol Editor you will be able to navigate that file's contents to show symbols for each of the different components contained within.

### Installing a third-party component footprint

- Download the archive file from the manufacturer or from the distributor (usually a compressed zip file) then extract it to find the `kicad_mod` file within that archive (typically found within a “pretty” subdirectory). It is recommended you move the extracted files to a stable location on your computer’s filesystem, but not directly within the nested directory structure used by KiCad’s stock symbols and footprints so that nothing will be overwritten when you upgrade to a later version of KiCad. The next steps describe how to use KiCad to import this new footprint data into its library structure.
- If you haven’t already, create a New Library for your imported footprints using the Footprint Editor within KiCad’s PCB Editor window (under the Tools menu). Make this new library “Global” so that it will be accessible for all projects (rather than being limited to just the project you happen to have open now) and then instruct KiCad to save this new directory to the stable footprint folder (directory) you’ve made for yourself in the previous step. You can name this footprint directory whatever you wish. For example, if you are installing a third-party footprint suitable for a model 74HC163 digital counter IC, you might want to rename this footprint directory something like `MyCounters.pretty` so as to hold future counter IC footprints you might import in the future.
- Again using the Footprint Editor tool, right-click the personal library you created that should now appear in the Libraries list, and select the Import Footprint option. This will open a file navigation window prompting you to find the footprint `.kicad_mod` file you downloaded from the internet.
- You will then need to “Save As” this new footprint file to the library of your choice, in this case the new library you just created previously. It will prompt you to save if you forget this step and tell it to close.
- When finished, the Footprint Editor’s libraries list should contain an item of your naming (e.g. `MyCounters`) and under this will be the manufacturer’s footprint with the appropriate part number. Also, if this footprint was imported along with its corresponding symbol from the manufacturer or distributor, the footprint will be mapped by default to that symbol when you insert that symbol into your schematic diagram.

### Creating a custom a component symbol

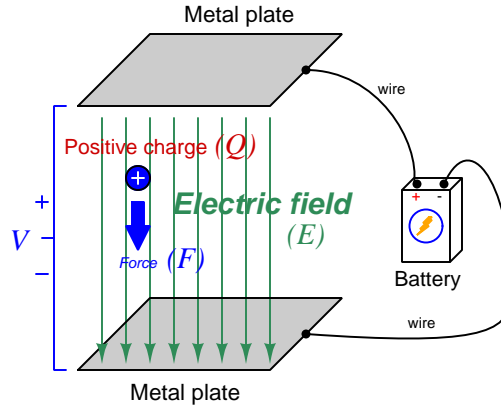
- **TO BE CONTINUED!**
- 
- 
-

Creating a custom a component footprint

- TO BE CONTINUED!
- 
- 
-

## 4.5 Electric field quantities

A useful definition of electric field ( $E$ ) is in terms of the force ( $F$ ) exerted on an electric charge ( $Q$ ) influenced by that field:



$$\vec{F} = Q\vec{E}$$

Where,

$\vec{F}$  = Force exerted on the charge (Newtons)

$Q$  = Charge quantity (Coulombs<sup>1</sup>)

$\vec{E}$  = Electric field (Newtons per Coulomb)

The small “arrow” symbols above the variables for force and electric field in the equation denote those variables as *vector quantities*, having both magnitude and direction. Charge is a *scalar quantity* having only magnitude but no direction, and as a scalar quantity when multiplied by the electric field vector it simply magnifies the magnitude but does not alter the direction. Therefore, the force and electric field vectors always point in the same direction.

Alternatively electric field may be defined in terms of the voltage between the end-points and the distance separating them, in which case we may express the electric field in units of *Volts per meter* as an alternative to *Newtons per Coulomb*:

$$\vec{E} = \frac{V}{d}$$

This measurement of electric field strength is very important for quantifying the *breakdown* of electrical insulators: the point at which the electric field becomes so powerful that otherwise immobile charges within the insulating substance are torn free to constitute a current and that substance is no longer an insulator. For rating the dielectric strength of insulating materials, we often see electric fields expressed in units of *kiloVolts per millimeter* rather than Volts per meter just to make the numerical quantities easier to manage (1 kV/mm = 1 *million* V/m).

<sup>1</sup>One Coulomb of electric charge is equal to  $6.2415 \times 10^{18}$  electrons.

The vector arrows shown in the previous illustration representing the electric field between two metal plates actually represent electric *flux* ( $\Phi_E$ ). The electric field ( $E$ ) is related to electric flux by area ( $A$ ), the field being a measurement of how densely-packed those flux lines are per unit area:

$$\vec{E} = \frac{\Phi_E}{\vec{A}}$$

Where,

$\vec{E}$  = Electric field, or electric flux density (Newtons per Coulomb)

$\Phi_E$  = Electric flux (Newton-meter squared per Coulomb)

$\vec{A}$  = Area over which flux is distributed (square meters)

The mere presence of an unbalanced electric charge at any point in space is sufficient to generate lines of electric flux, the total magnitude of that flux predicted by the following equation:

$$\Phi_E = \frac{Q}{\epsilon}$$

Where,

$\Phi_E$  = Electric flux (Newton-meter squared per Coulomb)

$Q$  = Charge quantity (Coulombs)

$\epsilon$  = Electric permittivity of the surrounding space (Coulombs squared per Newton-meter squared, approximately  $8.85 \times 10^{-12}$  for empty space)

By convention, these flux vectors point *away* from positive charges and point *toward* negative charges, their direction indicating force exerted on any positively-charged particle influenced by that field<sup>2</sup>. As the equation states, the amount of flux depends on how much charge exists at each particle as well as the permittivity of the surrounding space:



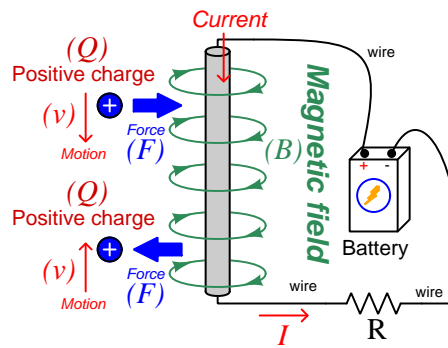
For example, identical charges suspended in a vacuum versus in a substance such as oil will have different amounts of flux associated with them as a result of oil and vacuum having different permittivity values. Perfectly empty space has the least amount of permittivity, which means anything else (gas, liquid, or solid matter) has greater  $\epsilon$  which acts to diminish the amount of electric flux surrounding any charged particle. Superconducting materials have infinite permittivity, which means they forbid the existence of any electric field inside their bulk.

---

<sup>2</sup>Conversely, the flux vectors point exactly opposite the direction of force applied to any *negatively*-charged particle within that field. This makes sense of the aphorism that *like charges repel and opposite charges attract*. If you consider the two charges shown in this illustration, the positive charge will be pulled in the direction of the flux vectors pointing toward the negative charge, as the negative charge will also be pulled opposite the direction of the flux vectors pointing away from the positive charge (i.e. the negative charge will be pulled *toward* the positive charge) – thus the positive and negative charges feel mutual attraction.

## 4.6 Magnetic field quantities

A useful definition of magnetic field ( $B$ ) is in terms of the force ( $F$ , called the *Lorentz force*) exerted on a moving electric charge ( $Q$ ) influenced by that field:



$$\vec{F} = Q\vec{v} \times \vec{B}$$

Where,

$\vec{F}$  = Force exerted on the charge (Newtons)

$Q$  = Charge quantity (Coulombs<sup>3</sup>)

$\vec{v}$  = Velocity of moving charge (meters per second)

$\vec{B}$  = Magnetic field (Tesla, Webers per square meter, or Newtons per Ampere-meter)

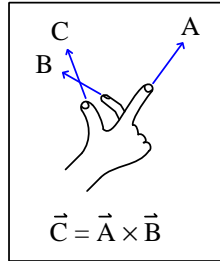
The small “arrow” symbols above the variables for force and velocity and magnetic field in the equation denote those variables as *vector quantities*, having both magnitude and direction. Charge is a *scalar quantity* having only magnitude but no direction, and as a scalar quantity when multiplied by the velocity vector it simply magnifies the magnitude but does not alter the direction. The “cross-product” ( $\times$ ) is a specific form of vector multiplication, and it results in a product at right angles to the vector directions of both terms. Therefore, the force and velocity and electric field vectors never all point in the same direction.

---

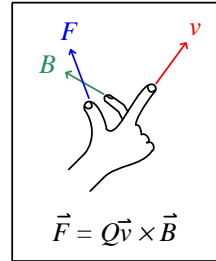
<sup>3</sup>One Coulomb of electric charge is equal to  $6.2415 \times 10^{18}$  electrons.

Vector cross-products conveniently relate to the fingers of the right hand, which is where the “right-hand rule” originates:

General "right-hand rule"  
for vector cross-products

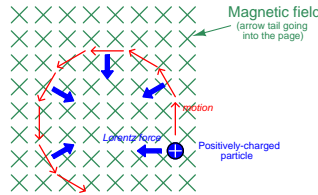


"Right-hand rule" specific  
to the Lorentz force



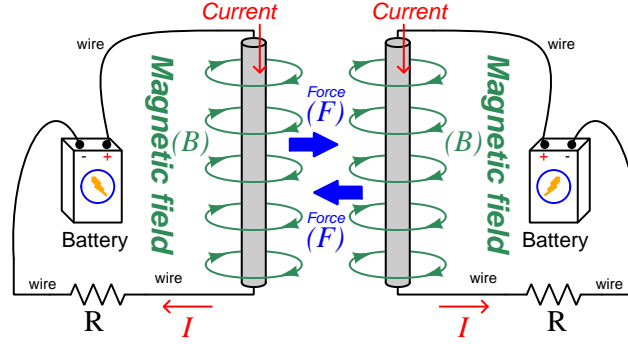
When holding the index finger, middle finger, and thumb of your right hand perpendicular to each other, your index finger points in the direction of the velocity vector ( $\vec{v}$ ), your middle finger in the direction of the magnetic field vector ( $\vec{B}$ ), and your thumb in the direction of the force vector ( $\vec{F}$ ). A simple mnemonic I use to remember these relationships of fingers to vectors is that the **I**ndex finger points in the direction of current<sup>4</sup> ( $I$ ), the **M**iddle finger points in the direction of the **m**agnetic field ( $B$ ), and the **T**humb points in the direction of the **t**hrust (i.e. force) acting upon the moving charge.

The Lorentz force’s effect on electrically-charged particles in motion has many applications, from redirecting the paths of charged-particle beams in particle accelerator machines, to bending the trajectory of electron beams in a cathode-ray tube (CRT), to forcing electrons to travel in spiral or circular trajectories inside of magnetron (microwave oscillator) tubes. An illustration of a positively-charged particle curving perpendicular to a magnetic field appears here:



<sup>4</sup>Some textbooks speak of a “left-hand rule” which is intended to make sense of electric charge motion (current) in terms of *electron flow*. As we know, electrons are the only real mobile charge carriers within metal conductors, and so technically “electron flow” notation is most physically accurate when describing the motion of electric charges in metallic circuits. However, the right-hand rule is a *mathematical definition* for vector cross products, the concept of the cross product arising in the late 18th century when electrical science was still in its infancy. Early explorers of electricity used the established mathematical tools of their time and applied it to their work with electric currents and magnetism. At that time, charge carriers in metal wires were assumed to be “positive” and this is how the motion of positively-charged carriers became associated with the first vector of the cross-product. As a result of this assumption which was later proven false, we have two different conventions for denoting the motion of electricity: electron-flow which is *physically* accurate (for metal wires, at least), and conventional flow which is *mathematically* accurate. This, perhaps more than any other reason, is why educational programs designed for mathematically rigorous fields (e.g. electrical engineering) exclusively use conventional flow notation rather than electron flow notation to denote the direction of current.

If the moving charge in question is not a single charged particle but rather part an electric *current* passing through a conductor parallel to the first, both conductors will experience a mutually-attracting force given by the following equation:



$$\vec{F} = I\vec{l} \times \vec{B}$$

Where,

$\vec{F}$  = Force exerted on both conductors (Newtons)

$I$  = Current (Amperes)

$\vec{l}$  = Length of wire (meters)

$\vec{B}$  = Magnetic field (Tesla, or Webers per square meter, or Newtons per Ampere-meter)

The point-charge Lorentz force equation and the two-conductor Lorentz force equation are not that different from one another. Dimensional analysis validates this: the Lorentz force on a moving charge uses that charge quantity (Coulombs) multiplied by the point-charge's velocity in meters per second to give Coulomb-meters per second for the first term:

$$Q\vec{v} = [\text{C}] \left[ \frac{\text{m}}{\text{s}} \right] = \left[ \frac{\text{C} \cdot \text{m}}{\text{s}} \right]$$

The Lorentz force on a current-carrying conductor uses the current (Amperes, which is Coulombs per second) multiplied by length in meters, for the same composite units of Coulomb-meters per second:

$$I\vec{l} = \left[ \frac{\text{C}}{\text{s}} \right] [\text{m}] = \left[ \frac{\text{C} \cdot \text{m}}{\text{s}} \right]$$

This dimensional equivalence makes conceptual sense as well: an electrically-charged particle moving through empty space is an electric current in its own right, and an electric current flowing through a conductor is just a collection of charged particles moving through space (just not *empty* space). In either case, the basis for the Lorentz force remains the same: the moving charge(s) create their own magnetic field, which reacts with the magnetic field of the original current-carrying wire to produce forces acting on both.

