

A  
Project Report on  
**PCB Design for PWM based DC Motor Speed  
Controller using Ki Cad**

**PRODUCT BUILD-A-THON**

By the Initiative of  
**SmartInternz & Voltera**

In the year 2021  
Submitted by  
**CHINTHA VINAY [182P5A0209]**

**Chaitanya Bharathi Institute of Technology Proddatur  
Department of Electrical and Electronics Engineering**

**Electronic Product Designing and Electronic Product Build-A-Thon**

*Powered by Voltera & SmartInternz  
(3 Day's Hands on Training + 7 Day's Mentor Guided Project Development)*

## ACKNOWLEDGEMENT

I am very much thankful to Voltera & SmartInternz, for their dedicated effort in providing mentorship and constant encouragement in completing this project.

We with a deep sense of gratitude and heart full thanks to **Sri. Katarina Ilić Co-founder, Voltera Inc, Canada, Sri. Rama Krishna Dasari CEO, Efftronics System Pvt. Ltd., Sri. Ravi Gujjula Chief General Manager-Technical, AP State Skill Development Corporation (APSSDC), Sri. Ravikumar Annavarapu Immdt Past Chairman, IETE, Sri. Amarender Katkam Founder & CEO, SmartBridge.**

I am thankful to the Management of **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, PRODDATUR** especially **Mr. N. VARA PRASAD**, Associate professor & Head of the department **Electrical and Electronics Engineering**, for permitting us to use facilities for doing this project work and boundless moral support.

I thank all the members of Voltera & SmartInternz Team, who has helped me in completing the project successfully. Last but not least I would like to thank parents for what I am today and finally my friends who helped me in successful completion of my project and for the betterment of my caree

# List figures

FIGURE NO	NAME OF THE FIGURE	PAGE NO
1.1	Workflow	5
2.1	Symbol editor	8
2.2	Symbol design	8
2.3	Schematic of the project	
3.1	Library view	11
3.2	Footprint of the project	14
4.1	Gerber file of the project	17
5.1	Future Scope	19

# List of tables

TABLE NO	NAME OF THE TABLE	PAGE NO
1.1	KiCad Extension	4

# TABLE OF CONTENTS

S.NO.	CONTENTS	PAGE NO
	List of figures	i
	List of Tables	ii
	Abstract	2
<b>CHAPTER 1: INTRODUCTION TO KI CAD</b>		<b>3-5</b>
	1.1. Introduction to KiCad	4
	1.2. Downloading KiCad	4
	1.3. KiCad workflow	4
<b>CHAPTER 2: INTRODUCTION TO EMBEDDED SYSTEMS</b>		<b>7-9</b>
	2.1. Make schematic symbols	7
	2.2. Symbol Design	8
<b>CHAPTER 3: FOOTPRINTS TO THE SCHEMATIC</b>		<b>10-14</b>
	3.1. Make component foot print	11
	3.2. Portability of KiCad project files	12
<b>CHAPTER 4: GENERATION OF GERBER FILES&amp; FINISHING THE DESIGN</b>		<b>15-17</b>
	4.1. Gerber files	16
	4.2. PCB manufacturing-Gerber files	16
	4.3. Generation of Gerber files in a PCB design	17
	4.4. Gerber file of the Project	17
<b>CHAPTER 5: Conclusion &amp; Future Scope</b>		<b>16-20</b>
	5.1. Conclusion	19
	5.2 Future Scope	20
	5.3. Reference	21

# **ABSTRACT**

Electronic Product Build-A-Thon, Unique program to build skills in Electronic Product Development is an exclusive skill building initiative by SmartBridge in association with Voltera (Canada) to enable Students & Educators with electronic circuit designing skills. This program is a learning by doing event packed with hands-on training, Project development and mentoring sessions powered by SmartInternz & Voltera.

Here I choose a simple project from different offered projects, PWM based DC Motor Speed Controller in which the speed of the motor would be adjusted by the application of 555 timer. The main motto of this project to develop a product using computer aided drawing tools (KiCad)

1. The Schematic diagram of the project
2. Screenshots & Gerber Files (Git Hub repository)
3. Video demonstration of the entire project
4. Project Document

## **Chapter:1**

### **Introduction to KiCad**

# 1.Introduction to KiCad

## 1.1 Introduction to KiCad

KiCad is an open-source software tool for the creation of electronic schematic diagrams and PCB artwork. Beneath its singular surface, KiCad incorporates an elegant ensemble of the following stand-alone software tools:

