PCB Auto-routing

Computer Science Coursework

Edwin Shepherd

Contents

[Analysis 1](#_Toc530325135)

[The Problem 1](#_Toc530325136)

[Research 3](#_Toc530325137)

[Interview 3](#_Toc530325138)

[Perquisites and missing requirements 3](#_Toc530325139)

[Objectives 3](#_Toc530325140)

[Prototyping 4](#_Toc530325141)

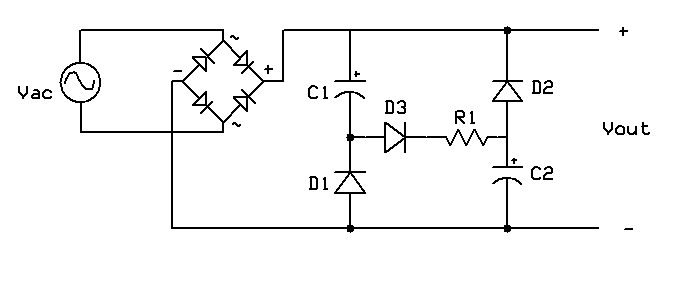
[Bibliography 4](#_Toc530325142)

# Analysis

## The Problem

Electronics engineers develop circuits which are often printed on to a circuit board (PCB), keeping wiring tidy. However routing the wires is a very tedious task. Additionally the wiring must conform to rules or else malfunctions in the circuit may occur. For example, high current tracks must be of a set width, certain components cannot have tracks under them. This is takes more time and is difficult to ensure 100% accuracy.

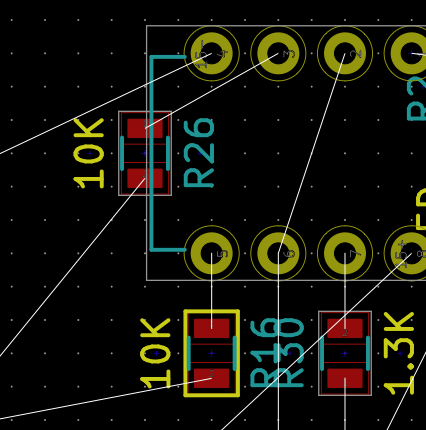
To solve this problem PCB Routing Software is common place in PCB design tools. Its purpose is to automatically route all of the components and to perform “Design Rule Checking” DRC. PCB software usually has two modes one where the circuit is designed:

[](https://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjSqtfQ-NPdAhVIWBoKHcJ_DAsQjRx6BAgBEAU&url=https://en.wikipedia.org/wiki/File:Valley-fill_circuit_schematic_1.png&psig=AOvVaw1u6-1MKZyxtGEkY9-Jxwfn&ust=1537888868345021) Figure 1 a schematic for a basic diode rectifier coupled with a valley-fill circuit - Image credit: WikiMedia available: <https://goo.gl/NJdaqG> , accessed 16:28 24/09/2018

Then another mode where the “nets” highlighted, these are the connections between components (in figure 2 they are blue) and then can be routed by the user (or by software). Finally it’s converted into a file that can be printed.

Figure - Nets are shown in white straight lines. This screen shot is taken from KiCAD's PCBnew software. Image credit; Scanny avalible <https://goo.gl/muyghy> date accessed 29/10/2018

In this project I will solely focus on the challenges that are presented by the routing.

[](https://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj-0Oz0rqveAhUOThoKHfDXAwgQjRx6BAgBEAU&url=https://electronics.stackexchange.com/questions/200093/what-does-yellow-border-around-component-in-kicad-pcbnew-mean&psig=AOvVaw12652bQlXOE6Df1Zygvx7C&ust=1540892724447123)**End user**: Stuart Jessup

## Research

### Interview

After speaking with my client there are a range of areas that my project could support and they were ranked according to value. The list below is ranked.

1. Different track widths
2. Keep out areas
3. Thermal sinks[[1]](#footnote-1)
4. Isometric routing[[2]](#footnote-2)
5. Integration with EDA (Electronic Design Application) software.

### Perquisites and missing requirements

After my interview with my user I investigated other useful features which auto-routers have which are perquisites to being able to implement the features requested. The definition for what auto-routing is on Wikipedia is “*the routing step adds wires needed to properly connect the placed components while obeying all*[*design rules*](https://en.wikipedia.org/wiki/Design_rules)” (Wikimedia Foundation, 2018). This definition clearly points out a few areas which my project would need to support before the requirements suggested by my user:

1. Being able to find a route between two points
2. The route such does not interfere with other routes

Other features which auto routers sell themselves on is the ability to route across multiple layers (Altium LLC, 2018).

### Objectives

The requirements listed above are quite vague and need to be made more precise.

**Being able to find a route between two points** – The route should not waste board space, this means that it is likely to be the shortest possible route and so is “efficient”.

**The route such does not interfere with other routes** – Interfere here means cause a short or a break in another track.

The other requirements have been reworded and can be seen bellow:

1. Finding an efficient routes between points
2. Ensuring that the routes do not violate any the DRC requirements listed in order bellow:
   1. Shorts and breaks do not occur
   2. No tracks cross “keep out area’s”
   3. Areas of board which are specified as thermal sinks meet the minimum size requirements
   4. Track widths are within the tolerances
   5. Routes set to be the same length are correct within tolerance
   6. Integration with EDA (Electronic Design Application) software, KiCAD[[3]](#footnote-3)

## Prototyping

From the prototyping period I needed answers to the following questions:

* What language should I use?
  + What editor am I using
  + How do I compile/run code
* How hard is this problem:
  + What solution and algorithms can I implement
  + How far do I get through my objectives in the week.

**Language:** I do not want to learn a new language for my project and of the languages which I know I feel most confident in python or JS, graphical output is hard in python and so I chose to use JS. The schools have node.js

# Bibliography

Altium LLC. (2018). *BEST AUTOROUTER*. Retrieved November 18, 2018, from Altium: https://www.altium.com/solution/best-autorouter

Wikimedia Foundation. (2018, October 22). *Routing (electronic design automation)*. Retrieved November 18, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Routing\_(electronic\_design\_automation)

1. A Thermal Sink is an area of the PCB which is entirely copper and is used to sink heat from high current devices [↑](#footnote-ref-1)
2. Isometric routing is where multiple routes are routed to be approximately the same length to reduce skew in parallel data transition. [↑](#footnote-ref-2)
3. KiCAD was chosen as the target platform as it is open source and so there is better documentation on the file structure. [↑](#footnote-ref-3)