If the two currents flow in the same direction, their mutual forces *attract*. If the two currents flow in opposite directions, their mutual forces *repel*. This is the basis of electric motors: causing



mechanical motion by electro-magnetic attraction and repulsion. It also represents an interesting contrast with electric fields:

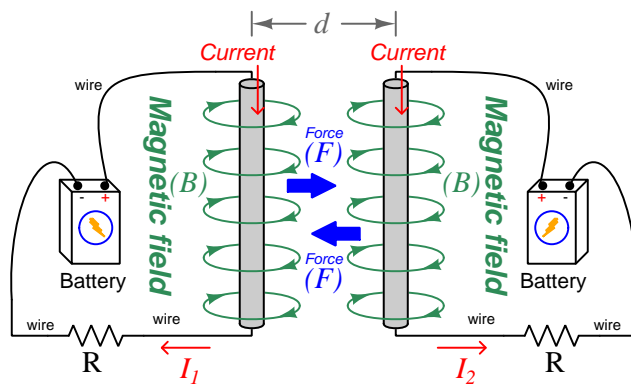
With electric fields, opposite *charges* attract and like *charges* repel.

With magnetic fields, opposite *poles* attract and like *poles* repel.

With parallel currents, opposite *directions* repel and like *directions* attract<sup>5</sup>.

Two parallel current-carrying conductors of length  $l$  and separated by a distance  $d$  will generate a mutual force proportional to both their currents:

$$F = l \frac{\mu I_1 I_2}{2\pi d}$$



<sup>5</sup>That is, assuming it's *like* charges moving in these directions! If the charges in question are opposite each other – for example electrons in one circuit and holes in another – then like directions will repel and opposite directions will attract!

The circular loops surrounding the current-carrying conductors in the previous illustrations represent the magnetic lines of *flux* ( $\Phi_B$ ) surrounding each of those conductors. The magnetic field ( $B$ ) is related to magnetic flux by area ( $A$ ), the field being a measurement of how densely-packed those flux lines are per unit area. For this reason, magnetic field ( $B$ ) is more properly known as *magnetic flux density*:

$$\vec{B} = \frac{\Phi_B}{\vec{A}}$$

Where,

$\vec{B}$  = Magnetic field or flux density (Tesla, Webers per square meter, or Newtons per Ampere-meter)

$\Phi_B$  = Magnetic flux (Webers)

$\vec{A}$  = Area over which flux is distributed (square meters)

An older unit of measurement for magnetic flux density  $B$  is the *Gauss* which is much smaller than a Tesla, with one Tesla equivalent to 10,000 Gauss. To put things into perspective, the Earth's natural magnetic field has a strength of approximately one-half of one Gauss<sup>6</sup>.

Magnetic field strength is an inverse function of distance from any current-carrying wire, and also depends on the magnetic permeability of the space adjacent to the wire:

$$B = \frac{\mu I}{2\pi d}$$

Where,

$B$  = Magnetic field or flux density (Tesla, Webers per square meter, or Newtons per Ampere-meter)

$\mu$  = Magnetic permeability of the surrounding space (Tesla-meters per Ampere,  $4\pi \times 10^{-7}$  for empty space)

$I$  = Current (Amperes)

$d$  = Distance from conductor (meters)

---

<sup>6</sup>Using the online *Magnetic Field Calculator* application provided by NOAA (the National Oceanic and Atmospheric Administration) at <https://ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm>, applying the World Magnetic Model WMM modeling algorithm for years 2019-2024, the total magnetic field strength at my home is 53,584.4 nano-Tesla (53,584.4 nT or 0.535844 Gauss), and presently (May 2020) decaying at a rate of  $-104.1$  nT per year.

The relation of magnetic *flux* to current through a conductor follows a similar equation:

$$\Phi = \frac{\mu AI}{2\pi d}$$

Where,

$\Phi$  = Magnetic flux (Webers)

$\mu$  = Magnetic permeability of the surrounding space (Tesla-meters per Ampere,  $4\pi \times 10^{-7}$  for empty space)

$A$  = Area over which flux is distributed (square meters)

$I$  = Current (Amperes)

$d$  = Distance from conductor (meters)

As this equation makes clear, the amount of magnetic flux surrounding a current-carrying conductor depends not only on the amount of current, but also on the sampled area, the distance from the wire, and also the surrounding material. Most<sup>7</sup> substances (gas, liquid, solid) have permeability values greater than that of empty space, and so this means magnetic flux is usually *enhanced* by the presence of matter around the current-carrying conductor.

The total magnetic flux enclosed by a circular wire loop follows a similar equation:

$$\Phi = \frac{\pi\mu Ir}{2}$$

Where,

$\Phi$  = Magnetic flux (Webers)

$\mu$  = Magnetic permeability of the surrounding space (Tesla-meters per Ampere,  $4\pi \times 10^{-7}$  for empty space)

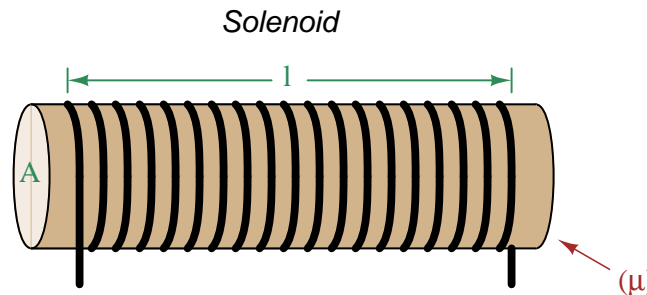
$I$  = Current (Amperes)

$r$  = Radius of circular loop (meters)

---

<sup>7</sup>Interestingly, superconducting materials *forbid* magnetic fields inside of their bulk, and so the permeability value of any superconductor must be zero!

A common form of electromagnet known as a *solenoid* takes the form of a wire coil wrapped in such a way as to form a long<sup>8</sup> cylinder, often wrapped around a plastic frame, and often with a ferromagnetic material such as iron in the center:



The amount of magnetic flux, and the flux density, within the interior of a current-carrying solenoid are given by the following formulae:

$$\Phi = \frac{\mu N A I}{l} \qquad B = \frac{\mu N I}{l}$$

Where,

$\Phi$  = Magnetic flux (Webers)

$B$  = Magnetic field or flux density (Tesla, Webers per square meter, or Newtons per Ampere-meter)

$\mu$  = Magnetic permeability of the surrounding space (Tesla-meters per Ampere,  $4\pi \times 10^{-7}$  for empty space)

$N$  = Number of turns of wire in the coil

$A$  = Cross-sectional area of solenoid coil (square meters)

$I$  = Current (Amperes)

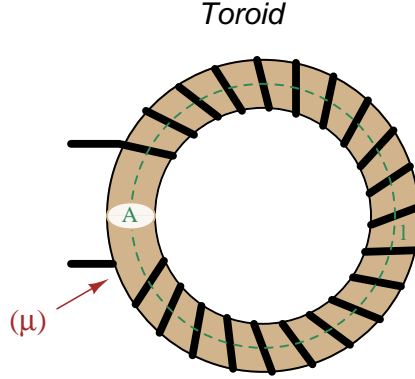
$l$  = Length of solenoid coil (meters)

These formulae have interesting implications for solenoid design. Note how a shorter (i.e. smaller length  $l$ ) solenoid identical in all other respects will generate a stronger magnetic field for a given current. Note also how the flux density ( $B$ ) remains constant with increasing cross-sectional area ( $A$ ) if all other factors are equal, and that this necessarily means a greater amount of total magnetic flux ( $\Phi$ ) for a greater area  $A$ .

---

<sup>8</sup>These magnetic field formulae apply perfectly to a solenoid coil that is closely-packed (i.e. each turn adjacent to the next) and infinitely long. Therefore, they only approximate real solenoid behavior. This fact may be understood by performing a thought experiment where we decrease the solenoid coil's length to zero, in which case the formulae predict an *infinite* amount of magnetism for any amount of current at all, which of course cannot be true.

Another common form of electromagnet known as a *toroid* is really just a solenoid bent in a circle so that its two ends meet<sup>9</sup> cylinder, often wrapped around a plastic frame, and often with a ferromagnetic material such as iron in the center. Toroids have the unusual property of *containing* their magnetic flux lines extremely well, unlike solenoids, wires, and simple coils which all radiate magnetic fields. They find application as energy-storage devices, or as electromagnets suitable for applying magnetic fields to specimens placed *inside* the toroid's cross-section:



The amount of magnetic flux, and the flux density, within the interior of a current-carrying toroid are identical to that within an otherwise identical solenoid having a length ( $l$ ) equal to the toroid's circumference:

$$\Phi = \frac{\mu N A I}{l} \qquad B = \frac{\mu N I}{l}$$

Where,

$\Phi$  = Magnetic flux (Webers)

$B$  = Magnetic field or flux density (Tesla, Webers per square meter, or Newtons per Ampere-meter)

$\mu$  = Magnetic permeability of the surrounding space (Tesla-meters per Ampere,  $4\pi \times 10^{-7}$  for empty space)

$N$  = Number of turns of wire in the coil

$A$  = Cross-sectional area of toroid (square meters)

$I$  = Current (Amperes)

$l$  = Circumference of toroid (meters)

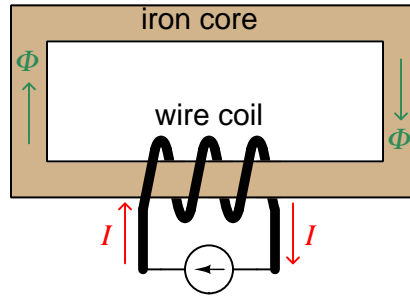
If we wish to substitute toroid radius ( $r$ ) for circumferential length ( $l$ ), the formulae become the following:

$$\Phi = \frac{\mu N A I}{2\pi r} \qquad B = \frac{\mu N I}{2\pi r}$$

---

<sup>9</sup>Again, the magnetic field formulae are only accurate for a toroidal coil that is closely-packed (i.e. each turn adjacent to the next) and infinitely long, and therefore only approximate real toroid behavior. This fact may be understood by performing an equivalent thought experiment as before where we decrease the toroid's circumference to zero and absurdly end up with *infinite* magnetism for a finite current.

Many applications of electromagnetism involve conductive *coils* wrapped around some form of ferromagnetic core material, the purpose of that core being to provide a higher-permeability pathway for the magnetic flux than would exist otherwise through air, and the purpose of the wire coil being to intensify the amount of magnetism developed by the electric current beyond what would be possible with a straight current-carrying wire. These magnetic cores typically form a closed loop, or *magnetic circuit* for the lines of magnetic flux to naturally form a closed path. A simple example appears here:



The amount of magnetic flux ( $\Phi$ ) present in the magnetic “circuit” formed by the iron core depends on many factors. First and foremost is the amount of electric current (in Amperes) passing through the wire coil and the number of turns that coil makes around the iron core. The product of this current and the number of turns is called the *magnetomotive force* or *mmf* of the magnetic circuit, analogous to “electromotive force” or “emf” often used as a synonym for voltage in an electric circuit. Not surprisingly, the standard metric unit of measurement for magnetomotive force is the *Ampere-turn*.

However, magnetomotive force alone does not fully describe the current’s effect on magnetism within the iron core. The total length of the magnetic circuit is also an important factor, since a longer path distributes that magnetomotive force over a greater distance. The quotient of magnetomotive force and magnetic circuit length is called *magnetic field intensity*, symbolized by the variable  $H$  and expressed in units of *Ampere-turns per meter*.

Magnetic permeability ( $\mu$ ) relates magnetic field intensity ( $H$ ) to the magnetic flux density ( $B$ ) within the core material, such that a greater permeability will result in higher flux density for any given amount of field intensity. Permeability is a property of the core material and not its geometry, mathematically defined as the ratio of flux density to field intensity:  $\mu = \frac{B}{H}$

Magnetic reluctance ( $\mathfrak{R}$ ) relates magnetomotive force (mmf) to magnetic flux ( $\Phi$ ), and is related not only to the core material’s permeability but also its geometry. It is mathematically defined as the ratio of magnetomotive force to magnetic flux:  $\mathfrak{R} = \frac{\text{mmf}}{\Phi}$

If all this seems confusing, you are in good company. Not only are there many magnetic variables, some related to physical geometry and others not, but there are *two different sets of metric units* appropriate for expressing each! The older units were based on the centimeter-gram-second (CGS) version of the metric system, while the newer units are based on the meter-kilogram-second or SI (Système International) version of the metric system.

Quantity	Symbol	SI unit	CGS unit
Magnetomotive force	mmf	Ampere-turn (A-t)	Gilbert (Gb)
Flux	$\Phi$	Weber (Wb)	Maxwell (Mx)
Field intensity	$H$	Ampere-turns per meter (A-t/m)	Oersted (Oe)
Flux density	$B$	Tesla (T)	Gauss (G)
Permeability	$\mu$	Tesla-meters per Ampere-turn (T-m/A-t)	Gauss per Oersted (G/Oe)
Reluctance	$\mathfrak{R}$	Ampere-turns per Weber (A-t/Wb)	Gilberts per Maxwell (G/Mx)

Magnetomotive force (mmf) and magnetic flux ( $\Phi$ ) may be thought of as the “raw” measures of magnetism, with Ampere-turns and Webers being their respective SI metric units. Reluctance ( $\mathfrak{R}$ ) is the ratio of the two for any given magnetic circuit with known dimensions and core material. Simply put, reluctance tells you how many Ampere-turns of magnetomotive force will be necessary to create one Weber of magnetic flux in a given space.