Program name	Description	File extension
KiCad	Project manager	*.pro
Eeschema	Schematic and component editor	*.sch, *.lib, *.net
Pcbnew	Circuit board and footprint editor	*.kicad_pcb, *.kicad_mod
GerbView	Gerber and drill file viewer	\*.g\*, *.drl, etc.
Bitmap2Component	Convert bitmap images to components or footprints	*.lib, *.kicad_mod, *.kicad_wks
PCB Calculator	Calculator for components, track width, electrical spacing, color codes, and more...	None
Pl Editor	Page layout editor	*.kicad_wks

**Table 1.1 KiCad extension**

## 1.2 Downloading and installing KiCad

KiCad runs on GNU/Linux, Apple macOS and Windows. You can find the most up to date instructions and download links at:

<http://www.kicad-pcb.org/download/>

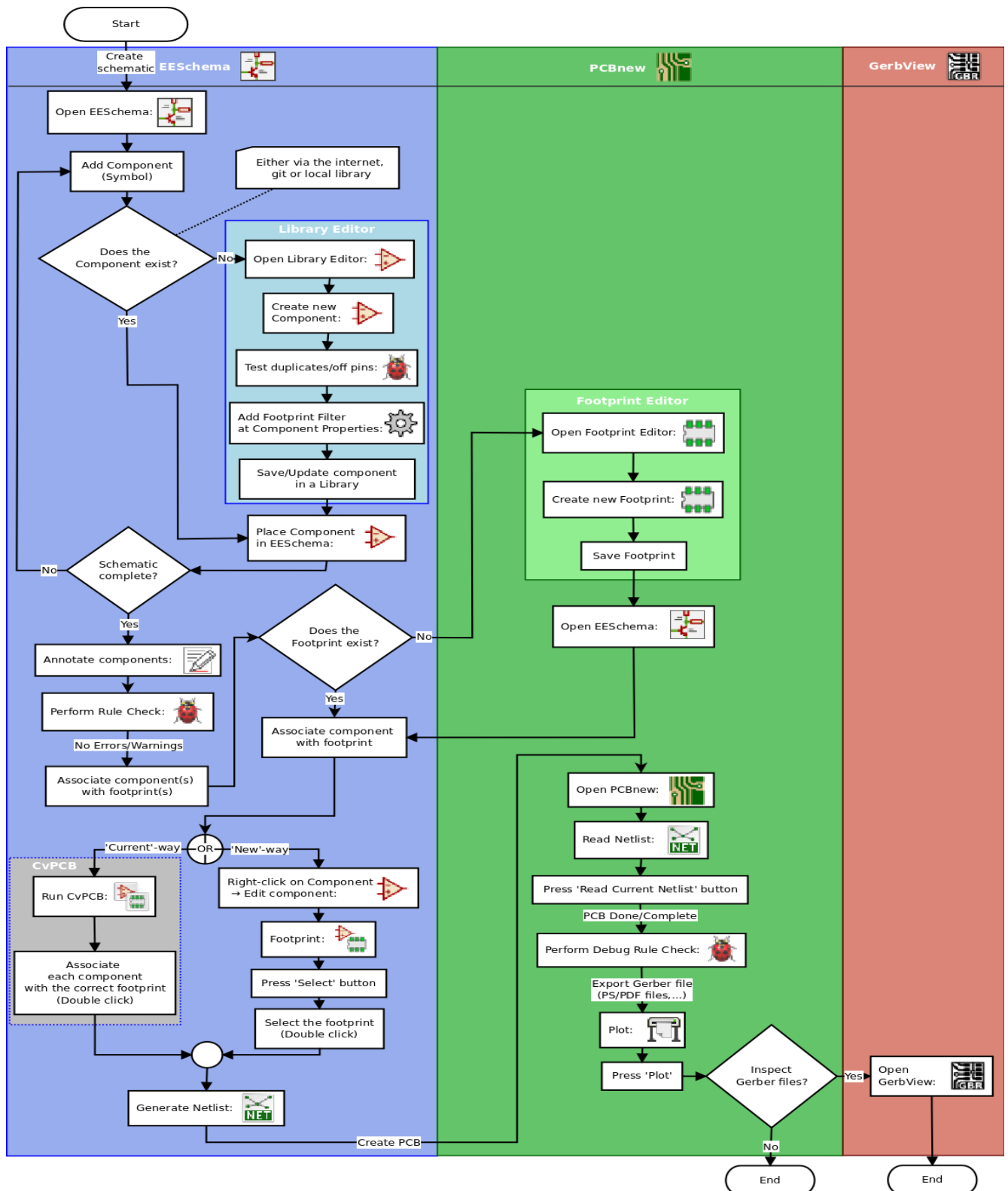
## 1.3 KiCad Workflow

Despite its similarities with other PCB design software, KiCad is characterised by a unique workflow in which schematic components and footprints are separate. Only after creating a schematic are footprints assigned to the components. The KiCad workflow is comprised of two main tasks: drawing the schematic and laying out the board. Both a schematic component library and a PCB footprint library are necessary for these two tasks. KiCad includes many components and footprints, and also has the tools to create new ones.