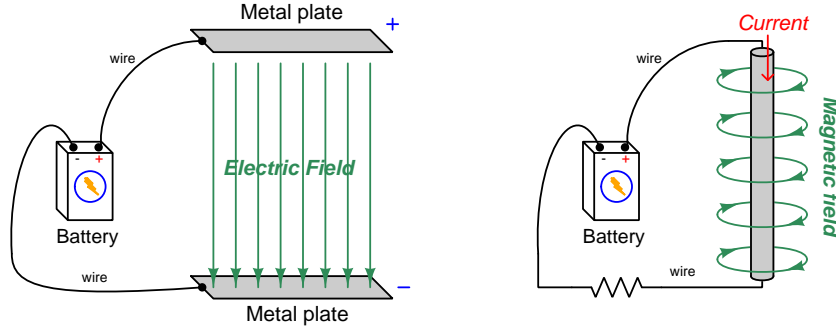
Magnetic field intensity ( $H$ ) and magnetic flux density ( $B$ ) may be thought of as the “normalized” measures of magnetism, with Ampere-turns per meter and Tesla being their respective SI metric units.  $H$  and  $B$  relate to mmf and flux by the physical dimensions of the magnetic circuit (length and cross-sectional area, respectively). Permeability is the ratio of the two for any given magnetic core material. Simply put, permeability tells you how many Tesla of magnetic field (i.e. flux density, or Webers of flux per square meter or cross-sectional core area) you will obtain for one Ampere-turn per meter of magnetic field intensity applied to a given core material.

Conversion between the newer SI and the older CGS metric units are as follows:

Quantity	Conversion equivalence
Magnetomotive force (mmf)	1 Ampere-turn = $\frac{4\pi}{10}$ Gilberts
Magnetic flux ( $\Phi$ )	1 Weber = $10^8$ Maxwells
Magnetic field intensity ( $H$ )	1 Ampere-turn/meter = $\frac{4\pi}{1000}$ Oersteds
Magnetic flux density ( $B$ )	1 Tesla = $10^4$ Gauss
Permeability ( $\mu$ )	1 Tesla-meter/Ampere-turn = $\frac{10^7}{4\pi}$ Gauss/Oersteds
Reluctance ( $\mathfrak{R}$ )	1 Ampere-turn/Weber = $\frac{4\pi}{10^9}$ Gilberts/Maxwell

## 4.7 Near-field versus far-field regions

An electric field exists wherever voltage exists (i.e. wherever an imbalance exists between positive and negative electrical charges), parallel to the axis of that voltage. A magnetic field exists wherever an electric charge moves, perpendicular to the axis of that charge's motion. These phenomena exist for DC (direct) as well as AC (alternating) electricity, and are shown in the two following illustrations:



Electric and magnetic fields, however, are not simply manifestations of voltage and current, respectively. Mathematical discoveries made by the Scottish physicist James Clerk Maxwell (and later simplified by Oliver Heaviside) also relate electric fields and magnetic fields *directly to each other*. Consider two<sup>10</sup> of Maxwell's equations shown below:

$$\oint \vec{E} \cdot d\vec{s} = -\frac{d\Phi_B}{dt}$$

$$\oint \vec{B} \cdot d\vec{s} = \mu_0 I + \mu_0 \epsilon_0 \frac{d\Phi_E}{dt}$$

The first equation describes how an electric field ( $\vec{E}$ ) is created by a varying magnetic flux ( $\frac{d\Phi_B}{dt}$ ), which is otherwise known as *Faraday's Law of Electromagnetic Induction*. This is the law exploited in the function of electro-mechanical generators whereby coils of wire are subjected to changing magnetic fields, creating voltage between the coils' end-points. The negative sign in this equation is an embodiment of *Lenz's Law*, which states that any current resulting from the induced voltage will produce its own magnetic field opposing the first magnetic field's direction of change.

The second equation shown here describes *two* different ways to produce a magnetic field ( $\vec{B}$ ). One way is to use a moving stream of electric charges known as a *current* ( $I$ ), a fact also known as *Ampère's Law*. This is the law exploited in the function of electromagnets, where we produce a magnetic field by connecting a coil of wire to an electrical source. The second way is to vary an electric flux ( $\frac{d\Phi_E}{dt}$ ) in empty space with surrounding magnetic permeability  $\mu_0$  and electric permittivity  $\epsilon_0$ .

<sup>10</sup>Maxwell's equations of electromagnetism are typically presented as a set of *four*. Here we only list two of these, as the other two are not relevant to electromagnetic waves in particular.

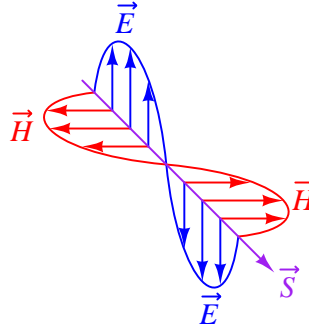


Maxwell's equations suggest that in pulsed DC and AC circuits, where the resulting electric and/or magnetic fields vary over time ( $\frac{d}{dt}$ ) rather than holding steady, those time-varying fields will mutually support one another over space and time. That is, a varying electric field will eventually produce a varying magnetic field, and vice-versa, those fields propagating one another as an *electromagnetic wave* radiating away from the circuit through space at the speed of light<sup>11</sup>.

At close range, immediately adjacent to the circuit conductors, electric and magnetic fields are strictly functions of those conductors' voltages and currents, respectively. These fields' shapes follow the basic forms shown in the previous illustrations: magnetic field lines circling current-carrying conductors and electric field lines stretching between conductive surfaces. We generally refer to this region of space around an energized circuit as the *near-field region*, where electric and magnetic fields are distinct from one another in form and in relative magnitude.

At farther distances from the energized circuit, the electric and magnetic fields are found in pairs oscillating at right angles to one another in space, forming a single electromagnetic wave radiating away from the source circuit spherically in all directions. This form of wave is a direct consequence of Maxwell's equations describing how the rate-of-change of one field creates the other field, and vice-versa. Below we see a crude representation of an electromagnetic wave, with an electric  $\vec{E}$  field, a magnetic  $\vec{H}$  field, and the vector of propagation  $\vec{S}$  called the *Poynting vector*:

*Electromagnetic wave*



This region where the electric and magnetic fields exist as coupled pairs comprising a well-formed wave is referred to as the *far-field region*. In this region the electric and magnetic fields are always perpendicular to one another and their relative magnitudes are in a fixed proportion governed by the permeability and permittivity of free space.

The demarcation between near- and far-field regions depends on the physical dimensions of the radiating circuit as well as the *wavelength* of the signal ( $\lambda$ ), wavelength being calculated by dividing the speed of light ( $2.9979 \times 10^8$  meters per second in empty space) by the signal frequency in Hertz. There is no simple rule for predicting where the near-field region ends and the far-field region begins, and in fact there is a gradual morphing of one to the other, but one may reliably consider any distance from the circuit in excess of multiple wavelengths to be far-field.

<sup>11</sup>In fact, this is precisely what light is: an electromagnetic wave of exceptionally high frequency, well beyond the frequency range of contemporary electric circuits.

The following list compares and contrasts several near-field versus far-field characteristics:

- In the near-field region electric and magnetic field strengths depend greatly on the geometry of the energized circuit's conductors as well as on the specific voltage and current levels existing at each location. In the far-field region, however, the electric and magnetic field strengths are always in fixed proportion to one another.
- Far-field effects may be ignored for low-frequency AC circuits because these circuits' wavelengths are so long. At 60 Hz, for example, the wavelength is nearly 5 *million* meters. If we consider the far-field region of a circuit to begin at least multiple wavelengths from the source, this puts the far-field region of a 60 Hz circuit at least one-quarter of the Earth's circumference away from the circuit in question!
- In the near-field region the proportionality between electric and magnetic field strengths is a function of circuit impedance: high-impedance circuits will have stronger electric fields than magnetic fields, and low-impedance circuits will have stronger magnetic fields than electric fields, all other factors being equal. This is due to Ohm's Law establishing the relationship between voltage and current ( $Z = \frac{V}{I}$ ), and how electric fields originate in a circuit from potential differences while magnetic fields originate in a circuit from currents. In the far-field region, however, where the electric and magnetic fields exist only through mutual support, their strengths are always in the same (fixed) proportion defined as the *characteristic impedance of free space* which is approximately 377 Ohms. This value is equal to the square root of the ratio of magnetic permeability to electric permittivity for free space ( $Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}}$ ) because these parameters of space dictate just how strongly a varying electric field creates a magnetic field.
- The relationship between field strength and physical distance from the circuit is a very complex one within the near-field region, as this depends greatly on the geometry of the circuit conductors. In the far-field region, however, we find the *Inverse-Square Law* always holds true: as distance from the circuit increases, electromagnetic wave strength simply and reliably diminishes with the *square* of that distance as if the wave were radiating away in a spherical fashion from a point-source<sup>12</sup>. For example, if we double the distance from the radiating circuit (from a point in the far-field region to another point in the far-field region twice as far away in empty space), the power conveyed by the electromagnetic radiation will always be *four times* less. Tripling the far-field distance always weakens the wave's power by a factor of *nine*. This is simply because the area over which the radiated energy spreads increases with the square of the distance from any point-source, viewing that point-source as the center of a sphere.
- In the far-field region every radiating circuit has a definite *radiation pattern* where the strength of the radiated electromagnetic wave as a function of the *angle* from the geometric centerline follows a consistent pattern regardless of distance. This is a very important feature of *antennas*, where different antenna designs feature unique far-field radiation patterns. By contrast, in the near-field region of an antenna the radiation pattern varies significantly with distance.

---

<sup>12</sup>A "point source" is a hypothetical point in space having no height nor width nor depth (i.e. it has *zero* physical dimensions) that emits radiation. Point-sources are a theoretical concept only. Real sources have spatial dimensions which makes their near-field strength/distance relationships complex, but the farther away you get from them the more their radiative behavior approaches that of a theoretical point-source.

- The spatial *polarization* of the electromagnetic wave (i.e. the orientation of its perpendicular electric and magnetic fields) is always well-defined in the far-field region but often ill-defined in the near-field region. This means, among other things, that two or more antennas will exchange electromagnetic energy efficiently only if they are appropriately oriented to one another over far-field distances, but may exchange energy fairly well regardless of orientation over near-field distances.
- Any “gain” specifications for an antenna structure apply only to apparent gains in signal power over the far-field range, because they refer to the degree to which an antenna focuses its energy in one direction more than another (i.e. its directionality), implying a distance over which the electromagnetic wave has become well-formed and the antenna’s radiation pattern is reliably established. At near-field distances these “gain” figures are meaningless.

## Chapter 5

# Questions

This learning module, along with all others in the ModEL collection, is designed to be used in an inverted instructional environment where students independently read<sup>1</sup> the tutorials and attempt to answer questions on their own *prior* to the instructor's interaction with them. In place of lecture<sup>2</sup>, the instructor engages with students in Socratic-style dialogue, probing and challenging their understanding of the subject matter through inquiry.

Answers are not provided for questions within this chapter, and this is by design. Solved problems may be found in the Tutorial and Derivation chapters, instead. The goal here is *independence*, and this requires students to be challenged in ways where others cannot think for them. Remember that you always have the tools of *experimentation* and *computer simulation* (e.g. SPICE) to explore concepts!

The following lists contain ideas for Socratic-style questions and challenges. Upon inspection, one will notice a strong theme of *metacognition* within these statements: they are designed to foster a regular habit of examining one's own thoughts as a means toward clearer thinking. As such these sample questions are useful both for instructor-led discussions as well as for self-study.

---

<sup>1</sup>Technical reading is an essential academic skill for any technical practitioner to possess for the simple reason that the most comprehensive, accurate, and useful information to be found for developing technical competence is in textual form. Technical careers in general are characterized by the need for continuous learning to remain current with standards and technology, and therefore any technical practitioner who cannot read well is handicapped in their professional development. An excellent resource for educators on improving students' reading prowess through intentional effort and strategy is the book *Reading For Understanding – How Reading Apprenticeship Improves Disciplinary Learning in Secondary and College Classrooms* by Ruth Schoenbach, Cynthia Greenleaf, and Lynn Murphy.

<sup>2</sup>Lecture is popular as a teaching method because it is easy to implement: any reasonably articulate subject matter expert can talk to students, even with little preparation. However, it is also quite problematic. A good lecture always makes complicated concepts seem easier than they are, which is bad for students because it instills a false sense of confidence in their own understanding; reading and re-articulation requires more cognitive effort and serves to verify comprehension. A culture of teaching-by-lecture fosters a debilitating dependence upon direct personal instruction, whereas the challenges of modern life demand independent and critical thought made possible only by gathering information and perspectives from afar. Information presented in a lecture is ephemeral, easily lost to failures of memory and dictation; text is forever, and may be referenced at any time.

## GENERAL CHALLENGES FOLLOWING TUTORIAL READING

- Summarize as much of the text as you can in one paragraph of your own words. A helpful strategy is to explain ideas as you would for an intelligent child: as simple as you can without compromising too much accuracy.
- Simplify a particular section of the text, for example a paragraph or even a single sentence, so as to capture the same fundamental idea in fewer words.
- Where did the text make the most sense to you? What was it about the text's presentation that made it clear?
- Identify where it might be easy for someone to misunderstand the text, and explain why you think it could be confusing.
- Identify any new concept(s) presented in the text, and explain in your own words.
- Identify any familiar concept(s) such as physical laws or principles applied or referenced in the text.
- Devise a proof of concept experiment demonstrating an important principle, physical law, or technical innovation represented in the text.
- Devise an experiment to disprove a plausible misconception.
- Did the text reveal any misconceptions you might have harbored? If so, describe the misconception(s) and the reason(s) why you now know them to be incorrect.
- Describe any useful problem-solving strategies applied in the text.
- Devise a question of your own to challenge a reader's comprehension of the text.

## GENERAL FOLLOW-UP CHALLENGES FOR ASSIGNED PROBLEMS

- Identify where any fundamental laws or principles apply to the solution of this problem, especially before applying any mathematical techniques.
- Devise a thought experiment to explore the characteristics of the problem scenario, applying known laws and principles to mentally model its behavior.
- Describe in detail your own strategy for solving this problem. How did you identify and organized the given information? Did you sketch any diagrams to help frame the problem?
- Is there more than one way to solve this problem? Which method seems best to you?
- Show the work you did in solving this problem, even if the solution is incomplete or incorrect.
- What would you say was the most challenging part of this problem, and why was it so?
- Was any important information missing from the problem which you had to research or recall?
- Was there any extraneous information presented within this problem? If so, what was it and why did it not matter?
- Examine someone else's solution to identify where they applied fundamental laws or principles.
- Simplify the problem from its given form and show how to solve this simpler version of it. Examples include eliminating certain variables or conditions, altering values to simpler (usually whole) numbers, applying a limiting case (i.e. altering a variable to some extreme or ultimate value).
- For quantitative problems, identify the real-world meaning of all intermediate calculations: their units of measurement, where they fit into the scenario at hand. Annotate any diagrams or illustrations with these calculated values.
- For quantitative problems, try approaching it qualitatively instead, thinking in terms of “increase” and “decrease” rather than definite values.
- For qualitative problems, try approaching it quantitatively instead, proposing simple numerical values for the variables.
- Were there any assumptions you made while solving this problem? Would your solution change if one of those assumptions were altered?
- Identify where it would be easy for someone to go astray in attempting to solve this problem.
- Formulate your own problem based on what you learned solving this one.

## GENERAL FOLLOW-UP CHALLENGES FOR EXPERIMENTS OR PROJECTS

- In what way(s) was this experiment or project easy to complete?
- Identify some of the challenges you faced in completing this experiment or project.

- Show how thorough documentation assisted in the completion of this experiment or project.
- Which fundamental laws or principles are key to this system's function?
- Identify any way(s) in which one might obtain false or otherwise misleading measurements from test equipment in this system.
- What will happen if (component  $X$ ) fails (open/shorted/etc.)?
- What would have to occur to make this system unsafe?

## 5.1 Conceptual reasoning

These questions are designed to stimulate your analytic and synthetic thinking<sup>3</sup>. In a Socratic discussion with your instructor, the goal is for these questions to prompt an extended dialogue where assumptions are revealed, conclusions are tested, and understanding is sharpened. Your instructor may also pose additional questions based on those assigned, in order to further probe and refine your conceptual understanding.

Questions that follow are presented to challenge and probe your understanding of various concepts presented in the tutorial. These questions are intended to serve as a guide for the Socratic dialogue between yourself and the instructor. Your instructor's task is to ensure you have a sound grasp of these concepts, and the questions contained in this document are merely a means to this end. Your instructor may, at his or her discretion, alter or substitute questions for the benefit of tailoring the discussion to each student's needs. The only absolute requirement is that each student is challenged and assessed at a level equal to or greater than that represented by the documented questions.

It is far more important that you convey your *reasoning* than it is to simply convey a correct answer. For this reason, you should refrain from researching other information sources to answer questions. What matters here is that *you* are doing the thinking. If the answer is incorrect, your instructor will work with you to correct it through proper reasoning. A correct answer without an adequate explanation of how you derived that answer is unacceptable, as it does not aid the learning or assessment process.

You will note a conspicuous lack of answers given for these conceptual questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your conceptual answers, where applicable, is to use circuit simulation software to explore the effects of changes made to circuits. For example, if one of these conceptual questions challenges you to predict the effects of altering some component parameter in a circuit, you may check the validity of your work by simulating that same parameter change within software and seeing if the results agree.

---

<sup>3</sup>*Analytical* thinking involves the “disassembly” of an idea into its constituent parts, analogous to dissection. *Synthetic* thinking involves the “assembly” of a new idea comprised of multiple concepts, analogous to construction. Both activities are high-level cognitive skills, extremely important for effective problem-solving, necessitating frequent challenge and regular practice to fully develop.



### 5.1.1 Reading outline and reflections

*“Reading maketh a full man; conference a ready man; and writing an exact man”* – Francis Bacon

Francis Bacon’s advice is a blueprint for effective education: reading provides the learner with knowledge, writing focuses the learner’s thoughts, and critical dialogue equips the learner to confidently communicate and apply their learning. Independent acquisition and application of knowledge is a powerful skill, well worth the effort to cultivate. To this end, students should read these educational resources closely, journal their own reflections on the reading, and discuss in detail their findings with classmates and instructor(s). You should be able to do all of the following after reading any instructional text:

☒ Briefly SUMMARIZE THE TEXT in the form of a journal entry documenting your learning as you progress through the course of study. Share this summary in dialogue with your classmates and instructor. Journaling is an excellent self-test of thorough reading because you cannot clearly express what you have not read or did not comprehend.

☒ Demonstrate ACTIVE READING STRATEGIES, including verbalizing your impressions as you read, simplifying long passages to convey the same ideas using fewer words, annotating text and illustrations with your own interpretations, working through mathematical examples shown in the text, cross-referencing passages with relevant illustrations and/or other passages, identifying problem-solving strategies applied by the author, etc. Technical reading is a special case of problem-solving, and so these strategies work precisely because they help solve any problem: paying attention to your own thoughts (metacognition), eliminating unnecessary complexities, identifying what makes sense, paying close attention to details, drawing connections between separated facts, and noting the successful strategies of others.

☒ Identify IMPORTANT THEMES, especially GENERAL LAWS and PRINCIPLES, expounded in the text and express them in the simplest of terms as though you were teaching an intelligent child. This emphasizes connections between related topics and develops your ability to communicate complex ideas to anyone.

☒ Form YOUR OWN QUESTIONS based on the reading, and then pose them to your instructor and classmates for their consideration. Anticipate both correct and incorrect answers, the incorrect answer(s) assuming one or more plausible misconceptions. This helps you view the subject from different perspectives to grasp it more fully.

☒ Devise EXPERIMENTS to test claims presented in the reading, or to disprove misconceptions. Predict possible outcomes of these experiments, and evaluate their meanings: what result(s) would confirm, and what would constitute disproof? Running mental simulations and evaluating results is essential to scientific and diagnostic reasoning.

☒ Specifically identify any points you found CONFUSING. The reason for doing this is to help diagnose misconceptions and overcome barriers to learning.

### 5.1.2 Foundational concepts

Correct analysis and diagnosis of electric circuits begins with a proper understanding of some basic concepts. The following is a list of some important concepts referenced in this module's full tutorial. Define each of them in your own words, and be prepared to illustrate each of these concepts with a description of a practical example and/or a live demonstration.

Energy

Conservation of Energy

Simplification as a problem-solving strategy

Thought experiments as a problem-solving strategy

Limiting cases as a problem-solving strategy

Annotating diagrams as a problem-solving strategy

Interpreting intermediate results as a problem-solving strategy

Graphing as a problem-solving strategy

Converting a qualitative problem into a quantitative problem

Converting a quantitative problem into a qualitative problem

Working “backwards” to validate calculated results

*Reductio ad absurdum*

Re-drawing schematics as a problem-solving strategy

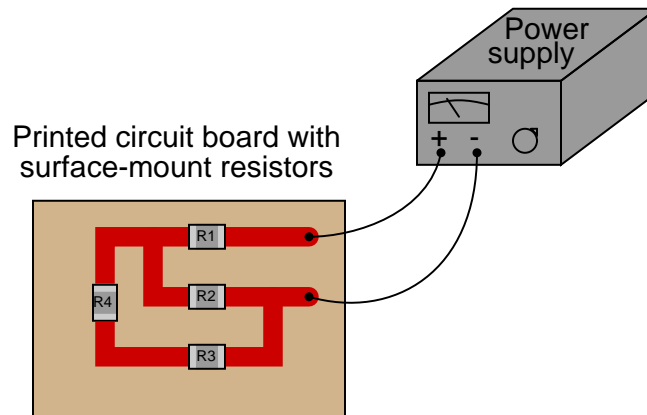
Cut-and-try problem-solving strategy

Algebraic substitution

???

### 5.1.3 Applying foundational concepts to a series-parallel circuit

Identify which foundational concept(s) apply to each of the declarations shown below regarding the following circuit, assuming a constant-voltage power supply. If a declaration is true, then identify it as such and note which concept supports that declaration; if a declaration is false, then identify it as such and note which concept is violated by that declaration:



- The current through resistor  $R_1$  will be the same as through resistor  $R_2$
- The current through resistor  $R_3$  will be the same as through resistor  $R_4$
- The voltage across resistor  $R_3$  will be the same as across resistor  $R_4$
- The voltage flowing through resistor  $R_1$  must be the same as through resistor  $R_4$
- If resistor  $R_1$  fails open, total current will decrease
- If resistor  $R_3$  fails open, total current will decrease
- The voltage dropped by resistor  $R_2$  will be equal to the sum of the voltages across resistors  $R_3$  and  $R_4$

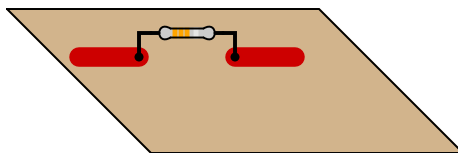
Here is a list of foundational concepts for your reference: **Conservation of Energy**, **Conservation of Electric Charge**, **behavior of sources vs. loads**, **Ohm's Law**, **Joule's Law**, **effects of open faults**, **effect of shorted faults**, **properties of series networks**, **properties of parallel networks**, **Kirchhoff's Voltage Law**, **Kirchhoff's Current Law**. More than one of these concepts may apply to a declaration, and some concepts may not apply to any listed declaration at all. Also, feel free to include foundational concepts not listed here.

#### Challenges

- Identify a component fault that would increase voltage across resistor  $R_2$ .

### 5.1.4 Parasitic inductance on a PCB

Components soldered into printed circuit boards often possess “stray” inductance, also known as *parasitic* inductance. Observe this resistor, soldered in to a circuit board:



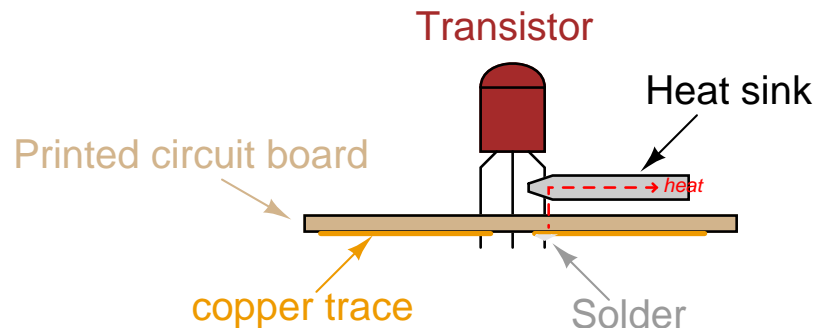
Where does the parasitic inductance come from? What is it about a resistor, mounted to a circuit board, that creates a (very) small amount of inductance? How is it possible to minimize this inductance, in case it is detrimental to the circuit’s operation?

#### Challenges

- Fast-switching digital logic circuits, even though operating on very low levels of current, nevertheless may experience extremely high *rates of change* due to the speed (frequency) of the signal oscillations. It is not uncommon in modern digital systems to witness rates as high as 500 A/ns (500 Amperes per nanosecond!). How much parasitic inductance would be necessary to drop 1 Volt at this high rate of change?

### 5.1.5 Heat sink for soldering

Semiconductor components such as diodes and transistors are easily damaged by the high temperatures of soldering, so care must be taken to protect these components during the soldering process. One way to do this is to use a *heat sink* to draw heat away from the component without cooling the connection point too much. Heat sinks made out of sheet metal may be temporarily clipped to the component leads, one at a time, to prevent the solder's high temperature from thermally conducting all the way to the component body:



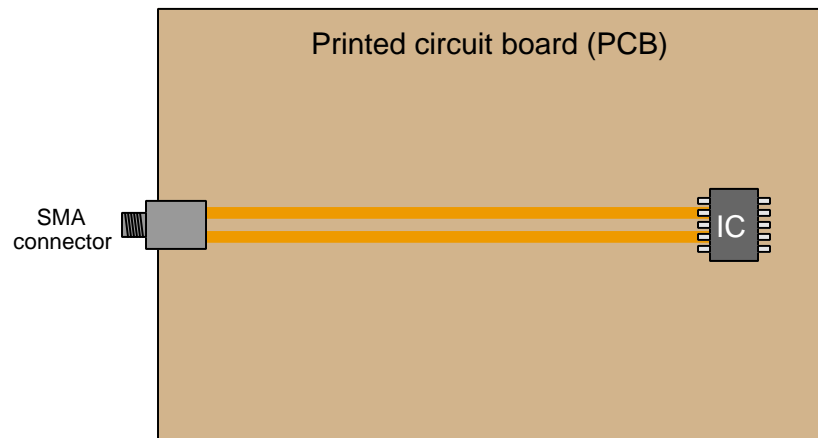
In the absence of a formal heat sink, can you think of any ways to fashion your own “impromptu” heat sinks out of commonly available tools and/or objects?