# PCB Design for PWM based DC Motor Speed Controller using Ki Cad

In the picture below, you see a flowchart representing the KiCad workflow. The flowchart explains which steps you need to take, and in which order. When applicable, the icon is added for convenience.



## **Chapter:2**

### **2.Schematic of the Project**





## 2. Schematic of the Project

### 2.1 Make schematic symbols in KiCad

Sometimes a symbol that you want to place on your schematic is not in a KiCad library. This is quite normal and there is no reason to worry. In this section we will see how a new schematic symbol can be quickly created with KiCad. Nevertheless, remember that you can always find KiCad components on the Internet.

In KiCad, a symbol is a piece of text that starts with 'DEF' and ends with 'ENDDEF'. One or more symbols are normally placed in a library file with the extension .lib. If you want to add symbols to a library file you can just use the cut and paste commands of a text editor

#### Using Component Library Editor

1. We can use the *Component Library Editor* (part of *Eeschema*) to make new components. In our project folder 'tutorial1' let's create a folder named 'library'. Inside we will put our new library file *myLib.lib* as soon as we have created our new component.
2. Now we can start creating our new component. From KiCad, start *Eeschema*, click on the 'Library Editor' icon  and then click on the 'New component' icon . The Component Properties window will appear. Name the new component 'MYCONN3', set the 'Default reference designator' as 'J', and the 'Number of units per package' as '1'. Click OK. If the warning appears just click yes. At this point the component is only made of its labels. Let's add some pins. Click on the 'Add Pins' icon  on the right toolbar. To place the pin, left click in the centre of the part editor sheet just below the 'MYCONN3' label.
3. In the Pin Properties window that appears, set the pin name to 'VCC', set the pin number to '1', and the 'Electrical type' to 'Power input' then click OK.
4. Place the pin by clicking on the location you would like it to go, right below the 'MYCONN3' label.
5. Repeat the place-pin steps, this time 'Pin name' should be 'INPUT', 'Pin number' should be '2', and 'Electrical Type' should be 'Passive'.
6. Repeat the place-pin steps, this time 'Pin name' should be 'GND', 'Pin number' should be '3', and 'Electrical Type' should be 'Passive'. Arrange the pins one on top of the other. The component label 'MYCONN3' should be in the centre of the page (where the blue lines cross). Next, draw the contour of the component. Click on the 'Add rectangle' icon . We want to draw a rectangle next to the pins, as shown below. To do this, click where you want the top left corner of the rectangle to be (do not

hold the mouse button down). Click again where you want the bottom right corner of the rectangle to be.