Challenges
------------

- Identify at least one passive (i.e. non-semiconductor) type of component that might also be heat-sensitive.

### 5.1.6 Impedance of lines on a printed circuit board

Copper traces on a printed circuit board (PCB) exhibit characteristic impedance just the same as conductors within a cable. Suppose it was determined that the traces connecting a SMA-style cable connector to an integrated circuit on a PCB had too much surge impedance<sup>4</sup>:



First, describe the problem(s) possibly created by improper impedance between the two traces on this board.

Next, explain what would need to be done to re-design the PCB to have less characteristic impedance between these two lines.

#### Challenges

- Does the material the PCB is made out of matter for  $Z_0$ ?

---

<sup>4</sup>This view of the PCB only shows these components and nothing else. A real PCB would likely be populated with a great many components!

## 5.2 Quantitative reasoning

These questions are designed to stimulate your computational thinking. In a Socratic discussion with your instructor, the goal is for these questions to reveal your mathematical approach(es) to problem-solving so that good technique and sound reasoning may be reinforced. Your instructor may also pose additional questions based on those assigned, in order to observe your problem-solving firsthand.

Mental arithmetic and estimations are strongly encouraged for all calculations, because without these abilities you will be unable to readily detect errors caused by calculator misuse (e.g. keystroke errors).

You will note a conspicuous lack of answers given for these quantitative questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. My advice is to use circuit simulation software such as SPICE to check the correctness of quantitative answers. Refer to those learning modules within this collection focusing on SPICE to see worked examples which you may use directly as practice problems for your own study, and/or as templates you may modify to run your own analyses and generate your own practice problems.

Completely worked example problems found in the Tutorial may also serve as “test cases<sup>5</sup>” for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely<sup>6</sup> on an answer key!

---

<sup>5</sup>In other words, set up the circuit simulation software to analyze the same circuit examples found in the Tutorial. If the simulated results match the answers shown in the Tutorial, it confirms the simulation has properly run. If the simulated results disagree with the Tutorial’s answers, something has been set up incorrectly in the simulation software. Using every Tutorial as practice in this way will quickly develop proficiency in the use of circuit simulation software.

<sup>6</sup>This approach is perfectly in keeping with the instructional philosophy of these learning modules: *teaching students to be self-sufficient thinkers*. Answer keys can be useful, but it is even more useful to your long-term success to have a set of tools on hand for checking your own work, because once you have left school and are on your own, there will no longer be “answer keys” available for the problems you will have to solve.



### 5.2.1 Miscellaneous physical constants

Note: constants shown in **bold** type are *exact*, not approximations. Values inside of parentheses show one standard deviation ( $\sigma$ ) of uncertainty in the final digits: for example, the magnetic permeability of free space value given as  $1.25663706212(19) \times 10^{-6}$  H/m represents a center value (i.e. the location parameter) of  $1.25663706212 \times 10^{-6}$  Henrys per meter with one standard deviation of uncertainty equal to  $0.0000000000019 \times 10^{-6}$  Henrys per meter.

Avogadro's number ( $N_A$ ) = **6.02214076**  $\times 10^{23}$  **per mole** (mol<sup>-1</sup>)

Boltzmann's constant ( $k$ ) = **1.380649**  $\times 10^{-23}$  **Joules per Kelvin** (J/K)

Electronic charge ( $e$ ) = **1.602176634**  $\times 10^{-19}$  **Coulomb** (C)

Faraday constant ( $F$ ) = **96,485.33212...**  $\times 10^4$  **Coulombs per mole** (C/mol)

Magnetic permeability of free space ( $\mu_0$ ) =  $1.25663706212(19) \times 10^{-6}$  Henrys per meter (H/m)

Electric permittivity of free space ( $\epsilon_0$ ) =  $8.8541878128(13) \times 10^{-12}$  Farads per meter (F/m)

Characteristic impedance of free space ( $Z_0$ ) =  $376.730313668(57)$  Ohms ( $\Omega$ )

Gravitational constant ( $G$ ) =  $6.67430(15) \times 10^{-11}$  cubic meters per kilogram-seconds squared (m<sup>3</sup>/kg-s<sup>2</sup>)

Molar gas constant ( $R$ ) = **8.314462618...** **Joules per mole-Kelvin** (J/mol-K) = 0.08205746(14) liters-atmospheres per mole-Kelvin

Planck constant ( $h$ ) = **6.62607015**  $\times 10^{-34}$  **joule-seconds** (J-s)

Stefan-Boltzmann constant ( $\sigma$ ) = **5.670374419...**  $\times 10^{-8}$  **Watts per square meter-Kelvin**<sup>4</sup> (W/m<sup>2</sup>·K<sup>4</sup>)

Speed of light in a vacuum ( $c$ ) = **299,792,458 meters per second** (m/s) = 186282.4 miles per second (mi/s)

Note: All constants taken from NIST data "Fundamental Physical Constants – Complete Listing", from <http://physics.nist.gov/constants>, National Institute of Standards and Technology (NIST), 2018 CODATA Adjustment.

### 5.2.2 Introduction to spreadsheets

A powerful computational tool you are encouraged to use in your work is a *spreadsheet*. Available on most personal computers (e.g. Microsoft Excel), *spreadsheet* software performs numerical calculations based on number values and formulae entered into cells of a grid. This grid is typically arranged as lettered columns and numbered rows, with each cell of the grid identified by its column/row coordinates (e.g. cell B3, cell A8). Each cell may contain a string of text, a number value, or a mathematical formula. The spreadsheet automatically updates the results of all mathematical formulae whenever the entered number values are changed. This means it is possible to set up a spreadsheet to perform a series of calculations on entered data, and those calculations will be re-done by the computer any time the data points are edited in any way.

For example, the following spreadsheet calculates average speed based on entered values of distance traveled and time elapsed:

	A	B	C	D
1	Distance traveled	46.9	Kilometers	
2	Time elapsed	1.18	Hours	
3	Average speed	= B1 / B2	km/h	
4				
5				

Text labels contained in cells A1 through A3 and cells C1 through C3 exist solely for readability and are not involved in any calculations. Cell B1 contains a sample distance value while cell B2 contains a sample time value. The formula for computing speed is contained in cell B3. Note how this formula begins with an “equals” symbol (=), references the values for distance and speed by lettered column and numbered row coordinates (B1 and B2), and uses a forward slash symbol for division (/). The coordinates B1 and B2 function as *variables*<sup>7</sup> would in an algebraic formula.

When this spreadsheet is executed, the numerical value 39.74576 will appear in cell B3 rather than the formula = B1 / B2, because 39.74576 is the computed speed value given 46.9 kilometers traveled over a period of 1.18 hours. If a different numerical value for distance is entered into cell B1 or a different value for time is entered into cell B2, cell B3’s value will automatically update. All you need to do is set up the given values and any formulae into the spreadsheet, and the computer will do all the calculations for you.

Cell B3 may be referenced by other formulae in the spreadsheet if desired, since it is a variable just like the given values contained in B1 and B2. This means it is possible to set up an entire chain of calculations, one dependent on the result of another, in order to arrive at a final value. The arrangement of the given data and formulae need not follow any pattern on the grid, which means you may place them anywhere.

---

<sup>7</sup>Spreadsheets may also provide means to attach text labels to cells for use as variable names (Microsoft Excel simply calls these labels “names”), but for simple spreadsheets such as those shown here it’s usually easier just to use the standard coordinate naming for each cell.

Common<sup>8</sup> arithmetic operations available for your use in a spreadsheet include the following:

- Addition (+)
- Subtraction (-)
- Multiplication (\*)
- Division (/)
- Powers (^)
- Square roots (sqrt())
- Logarithms (ln() , log10())

Parentheses may be used to ensure<sup>9</sup> proper order of operations within a complex formula. Consider this example of a spreadsheet implementing the *quadratic formula*, used to solve for roots of a polynomial expression in the form of  $ax^2 + bx + c$ :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

	A	B
1	x_1	= (-B4 + sqrt((B4^2) - (4*B3*B5))) / (2*B3)
2	x_2	= (-B4 - sqrt((B4^2) - (4*B3*B5))) / (2*B3)
3	a =	9
4	b =	5
5	c =	-2

This example is configured to compute roots<sup>10</sup> of the polynomial  $9x^2 + 5x - 2$  because the values of 9, 5, and  $-2$  have been inserted into cells B3, B4, and B5, respectively. Once this spreadsheet has been built, though, it may be used to calculate the roots of *any* second-degree polynomial expression simply by entering the new  $a$ ,  $b$ , and  $c$  coefficients into cells B3 through B5. The numerical values appearing in cells B1 and B2 will be automatically updated by the computer immediately following any changes made to the coefficients.

<sup>8</sup>Modern spreadsheet software offers a bewildering array of mathematical functions you may use in your computations. I recommend you consult the documentation for your particular spreadsheet for information on operations other than those listed here.

<sup>9</sup>Spreadsheet programs, like text-based programming languages, are designed to follow standard order of operations by default. However, my personal preference is to use parentheses even where strictly unnecessary just to make it clear to any other person viewing the formula what the intended order of operations is.

<sup>10</sup>Reviewing some algebra here, a *root* is a value for  $x$  that yields an overall value of zero for the polynomial. For this polynomial ( $9x^2 + 5x - 2$ ) the two roots happen to be  $x = 0.269381$  and  $x = -0.82494$ , with these values displayed in cells B1 and B2, respectively upon execution of the spreadsheet.

Alternatively, one could break up the long quadratic formula into smaller pieces like this:

$$y = \sqrt{b^2 - 4ac} \quad z = 2a$$

$$x = \frac{-b \pm y}{z}$$

	A	B	C
<b>1</b>	x_1	= ( -B4 + C1 ) / C2	= sqrt ( (B4^2) - (4*B3*B5) )
<b>2</b>	x_2	= ( -B4 - C1 ) / C2	= 2*B3
<b>3</b>	a =	9	
<b>4</b>	b =	5	
<b>5</b>	c =	-2	

Note how the square-root term ( $y$ ) is calculated in cell C1, and the denominator term ( $z$ ) in cell C2. This makes the two final formulae (in cells B1 and B2) simpler to interpret. The positioning of all these cells on the grid is completely arbitrary<sup>11</sup> – all that matters is that they properly reference each other in the formulae.

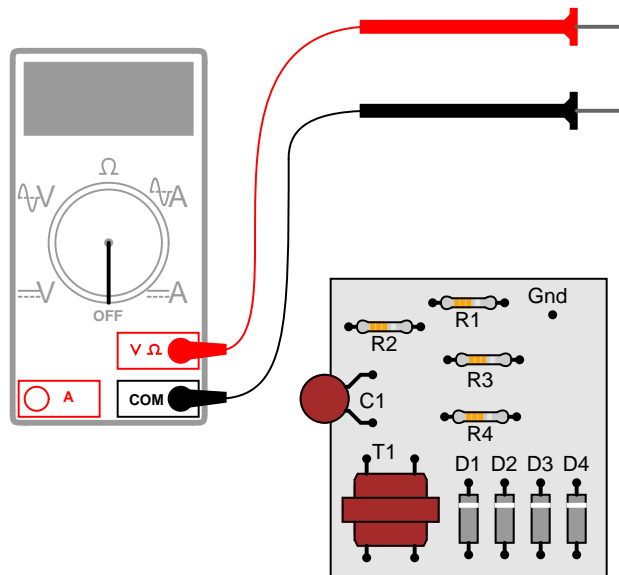
Spreadsheets are particularly useful for situations where the same set of calculations representing a circuit or other system must be repeated for different initial conditions. The power of a spreadsheet is that it automates what would otherwise be a tedious set of calculations. One specific application of this is to simulate the effects of various components within a circuit failing with abnormal values (e.g. a shorted resistor simulated by making its value nearly zero; an open resistor simulated by making its value extremely large). Another application is analyzing the behavior of a circuit design given new components that are out of specification, and/or aging components experiencing drift over time.