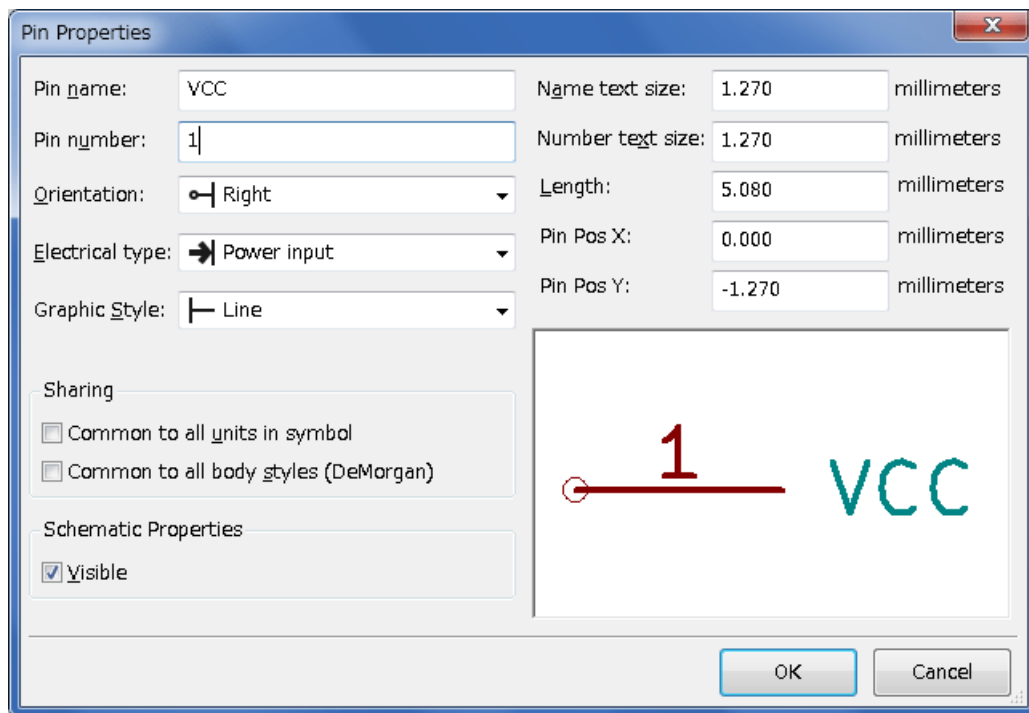


Figure 2.1 Symbol Editor

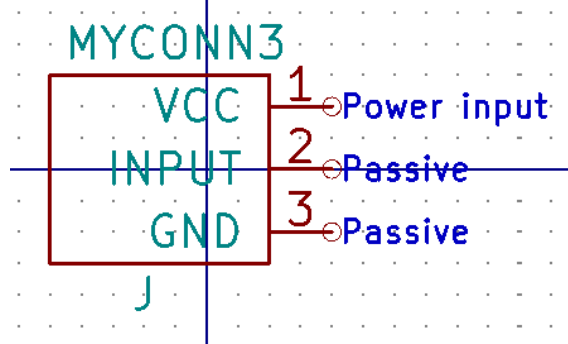



Fig 2.2 Symbol design

7. If you want to fill the rectangle with yellow, set the fill colour to 'yellow 4' in **Preferences** → **Select color scheme**, then select the rectangle in the editing screen with [e], selecting 'Fill background'.
8. Save the component in your library *myLib.lib*. Click on the 'New Library'
9. , navigate into *tutorial1/library/* folder and save the new library file with the name *myLib.lib*.



## **Chapter: 3**

### **Footprints to the Schematic**

## 3. Foot Prints to the Schematic





### 3.1 Make component footprints



Unlike other EDA software tools, which have one type of library that contains both the schematic symbol and the footprint variations, KiCad *.lib* files contain schematic symbols and *kicad\_mod* files contain footprints. *Cypcb* is used to map footprints to symbols.

As for *.lib* files, *kicad\_mod* library files are text files that can contain anything from one to several parts.

There is an extensive footprint library with KiCad, however on occasion you might find that the footprint you need is not in the KiCad library. Here are the steps for creating a new PCB footprint in KiCad:

#### Using Footprint Editor

1. From the KiCad project manager start the *Pcbnew* tool. Click on the 'Open Footprint Editor' icon  on the top toolbar. This will open the 'Footprint Editor'.
2. We are going to save the new footprint 'MYCONN3' in the new footprint library 'myfootprint'. Create a new folder *myfootprint.pretty* in the *tutorial1/* project folder. Click on the **Preferences** → **Footprint Libraries Manager** and press 'Append Library' button. In the table, enter "myfootprint" as Nickname, enter "\${KIPRJMOD}/myfootprint.pretty" as Library Path and enter "KiCad" as Plugin Type. Press OK to close the PCB Library Tables window. Click on the 'Select active library' icon  on the top toolbar. Select the 'myfootprint' library.
3. Click on the 'New Footprint' icon  on the top toolbar. Type 'MYCONN3' as the 'footprint name'. In the middle of the screen the 'MYCONN3' label will appear. Under the label you can see the 'REF\*' label. *Right click on 'MYCONN3' and move it above 'REF\*'.* Right click on 'REF\_\_\*', select 'Edit Text' and rename it to 'SMD'. Set the 'Display' value to 'Invisible'.
4. Select the 'Add Pads' icon  on the right toolbar. Click on the working sheet to place the pad. Right click on the new pad and click 'Edit Pad'. You can also use [e].
5. Set the 'Pad Num' to '1', 'Pad Shape' to 'Rect', 'Pad Type' to 'SMD', 'Shape Size X' to '0.4', and 'Shape Size Y' to '0.8'. Click OK. Click on 'Add Pads' again and place two more pads. If you want to change the grid size, **Right click** → **Grid Select**. Be sure to select the appropriate grid size before laying down the components.

6. Move the 'MYCONN3' label and the 'SMD' label out of the way so that it looks like the image shown above.
7. When placing pads, it is often necessary to measure relative distances. Place the cursor where you want the relative coordinate point  $(0,0)$  to be and press the space bar. While moving the cursor around, you will see a relative indication of the position of the cursor at the bottom of the page. Press the space bar at any time to set the new origin.
8. Now add a footprint contour. Click on the 'Add graphic line or polygon' button  in the right toolbar. Draw an outline of the connector around the component.
9. Click on the 'Save Footprint in Active Library' icon  on the top toolbar, using the default name MYCONN3.

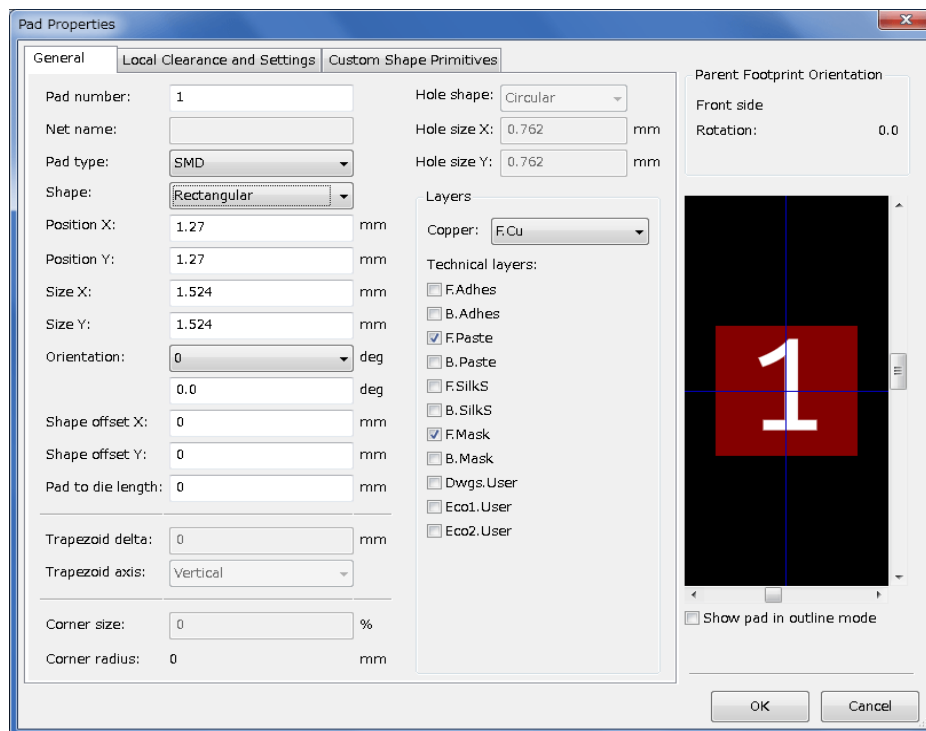


Fig 3.1 Library View

### 3.2 Portability of KiCad project files

When you have a KiCad project to share with somebody, it is important that the schematic file `.sch`, the board file `.kicad_pcb`, the project file `.pro` and the netlist file `.net`, are sent together with both the schematic parts file `.lib` and the footprints file `.kicad_mod`. Only this way will people have total freedom to modify the schematic and the board.

With KiCad schematics, people need the `.lib` files that contain the symbols. Those library files need to be loaded in the *Eeschema* preferences. On the other hand, with boards



(*.kicad\_pcb* files), footprints can be stored inside the *.kicad\_pcb* file. You can send someone a *.kicad\_pcb* file and nothing else, and they would still be able to look at and edit the board. However, when they want to load components from a netlist, the footprint libraries (*.kicad\_mod* files) need to be present and loaded in the *Pcbnew* preferences just as for schematics. Also, it is necessary to load the *.kicad\_mod* files in the preferences of *Pcbnew* in order for those footprints to show up in *Cvpcb*.

If someone sends you a *.kicad\_pcb* file with footprints you would like to use in another board, you can open the Footprint Editor, load a footprint from the current board, and save or export it into another footprint library. You can also export all the footprints from a *.kicad\_pcb* file at once via **Pcbnew** → **File** → **Archive** → **Footprints** → **Create footprint archive**, which will create a new *.kicad\_mod* file with all the board's footprints.

```
tutorial1/
|-- tutorial1.pro
|-- tutorial1.sch
|-- tutorial1.kicad_pcb
|-- tutorial1.net
|-- library/
|   |-- myLib.lib
|   |-- myOwnLib.lib
|   \-- myQuickLib.lib
|
|-- myfootprint.pretty/
|   \-- MYCONN3.kicad_mod
|
\-- gerber/
    |-- ...
    \-- ...
```