---

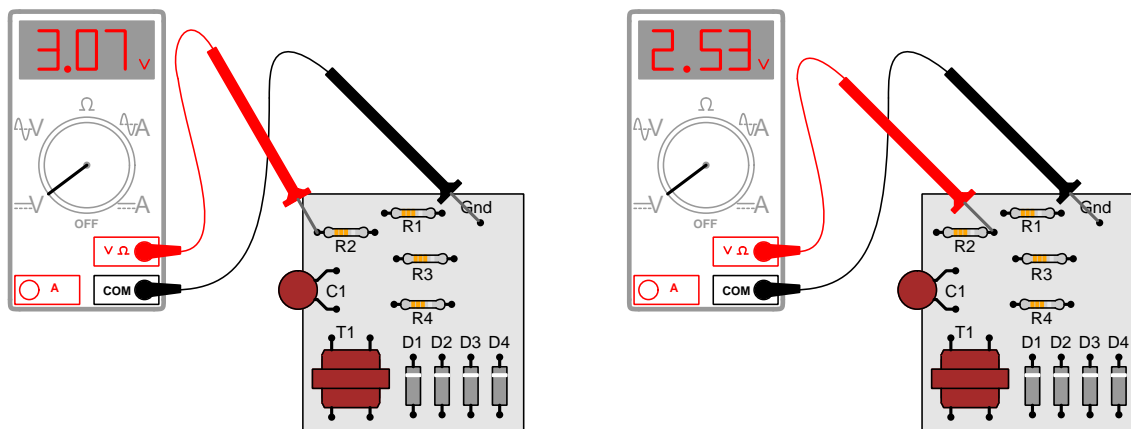
<sup>11</sup>My personal preference is to locate all the “given” data in the upper-left cells of the spreadsheet grid (each data point flanked by a sensible name in the cell to the left and units of measurement in the cell to the right as illustrated in the first distance/time spreadsheet example), sometimes coloring them in order to clearly distinguish which cells contain entered data versus which cells contain computed results from formulae. I like to place all formulae in cells below the given data, and try to arrange them in logical order so that anyone examining my spreadsheet will be able to figure out *how* I constructed a solution. This is a general principle I believe all computer programmers should follow: *document and arrange your code to make it easy for other people to learn from it.*

### 5.2.3 Current measurement on a PCB

Suppose you wanted to measure the amount of current going through resistor  $R_2$  on this printed circuit board (PCB), but did not have the luxury of breaking the circuit to do so (unsoldering one end of the resistor, detaching it from the PCB, and connecting an ammeter in series). All you can do while the circuit is powered is measure voltage with a voltmeter:



So, you decide to touch the black probe of the voltmeter to the circuit's "Gnd" (ground) test point, and measure the voltage with reference to ground on both sides of  $R_2$ . The results are shown here:



$R_2$ 's color code is Orange, Orange, Red, Gold. Based on this information, determine both the

direction and the magnitude of DC current through this resistor, and explain how you did so.

Challenges
------------

- Devise a simpler way to determine  $R_2$ 's current, using the same test equipment.
- Identify possible sources of error in this procedure, where we infer current without taking a direct current measurement.
- Suppose the meter's red test lead were the one connected to ground. What effect would this have on the results, and would it impact the determination of  $R_2$ 's current?

### 5.3 Diagnostic reasoning

These questions are designed to stimulate your deductive and inductive thinking, where you must apply general principles to specific scenarios (deductive) and also derive conclusions about the failed circuit from specific details (inductive). In a Socratic discussion with your instructor, the goal is for these questions to reinforce your recall and use of general circuit principles and also challenge your ability to integrate multiple symptoms into a sensible explanation of what's wrong in a circuit. Your instructor may also pose additional questions based on those assigned, in order to further challenge and sharpen your diagnostic abilities.

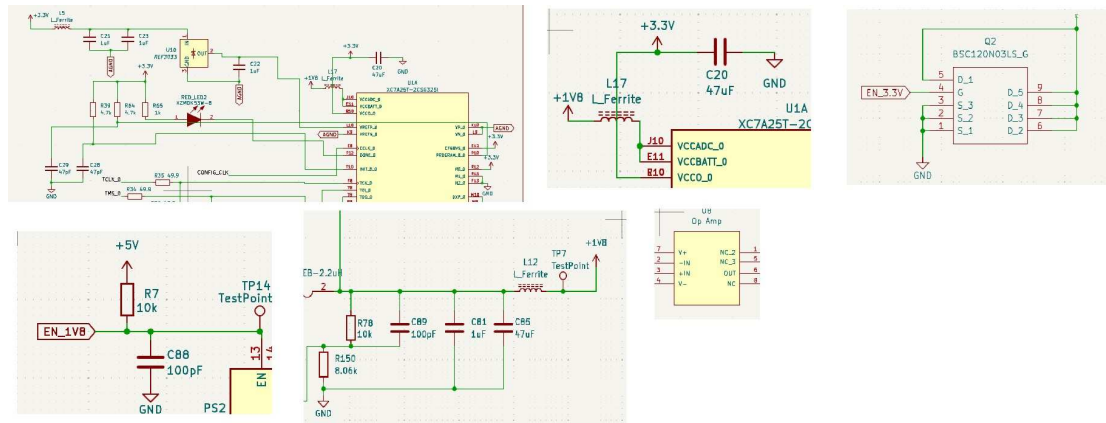
As always, your goal is to fully *explain* your analysis of each problem. Simply obtaining a correct answer is not good enough – you must also demonstrate sound reasoning in order to successfully complete the assignment. Your instructor's responsibility is to probe and challenge your understanding of the relevant principles and analytical processes in order to ensure you have a strong foundation upon which to build further understanding.

You will note a conspicuous lack of answers given for these diagnostic questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your diagnostic answers, where applicable, is to use circuit simulation software to explore the effects of faults placed in circuits. For example, if one of these diagnostic questions requires that you predict the effect of an open or a short in a circuit, you may check the validity of your work by simulating that same fault (substituting a very high resistance in place of that component for an open, and substituting a very low resistance for a short) within software and seeing if the results agree.

### 5.3.1 Sloppy schematic capture

Good printed circuit board design always begins with a well-drawn schematic diagram. The following collage is a set of screenshot snippers from a real student-drawn schematic diagram (in KiCad) with multiple examples of poor layout. The circuit itself is electrically correct, but the layout is poorly done. See how many problems you can identify!



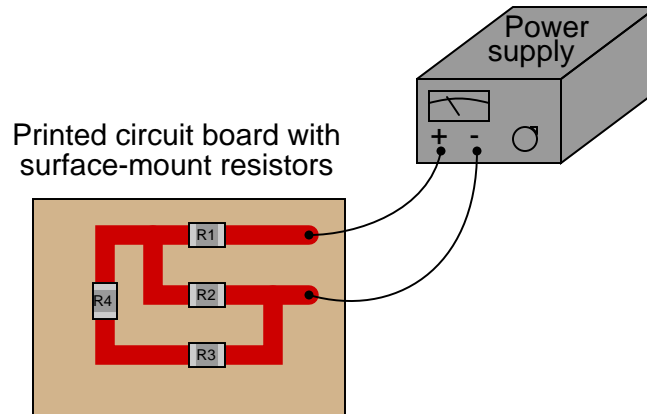
#### Challenges

- So long as the PCB layout software is able to accurately “capture” the schematic diagram’s logical connections to create a ratsnets, what does it actually matter if the schematic isn’t neatly laid out?



### 5.3.2 Faults in a PCB circuit

Predict the effects of the following faults in this circuit, one fault at a time, assuming the power supply is a fixed-voltage unit:



- If  $R_1$  fails open . . .
- If  $R_2$  fails open . . .
- If  $R_3$  fails open . . .
- If  $R_4$  fails open . . .
- If  $R_1$  fails shorted . . .
- If  $R_2$  fails shorted . . .
- If  $R_3$  fails shorted . . .
- If  $R_4$  fails shorted . . .

#### Challenges

- Sometimes qualitative problems become easier if we add some numbers to them. Re-answer these questions given a voltage source value chosen by you, and resistors that are all equal-valued (before failing).
- Components soldered to circuit boards cannot be easily removed, as de-soldering is required. This makes it difficult to test for current through any component, as there is no convenient way to break the circuit so as to insert an ammeter. Describe how it may be possible to infer current through certain circuit board components using only a voltmeter.

## Appendix A

# Problem-Solving Strategies

The ability to solve complex problems is arguably one of the most valuable skills one can possess, and this skill is particularly important in any science-based discipline.

- Study principles, not procedures. Don't be satisfied with merely knowing how to compute solutions – learn *why* those solutions work.
- Identify what it is you need to solve, identify all relevant data, identify all units of measurement, identify any general principles or formulae linking the given information to the solution, and then identify any “missing pieces” to a solution. Annotate all diagrams with this data.
- Sketch a diagram to help visualize the problem. When building a real system, always devise a plan for that system and analyze its function *before* constructing it.
- Follow the units of measurement and meaning of every calculation. If you are ever performing mathematical calculations as part of a problem-solving procedure, and you find yourself unable to apply each and every intermediate result to some aspect of the problem, it means you don't understand what you are doing. Properly done, every mathematical result should have practical meaning for the problem, and not just be an abstract number. You should be able to identify the proper units of measurement for each and every calculated result, and show where that result fits into the problem.
- Perform “thought experiments” to explore the effects of different conditions for theoretical problems. When troubleshooting real systems, perform *diagnostic tests* rather than visually inspecting for faults, the best diagnostic test being the one giving you the most information about the nature and/or location of the fault with the fewest steps.
- Simplify the problem until the solution becomes obvious, and then use that obvious case as a model to follow in solving the more complex version of the problem.
- Check for exceptions to see if your solution is incorrect or incomplete. A good solution will work for *all* known conditions and criteria. A good example of this is the process of testing scientific hypotheses: the task of a scientist is not to find support for a new idea, but rather to *challenge* that new idea to see if it holds up under a battery of tests. The philosophical

principle of *reductio ad absurdum* (i.e. disproving a general idea by finding a specific case where it fails) is useful here.

- Work “backward” from a hypothetical solution to a new set of given conditions.
- Add quantities to problems that are qualitative in nature, because sometimes a little math helps illuminate the scenario.
- Sketch graphs illustrating how variables relate to each other. These may be quantitative (i.e. with realistic number values) or qualitative (i.e. simply showing increases and decreases).
- Treat quantitative problems as qualitative in order to discern the relative magnitudes and/or directions of change of the relevant variables. For example, try determining what happens if a certain variable were to increase or decrease before attempting to precisely calculate quantities: how will each of the dependent variables respond, by increasing, decreasing, or remaining the same as before?
- Consider limiting cases. This works especially well for qualitative problems where you need to determine which direction a variable will change. Take the given condition and magnify that condition to an extreme degree as a way of simplifying the direction of the system’s response.
- Check your work. This means regularly testing your conclusions to see if they make sense. This does *not* mean repeating the same steps originally used to obtain the conclusion(s), but rather to use some other means to check validity. Simply repeating procedures often leads to *repeating the same errors* if any were made, which is why alternative paths are better.

## Appendix B

# Instructional philosophy

*“The unexamined circuit is not worth energizing”* – Socrates (if he had taught electricity)

These learning modules, although useful for self-study, were designed to be used in a formal learning environment where a subject-matter expert challenges students to digest the content and exercise their critical thinking abilities in the answering of questions and in the construction and testing of working circuits.

The following principles inform the instructional and assessment philosophies embodied in these learning modules:

- The first goal of education is to enhance clear and independent thought, in order that every student reach their fullest potential in a highly complex and inter-dependent world. Robust reasoning is *always* more important than particulars of any subject matter, because its application is universal.
- Literacy is fundamental to independent learning and thought because text continues to be the most efficient way to communicate complex ideas over space and time. Those who cannot read with ease are limited in their ability to acquire knowledge and perspective.
- Articulate communication is fundamental to work that is complex and interdisciplinary.
- Faulty assumptions and poor reasoning are best corrected through challenge, not presentation. The rhetorical technique of *reductio ad absurdum* (disproving an assertion by exposing an absurdity) works well to discipline student’s minds, not only to correct the problem at hand but also to learn how to detect and correct future errors.
- Important principles should be repeatedly explored and widely applied throughout a course of study, not only to reinforce their importance and help ensure their mastery, but also to showcase the interconnectedness and utility of knowledge.

These learning modules were expressly designed to be used in an “inverted” teaching environment<sup>1</sup> where students first read the introductory and tutorial chapters on their own, then individually attempt to answer the questions and construct working circuits according to the experiment and project guidelines. The instructor never lectures, but instead meets regularly with each individual student to review their progress, answer questions, identify misconceptions, and challenge the student to new depths of understanding through further questioning. Regular meetings between instructor and student should resemble a Socratic<sup>2</sup> dialogue, where questions serve as scalpels to dissect topics and expose assumptions. The student passes each module only after consistently demonstrating their ability to logically analyze and correctly apply all major concepts in each question or project/experiment. The instructor must be vigilant in probing each student’s understanding to ensure they are truly *reasoning* and not just *memorizing*. This is why “Challenge” points appear throughout, as prompts for students to think deeper about topics and as starting points for instructor queries. Sometimes these challenge points require additional knowledge that hasn’t been covered in the series to answer in full. This is okay, as the major purpose of the Challenges is to stimulate analysis and synthesis on the part of each student.

The instructor must possess enough mastery of the subject matter and awareness of students’ reasoning to generate their own follow-up questions to practically any student response. Even completely correct answers given by the student should be challenged by the instructor for the purpose of having students practice articulating their thoughts and defending their reasoning. Conceptual errors committed by the student should be exposed and corrected not by direct instruction, but rather by reducing the errors to an absurdity<sup>3</sup> through well-chosen questions and thought experiments posed by the instructor. Becoming proficient at this style of instruction requires time and dedication, but the positive effects on critical thinking for both student and instructor are spectacular.

An inspection of these learning modules reveals certain unique characteristics. One of these is a bias toward thorough explanations in the tutorial chapters. Without a live instructor to explain concepts and applications to students, the text itself must fulfill this role. This philosophy results in lengthier explanations than what you might typically find in a textbook, each step of the reasoning process fully explained, including footnotes addressing common questions and concerns students raise while learning these concepts. Each tutorial seeks to not only explain each major concept in sufficient detail, but also to explain the logic of each concept and how each may be developed

---

<sup>1</sup>In a traditional teaching environment, students first encounter new information via *lecture* from an expert, and then independently apply that information via *homework*. In an “inverted” course of study, students first encounter new information via *homework*, and then independently apply that information under the scrutiny of an expert. The expert’s role in lecture is to simply *explain*, but the expert’s role in an inverted session is to *challenge*, *critique*, and if necessary *explain* where gaps in understanding still exist.

<sup>2</sup>Socrates is a figure in ancient Greek philosophy famous for his unflinching style of questioning. Although he authored no texts, he appears as a character in Plato’s many writings. The essence of Socratic philosophy is to leave no question unexamined and no point of view unchallenged. While purists may argue a topic such as electric circuits is too narrow for a true Socratic-style dialogue, I would argue that the essential thought processes involved with scientific reasoning on *any* topic are not far removed from the Socratic ideal, and that students of electricity and electronics would do very well to challenge assumptions, pose thought experiments, identify fallacies, and otherwise employ the arsenal of critical thinking skills modeled by Socrates.

<sup>3</sup>This rhetorical technique is known by the Latin phrase *reductio ad absurdum*. The concept is to expose errors by counter-example, since only one solid counter-example is necessary to disprove a universal claim. As an example of this, consider the common misconception among beginning students of electricity that voltage cannot exist without current. One way to apply *reductio ad absurdum* to this statement is to ask how much current passes through a fully-charged battery connected to nothing (i.e. a clear example of voltage existing without current).

from “first principles”. Again, this reflects the goal of developing clear and independent thought in students’ minds, by showing how clear and logical thought was used to forge each concept. Students benefit from witnessing a model of clear thinking in action, and these tutorials strive to be just that.

Another characteristic of these learning modules is a lack of step-by-step instructions in the Project and Experiment chapters. Unlike many modern workbooks and laboratory guides where step-by-step instructions are prescribed for each experiment, these modules take the approach that students must learn to closely read the tutorials and apply their own reasoning to identify the appropriate experimental steps. Sometimes these steps are plainly declared in the text, just not as a set of enumerated points. At other times certain steps are implied, an example being assumed competence in test equipment use where the student should not need to be told *again* how to use their multimeter because that was thoroughly explained in previous lessons. In some circumstances no steps are given at all, leaving the entire procedure up to the student.

This lack of prescription is not a flaw, but rather a feature. Close reading and clear thinking are foundational principles of this learning series, and in keeping with this philosophy all activities are designed to *require* those behaviors. Some students may find the lack of prescription frustrating, because it demands more from them than what their previous educational experiences required. This frustration should be interpreted as an unfamiliarity with autonomous thinking, a problem which must be corrected if the student is ever to become a self-directed learner and effective problem-solver. Ultimately, the need for students to read closely and think clearly is more important both in the near-term and far-term than any specific facet of the subject matter at hand. If a student takes longer than expected to complete a module because they are forced to outline, digest, and reason on their own, so be it. The future gains enjoyed by developing this mental discipline will be well worth the additional effort and delay.

Another feature of these learning modules is that they do not treat topics in isolation. Rather, important concepts are introduced early in the series, and appear repeatedly as stepping-stones toward other concepts in subsequent modules. This helps to avoid the “compartmentalization” of knowledge, demonstrating the inter-connectedness of concepts and simultaneously reinforcing them. Each module is fairly complete in itself, reserving the beginning of its tutorial to a review of foundational concepts.

This methodology of assigning text-based modules to students for digestion and then using Socratic dialogue to assess progress and hone students’ thinking was developed over a period of several years by the author with his Electronics and Instrumentation students at the two-year college level. While decidedly unconventional and sometimes even unsettling for students accustomed to a more passive lecture environment, this instructional philosophy has proven its ability to convey conceptual mastery, foster careful analysis, and enhance employability so much better than lecture that the author refuses to ever teach by lecture again.

Problems which often go undiagnosed in a lecture environment are laid bare in this “inverted” format where students must articulate and logically defend their reasoning. This, too, may be unsettling for students accustomed to lecture sessions where the instructor cannot tell for sure who comprehends and who does not, and this vulnerability necessitates sensitivity on the part of the “inverted” session instructor in order that students never feel discouraged by having their errors exposed. *Everyone* makes mistakes from time to time, and learning is a lifelong process! Part of the instructor’s job is to build a culture of learning among the students where errors are not seen as shameful, but rather as opportunities for progress.

To this end, instructors managing courses based on these modules should adhere to the following principles:

- Student questions are always welcome and demand thorough, honest answers. The only type of question an instructor should refuse to answer is one the student should be able to easily answer on their own. Remember, *the fundamental goal of education is for each student to learn to think clearly and independently*. This requires hard work on the part of the student, which no instructor should ever circumvent. Anything done to bypass the student's responsibility to do that hard work ultimately limits that student's potential and thereby does real harm.
- It is not only permissible, but encouraged, to answer a student's question by asking questions in return, these follow-up questions designed to guide the student to reach a correct answer through their own reasoning.
- All student answers demand to be challenged by the instructor and/or by other students. This includes both correct and incorrect answers – the goal is to practice the articulation and defense of one's own reasoning.
- No reading assignment is deemed complete unless and until the student demonstrates their ability to accurately summarize the major points in their own terms. Recitation of the original text is unacceptable. This is why every module contains an "Outline and reflections" question as well as a "Foundational concepts" question in the Conceptual reasoning section, to prompt reflective reading.
- No assigned question is deemed answered unless and until the student demonstrates their ability to consistently and correctly apply the concepts to *variations* of that question. This is why module questions typically contain multiple "Challenges" suggesting different applications of the concept(s) as well as variations on the same theme(s). Instructors are encouraged to devise as many of their own "Challenges" as they are able, in order to have a multitude of ways ready to probe students' understanding.
- No assigned experiment or project is deemed complete unless and until the student demonstrates the task in action. If this cannot be done "live" before the instructor, video-recordings showing the demonstration are acceptable. All relevant safety precautions must be followed, all test equipment must be used correctly, and the student must be able to properly explain all results. The student must also successfully answer all Challenges presented by the instructor for that experiment or project.

Students learning from these modules would do well to abide by the following principles:

- No text should be considered fully and adequately read unless and until you can express every idea *in your own words, using your own examples*.
- You should always articulate your thoughts as you read the text, noting points of agreement, confusion, and epiphanies. Feel free to print the text on paper and then write your notes in the margins. Alternatively, keep a journal for your own reflections as you read. This is truly a helpful tool when digesting complicated concepts.
- Never take the easy path of highlighting or underlining important text. Instead, *summarize* and/or *comment* on the text using your own words. This actively engages your mind, allowing you to more clearly perceive points of confusion or misunderstanding on your own.
- A very helpful strategy when learning new concepts is to place yourself in the role of a teacher, if only as a mental exercise. Either explain what you have recently learned to someone else, or at least *imagine* yourself explaining what you have learned to someone else. The simple act of having to articulate new knowledge and skill forces you to take on a different perspective, and will help reveal weaknesses in your understanding.
- Perform each and every mathematical calculation and thought experiment shown in the text on your own, referring back to the text to see that your results agree. This may seem trivial and unnecessary, but it is critically important to ensuring you actually understand what is presented, especially when the concepts at hand are complicated and easy to misunderstand. Apply this same strategy to become proficient in the use of *circuit simulation software*, checking to see if your simulated results agree with the results shown in the text.
- Above all, recognize that learning is hard work, and that a certain level of frustration is unavoidable. There are times when you will struggle to grasp some of these concepts, and that struggle is a natural thing. Take heart that it will yield with persistent and varied<sup>4</sup> effort, and never give up!

Students interested in using these modules for self-study will also find them beneficial, although the onus of responsibility for thoroughly reading and answering questions will of course lie with that individual alone. If a qualified instructor is not available to challenge students, a workable alternative is for students to form study groups where they challenge<sup>5</sup> one another.

To high standards of education,

Tony R. Kuphaldt

---

<sup>4</sup>As the old saying goes, “Insanity is trying the same thing over and over again, expecting different results.” If you find yourself stumped by something in the text, you should attempt a different approach. Alter the thought experiment, change the mathematical parameters, do whatever you can to see the problem in a slightly different light, and then the solution will often present itself more readily.

<sup>5</sup>Avoid the temptation to simply share answers with study partners, as this is really counter-productive to learning. Always bear in mind that the answer to any question is far less important in the long run than the method(s) used to obtain that answer. The goal of education is to empower one’s life through the improvement of clear and independent thought, literacy, expression, and various practical skills.





## Appendix C

# Tools used

I am indebted to the developers of many open-source software applications in the creation of these learning modules. The following is a list of these applications with some commentary on each.

You will notice a theme common to many of these applications: a bias toward *code*. Although I am by no means an expert programmer in any computer language, I understand and appreciate the flexibility offered by code-based applications where the user (you) enters commands into a plain ASCII text file, which the software then reads and processes to create the final output. Code-based computer applications are by their very nature *extensible*, while WYSIWYG (What You See Is What You Get) applications are generally limited to whatever user interface the developer makes for you.

### The GNU/Linux computer operating system

There is so much to be said about Linus Torvalds' **Linux** and Richard Stallman's **GNU** project. First, to credit just these two individuals is to fail to do justice to the *mob* of passionate volunteers who contributed to make this amazing software a reality. I first learned of **Linux** back in 1996, and have been using this operating system on my personal computers almost exclusively since then. It is *free*, it is completely *configurable*, and it permits the continued use of highly efficient **Unix** applications and scripting languages (e.g. shell scripts, Makefiles, **sed**, **awk**) developed over many decades. **Linux** not only provided me with a powerful computing platform, but its open design served to inspire my life's work of creating open-source educational resources.

### Bram Moolenaar's **Vim** text editor

Writing code for any code-based computer application requires a *text editor*, which may be thought of as a word processor strictly limited to outputting plain-ASCII text files. Many good text editors exist, and one's choice of text editor seems to be a deeply personal matter within the programming world. I prefer **Vim** because it operates very similarly to **vi** which is ubiquitous on **Unix/Linux** operating systems, and because it may be entirely operated via keyboard (i.e. no mouse required) which makes it fast to use.

### Donald Knuth's $\text{\TeX}$ typesetting system

Developed in the late 1970's and early 1980's by computer scientist extraordinaire Donald Knuth to typeset his multi-volume magnum opus *The Art of Computer Programming*, this software allows the production of formatted text for screen-viewing or paper printing, all by writing plain-text code to describe how the formatted text is supposed to appear.  $\text{\TeX}$  is not just a markup language for documents, but it is also a Turing-complete programming language in and of itself, allowing useful algorithms to be created to control the production of documents. Simply put,  *$\text{\TeX}$  is a programmer's approach to word processing*. Since  $\text{\TeX}$  is controlled by code written in a plain-text file, this means anyone may read that plain-text file to see exactly how the document was created. This openness afforded by the code-based nature of  $\text{\TeX}$  makes it relatively easy to learn how other people have created their own  $\text{\TeX}$  documents. By contrast, examining a beautiful document created in a conventional WYSIWYG word processor such as Microsoft **Word** suggests nothing to the reader about *how* that document was created, or what the user might do to create something similar. As Mr. Knuth himself once quipped, conventional word processing applications should be called WYSIAYG (What You See Is *All* You Get).

### Leslie Lamport's $\text{\LaTeX}$ extensions to $\text{\TeX}$

Like all true programming languages,  $\text{\TeX}$  is inherently extensible. So, years after the release of  $\text{\TeX}$  to the public, Leslie Lamport decided to create a massive extension allowing easier compilation of book-length documents. The result was  $\text{\LaTeX}$ , which is the markup language used to create all ModEL module documents. You could say that  $\text{\TeX}$  is to  $\text{\LaTeX}$  as **C** is to **C++**. This means it is permissible to use any and all  $\text{\TeX}$  commands within  $\text{\LaTeX}$  source code, and it all still works. Some of the features offered by  $\text{\LaTeX}$  that would be challenging to implement in  $\text{\TeX}$  include automatic index and table-of-content creation.

### Tim Edwards' **Xcircuit** drafting program

This wonderful program is what I use to create all the schematic diagrams and illustrations (but not photographic images or mathematical plots) throughout the ModEL project. It natively outputs PostScript format which is a true vector graphic format (this is why the images do not pixellate when you zoom in for a closer view), and it is so simple to use that I have never had to read the manual! Object libraries are easy to create for **Xcircuit**, being plain-text files using PostScript programming conventions. Over the years I have collected a large set of object libraries useful for drawing electrical and electronic schematics, pictorial diagrams, and other technical illustrations.

### Gimp graphic image manipulation program

Essentially an open-source clone of Adobe's **PhotoShop**, I use **Gimp** to resize, crop, and convert file formats for all of the photographic images appearing in the **ModEL** modules. Although **Gimp** does offer its own scripting language (called **Script-Fu**), I have never had occasion to use it. Thus, my utilization of **Gimp** to merely crop, resize, and convert graphic images is akin to using a sword to slice bread.