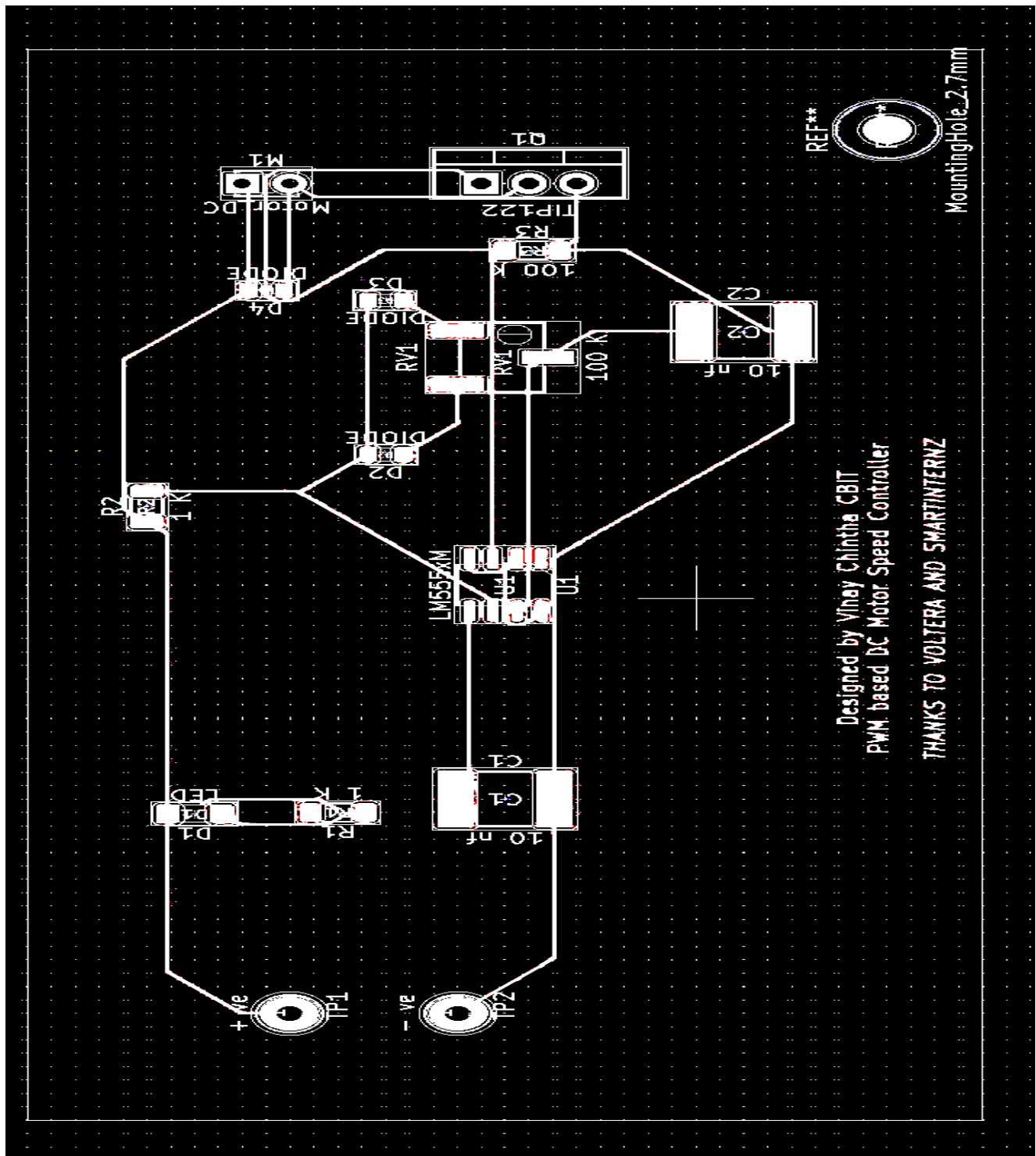


Fig 3.2 Foot Print of the Project

## **Chapter: 4**

### **Generation of the Gerber Files & Finishing the design**

## **4. Generation of the Gerber Files & Finishing the design**

### **4.1 Gerber files**

Gerber files are open ASCII vector format files that contain information on each physical board layer of your PCB design. Circuit board objects, like copper traces, vias, pads, solder mask and silkscreen images, are all represented by a flash or draw code, and defined by a series of vector coordinates. These files are used by PCB manufacturers to translate the details of your design into the physical properties of the PCB.

The Gerber files are typically generated by the PCB design software that you're using, although the process of doing so will vary with each CAD tool. Most of today's Gerber files are generated according to the RS-274X Gerber format, which supersedes obsolete formats such as the RS-274-D Gerber standard. As a text file, Gerber data does not have to have a specific identifying file name but is often given a common extension such as .gb, or .gbr.

### **4.2 PCB Manufacturing – Gerber files**

The PCB manufacturing technology used today has come a long way over the past decades. Previously, vector photoplotters were used to create the tooling film used in the PCB manufacturing process. A focused light conducted through an aperture was used to expose the film to create the flashes and draws for each individual pad and trace.

There was only a minimal set of apertures available and designers had to be creative in restricting their flashes and lines to only the apertures that were available to them. These older vector machines have now been replaced by a newer breed that uses a raster laser process to expose the film. For larger pieces of film that had a lot of line drawing on it, a vector photoplotter could take many hours. This time has been cut down to a matter of minutes with a laser plotter.

Gerber file history can be traced back to the needs of the original vector photoplotters. In order to give the plotter its instructions, a Gerber file contained minimal plotter configuration information, and X/Y coordinates followed by a flash or drawing command and which aperture position to use. The Gerber data has increased in functionality over the years; it now includes additional configuration information as well as macro and aperture definitions.

The laser plotters used today still use the same Gerber information, but the aperture restrictions of the older vector plotters no longer apply. The laser plotters convert the Gerber coordinates into a raster file, and that information instructs the laser plotter on how, where, and what is to be created on the films. For example, the aperture definitions convey the thickness and sizes of the traces and pads while the drawing commands define whether lines, polygon fills or flashes are to be created. Then the laser sweeps across the film exposing the image as it goes.

Time does not sit still though, and the world of Gerber files continues to change. Circuit board manufacturers have now begun using direct laser imaging to create PCB images directly onto the copper, and bypassing the need for film.

### 4.3 Generate Gerber Files in a PCB Design

After your design is complete and you've done the final check, the next step will be to generate the Gerber files for your PCB manufacturer. The process of doing so varies according to the PCB design software that you're using. Some older tools may take many steps to set up and generate files, but today, most CAD packages have simplified the process of creating these files.

Typically, the Gerber files you will need to produce will be an individual file for each physical layer of the board. If your PCB design is a six-layer board with four signal layers and two planes for power and ground, then you will need to output those layers into six Gerber files. Additionally, you will need to generate a separate Gerber file for the top and bottom solder mask layers, the top and bottom silkscreen layers (if required), and the top and bottom solder paste layers (if required).

Although not a Gerber file, an NC drill file is usually created along with the Gerber files. This file instructs the drilling machines used by the board fabricator where to drill the holes in the board. This file is very similar to a Gerber formatted file in that it also contains drill size information and vector data for the different drill locations.