### SPICE circuit simulation program

**SPICE** is to circuit analysis as **T<sub>E</sub>X** is to document creation: it is a form of markup language designed to describe a certain object to be processed in plain-ASCII text. When the plain-text "source file" is compiled by the software, it outputs the final result. More modern circuit analysis tools certainly exist, but I prefer **SPICE** for the following reasons: it is *free*, it is *fast*, it is *reliable*, and it is a fantastic tool for *teaching* students of electricity and electronics how to write simple code. I happen to use rather old versions of **SPICE**, version 2g6 being my "go to" application when I only require text-based output. **NGSPICE** (version 26), which is based on Berkeley **SPICE** version 3f5, is used when I require graphical output for such things as time-domain waveforms and Bode plots. In all **SPICE** example netlists I strive to use coding conventions compatible with all **SPICE** versions.

### Andrew D. Hwang's ePiX mathematical visualization programming library

This amazing project is a **C++** library you may link to any **C/C++** code for the purpose of generating PostScript graphic images of mathematical functions. As a completely free and open-source project, it does all the plotting I would otherwise use a Computer Algebra System (CAS) such as **Mathematica** or **Maple** to do. It should be said that **ePiX** is *not* a Computer Algebra System like **Mathematica** or **Maple**, but merely a mathematical *visualization* tool. In other words, it won't determine integrals for you (you'll have to implement that in your own **C/C++** code!), but it can graph the results, and it does so beautifully. What I really admire about **ePiX** is that it is a **C++** programming library, which means it builds on the existing power and toolset available with that programming language. Mr. Hwang could have probably developed his own stand-alone application for mathematical plotting, but by creating a **C++** library to do the same thing he accomplished something much greater.

### `gnuplot` mathematical visualization software

Another open-source tool for mathematical visualization is `gnuplot`. Interestingly, this tool is *not* part of Richard Stallman’s GNU project, its name being a coincidence. For this reason the authors prefer “gnu” *not* be capitalized at all to avoid confusion. This is a much “lighter-weight” alternative to a spreadsheet for plotting tabular data, and the fact that it easily outputs directly to an X11 console or a file in a number of different graphical formats (including PostScript) is very helpful. I typically set my `gnuplot` output format to default (X11 on my Linux PC) for quick viewing while I’m developing a visualization, then switch to PostScript file export once the visual is ready to include in the document(s) I’m writing. As with my use of `Gimp` to do rudimentary image editing, my use of `gnuplot` only scratches the surface of its capabilities, but the important points are that it’s *free* and that it *works well*.

### Python programming language

Both Python and C++ find extensive use in these modules as instructional aids and exercises, but I’m listing Python here as a *tool* for myself because I use it almost daily as a *calculator*. If you open a Python interpreter console and type `from math import *` you can type mathematical expressions and have it return results just as you would on a hand calculator. Complex-number (i.e. *phasor*) arithmetic is similarly supported if you include the complex-math library (`from cmath import *`). Examples of this are shown in the Programming References chapter (if included) in each module. Of course, being a fully-featured programming language, Python also supports conditionals, loops, and other structures useful for calculation of quantities. Also, running in a console environment where all entries and returned values show as text in a chronologically-ordered list makes it easy to copy-and-paste those calculations to document exactly how they were performed.

## Appendix D

# Creative Commons License

### Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License (“Public License”). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

#### Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter’s License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or

limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

## Section 2 – Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures.

For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

### **Section 3 – License Conditions.**

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;



- iii. a notice that refers to this Public License;
- iv. a notice that refers to the disclaimer of warranties;
- v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

#### **Section 4 – Sui Generis Database Rights.**

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

#### **Section 5 – Disclaimer of Warranties and Limitation of Liability.**

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors,

whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

## **Section 6 – Term and Termination.**

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

## **Section 7 – Other Terms and Conditions.**

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

## **Section 8 – Interpretation.**

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully

be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at [creativecommons.org/policies](https://creativecommons.org/policies), Creative Commons does not authorize the use of the trademark “Creative Commons” or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at [creativecommons.org](https://creativecommons.org).



## Appendix E

# References

Aravamudhan, Srinivasa; Combs, Christopher; Prasad, Abhishek; Abraham, Amit; “Multi-Faceted Approach to Minimize Printed Circuit Board Warpage in Board Assembly Process”, Proceedings of SMTA International, 25-29 September 2016.

“The Designer’s Handbook for DFM”, Sierra Circuits Incorporated, Sunnyvale, CA, 2022.

“The Gerber Layer Format Specification”, revision 2022.02, Ucamco NV, February 2022.

“High-Speed Layout Guidelines”, document SCAA082A, Texas Instruments Incorporated, August 2017.

“IPC-2152 Standard for Determining Current-Carrying Capacity in Printed Board Design”, IPC, Northbrook, IL, May 2003.

“IPC-2221 Generic Standard on Printed Board Design”, IPC, Northbrook, IL, February 1998.

Jones, David L., *PCB Design Tutorial*, revision A, 29 June 2004.

Tavernier, Karel, “PCB Fabrication Data – Design to Fabrication Data Transfer”, version 6, September 2015.

“ThermaWick Thermal Jumper Surface Mount Chip”, document number 60157, Vishay Intertechnology Incorporated, 1 January 2022.



# Appendix F

## Version history

This is a list showing all significant additions, corrections, and other edits made to this learning module. Each entry is referenced by calendar date in reverse chronological order (newest version first), which appears on the front cover of every learning module for easy reference. Any contributors to this open-source document are listed here as well.

**7-13 April 2025** – added a new Technical Reference section, this one on KiCad recommendations. Also renamed “Tony’s layout recommendations” to “Tony’s PCB layout recommendations” to make it more sense in the automatically-generated section index for the entire Modular Electronics Learning Project archive. Also corrected some spelling errors.

**3-4 February 2025** – minor edits to the Tutorial, including some typographical error corrections. Also added a qualification to the “Applying foundational concepts to a series-parallel circuit” Conceptual Reasoning question that the power supply is constant-voltage.

**21 December 2024** – added a new Diagnostic Reasoning question regarding poor schematic capture design.

**11 September 2024** – added comments on copper balancing when using planes on a multi-layer PCB design.

**9 September 2024** – divided the Introduction chapter into sections, one with recommendations for students, one with a listing of challenging concepts, and one with recommendations for instructors. Also corrected a typographical error where I spelled “ratsnest” as “ratsnets”.

**19 February 2024** – added a new section to the Technical References chapter on near-field versus far-field effects.

**3-6 February 2024** – added more instructor comments for Questions. Also corrected a typo in one of the instructor comments.

**25-27 October 2023** – deleted section title for an un-started section in the Tutorial chapter. Also added comments on PCB dimensions and added a new Tutorial section on “Preserving signal



integrity”. Also added Technical Reference sections on electric and magnetic fields.

**3 February 2023** – added Metro Logic as another professional-grade PCB layout software application.

**2 December 2022** – added notes on using KiCad version 6, as well as some other details to the Tutorial chapter.

**3-30 November 2022** – added more content to the Tutorial as well as the Derivations and Technical References chapters.

**13-16 September 2022** – document first created.

# Index

- Adding quantities to a qualitative problem, 96
- Aggressor, 27
- Altium software, 19
- Ampère's Law, 69
- Annotating diagrams, 95
- Autodesk Eagle software, 19
  
- Balancing, copper, 25
- Balun, 11
- Board layout, 19
- Breadboard, 14
- Breakdown, dielectric, 57
  
- Capacitance, decoupling, 25
- Capacitance, parasitic, 27
- Cathode-ray tube, 60
- CGS, 68
- Characteristic impedance of free space, 71
- Checking for exceptions, 96
- Checking your work, 96
- Code, computer, 103
- Common-mode balun, 11
- Conventional flow notation, 60
- Copper balancing, 25
- Copper distribution, 25
- Copper fill, 24, 25
- Copper plane, 25, 27, 30
- Copper pour, 25, 27, 30
- Coulomb, 57–59
- Cross product, 60
- Cross-product, 59
- Crosstalk, 27
- CRT, 60
- Current balun, 11
  
- Decoupling capacitance, 25
- Design Rule Checking, 19, 22, 38, 41
  
- Dielectric breakdown, 57
- Dielectric strength, 57
- Dimensional analysis, 61, 95
- DIP, 21
- Distribution, copper, 25
- DRC, 19, 22, 38, 41
  
- Eagle software, 19
- EDA, 19
- Edwards, Tim, 104
- Electrically common points, 21
- Electromagnetic wave, 70
- Electromotive force, 67
- Electron flow notation, 60
- emf, 67
  
- Far field region, 70
- Faraday's Law of Electromagnetic Induction, 69
- Field intensity, magnetic, 67
- Fill, copper, 24, 25
- Flux density, magnetic, 63
- Footprint, 21, 35
- Footprints versus symbols, KiCad, 53
  
- Gauss, 63
- gEDA, 19
- Graph values to solve a problem, 96
- Greenleaf, Cynthia, 73
- gschem, 19
  
- Heatsink pad, 26
- Heaviside, Oliver, 69
- How to teach with these modules, 98
- Hwang, Andrew D., 105
  
- IC, 8
- Identify given data, 95
- Identify relevant principles, 95

- Impedance of free space, 71
- Inductance, mutual, 27
- Inductance, parasitic, 25, 27
- Instructions for projects and experiments, 99
- Integrated circuit, 8
- Intermediate results, 95
- Inverse-Square Law, 71
- Inverted instruction, 98
- JLCPCB, 45
- Jumper, thermal, 27
- KiCad, 19, 32
- KiCad footprints versus symbols, 53
- KiCad symbols versus footprints, 53
- Knuth, Donald, 104
- Lamport, Leslie, 104
- Land, 14
- Layout, board, 19
- Left-hand rule, 60
- Lenz's Law, 69
- Limiting cases, 96
- Lorentz force, 59
- Magnetic circuit, 67
- Magnetic field intensity, 67
- Magnetic flux density, 63
- Magnetomotive force, 67
- Maxwell's electromagnetic equations, 69
- Maxwell, James Clerk, 69
- Metacognition, 78
- Metric system, CGS, 68
- Metric system, SI, 68
- Metro Logic software, 19
- Mil, 21
- mmf, 67
- Moolenaar, Bram, 103
- Murphy, Lynn, 73
- Mutual inductance, 27
- Near field region, 70
- Net, 21, 38
- Newton, 57–59, 61
- NOAA, 63
- Open-source, 103
- OrCad software, 19
- Pad, 14, 21
- Pad, heatsink, 26
- Parasitic effect, 27
- Parasitic inductance, 25
- Particle accelerator, 60
- PCB, 8, 90
- pcb, 19
- PCB layout, 19
- Permeability, 63
- Permittivity, 58
- Plane, copper, 25, 27, 30
- Pour, copper, 25, 27, 30
- Poynting vector, 70
- Printed circuit board, 8, 90
- Problem-solving: annotate diagrams, 95
- Problem-solving: check for exceptions, 96
- Problem-solving: checking work, 96
- Problem-solving: dimensional analysis, 95
- Problem-solving: graph values, 96
- Problem-solving: identify given data, 95
- Problem-solving: identify relevant principles, 95
- Problem-solving: interpret intermediate results, 95
- Problem-solving: limiting cases, 96
- Problem-solving: qualitative to quantitative, 96
- Problem-solving: quantitative to qualitative, 96
- Problem-solving: reductio ad absurdum, 96
- Problem-solving: simplify the system, 95
- Problem-solving: thought experiment, 65, 66, 95
- Problem-solving: track units of measurement, 95
- Problem-solving: visually represent the system, 95
- Problem-solving: work in reverse, 96
- Proteus software, 19
- Qualitatively approaching a quantitative problem, 96
- Ratsnest, 21, 23, 38
- Reading Apprenticeship, 73
- Reductio ad absurdum, 96–98
- Reflow soldering, 16
- Region, far-field, 70
- Region, near-field, 70

- Right-hand rule, 60
- Schematic capture, 19
- Schoenbach, Ruth, 73
- Scientific method, 78
- SI, 68
- Simplifying a system, 95
- SMD, 11, 15
- Socrates, 97
- Socratic dialogue, 98
- Solder paste, 16
- Soldering, 14
- Solderless breadboard, 14
- Soldermask, 22
- Solenoid, 65
- SPICE, 73
- Stallman, Richard, 103
- Standing wave, 30
- Superconductor, 58, 64
- Surface mount device, 11
- Surface-mount component, 15
- Symbols versus footprints, KiCad, 53
- Tesla, 59, 61, 63, 65, 66
- Thermal jumper, 27
- Thou, 21
- Thought experiment, 65, 66, 95
- Through-hole component, 14
- Through-hole components, 16
- THT, 14
- Toroid, 66
- Torvalds, Linus, 103
- Trace, 14
- Track, 14
- Units of measurement, 95
- Vector, 59
- Vector cross-product, 60
- Vector, Poynting, 70
- Via, 22
- Victim, 27
- Visualizing a system, 95
- Voltage, 67
- Wave soldering, 14
- Wave, electromagnetic, 70
- Weber, 63
- WMM, 63
- Work in reverse to solve a problem, 96
- World Magnetic Model, 63
- WYSIWYG, 103, 104