### 4.4 Gerber file of the project

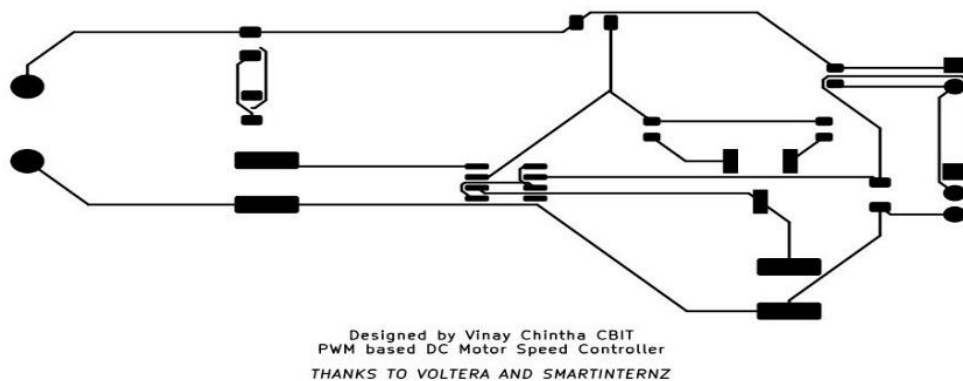


Figure 4.1 Gerber View of the Project

## **Chapter: 5**

### **Conclusion& Future Scope**

# 5. Conclusion & Future Scope

## 5.1 Conclusion

*“The future of PCB designer is truly very bright.”*

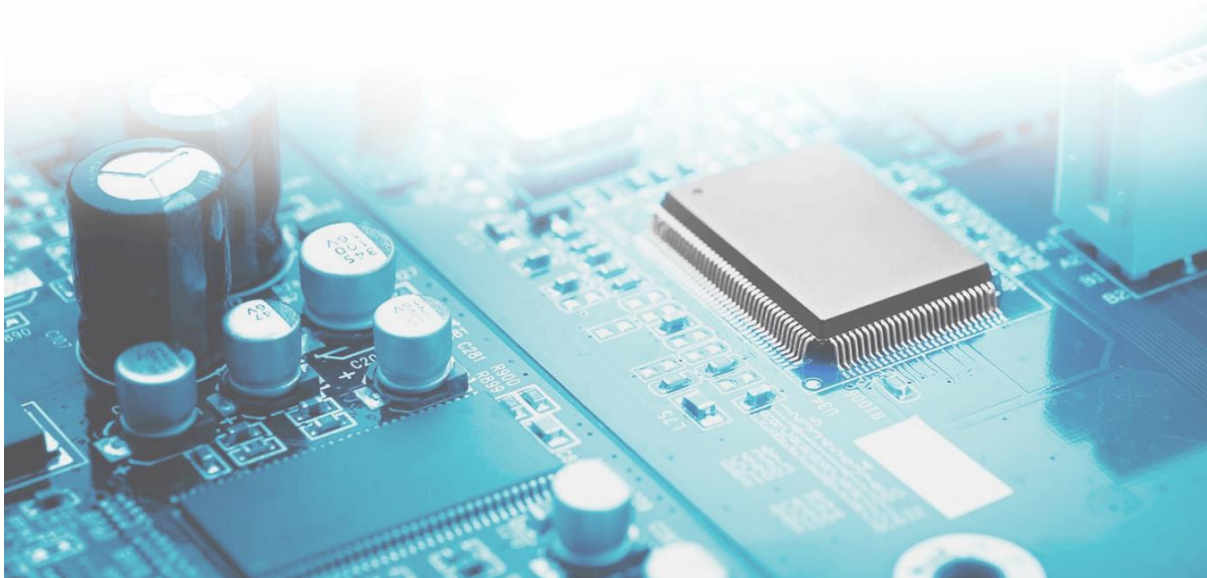
There are plenty of indications from the market that show there is clearly a lot of opportunities in PCB Designing field. With competitive growth in consumer electronics and automotive industries, the thrive for making new cool, “smart” products is increasing. With IOT taking over the market, huge number of small products with smart features are being made. For all these products, designing of circuits on small PCB is becoming a must.

The technological and electronic advancements that you ‘ll see in the future is totally miraculous and astounding. When you purchase a new item, think of how incredible the PCB makes it function.

Well, my confidence in the PCB designing field is unwavering as until new products are coming into market (almost daily a new idea for a product is made), the scope of PCB Designing will be there.

## 5.2 Future Scope

Modern PCBs are produced at incredible speed with incredible complexity, but there is always room for development. Regardless of the shape of the PCB itself or the accessories attached directly to the board, consumers are constantly pushing for new and varied PCB and PCB functions



**Figure 5.1 Future Scope**

Embedded Field is growing due to opportunities in IoT, Computer Vision, AI, AR and those are driving semiconductor companies to come up with more high performing chips, more integrated chips, and you need PCBs in every electronics, embedded product.

as new challenges arise for manufacturing companies, there is ample room for growth in the manufacturing process itself. That's why most forecasts for the future of printed circuit boards focus on the following areas:

### 1. PCB board

- Home Appliances
- Medical Equipment

### 2. 3D printing electronic

- novel design
- Increased efficiency
- Environment friendly

### 3. PCB Auto placers

- Auto Router
- Automatic placement

### 4. High speed function

- digital signal
- analog signal

### 5. Focus on Flexible PCB

- LED lights
- Wearable Technology
- flexible display
- Medical Equipment

### 6. Biodegradable PCB



## Reference

- [1] <https://smartinternz.com/voltera-electronic-product-build-a-thon>
- [2] [https://docs.kicad.org/5.1/en/getting\\_started\\_in\\_kicad/getting\\_started\\_in\\_kicad.html](https://docs.kicad.org/5.1/en/getting_started_in_kicad/getting_started_in_kicad.html)
- [3] <https://docs.kicad.org/5.1/en/kicad/kicad.html>
- [4] <https://docs.kicad.org/5.1/en/gerbview/gerbview.html>
- [5] [https://docs.kicad.org/5.1/en/pl\\_editor/pl\\_editor.html](https://docs.kicad.org/5.1/en/pl_editor/pl_editor.html)