



Expertise
and insight
for the future

Sachin Bashyal

PCB Development for an IoT System

Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Bachelor's Thesis

26 November 2018

Author Title	Sachin Bashyal PCB Design for an IoT System
Number of Pages Date	32 pages 26 November 2018
Degree	Bachelor of Engineering
Degree Programme	Electronics
Professional Major	
Instructors	Anssi Ikonen, Senior Lecturer
<p>The main objective for the project was to develop PCB for a prototype of the IoT system of an old home light and to study the three free design tools that can be used for this project. The first part of the project was to study free PCB design tools available on the Internet and to select the best one among them for this project, which was initially carried out. Second and the main objective was to design and manufacture the professional PCB layout according to the given requirements and information. This project aims to develop a part of the smart home project.</p> <p>The design was carried in two phases, namely schematic design and layout design. All the components required for the projects were custom created even some of them were available in the software library. The main idea behind not using the available parts from the library was to explain and understand the detailed process of PCB development task. The designed PCB will act as the IoT system prototype after feeding the codes, which were not the part of the project. The total system is designed in such way that the old light system of a house will turn to IoT so that it can be controlled by a user over Internet network being able to customize the light colors, brightness and also able to react according to the surrounding room temperature. The sole purpose of this project was to develop a hardware PCB system required for this project. The project was carried out using the tool called KiCad among two other tools, namely: DesignSpark and ExpressPCB. The design is ready for milling and testing.</p> <p>The result of the project is to manufacture a functioning PCB. The design was carried out using the schematic design and layout design which will be eventually used to print out the board and place the require components on them and finally feeding the codes in order to function as required functioning IoT system.</p>	
Keywords	KiCad, PCB design, Schematic, Layout, Fabrication, Milling, IoT system

Contents

List of Abbreviations

1	Introduction	1
2	Free PCB Design Tools For Designing Printed Circuit Board (PCB)	2
2.1	KiCad	2
2.1.1	KiCad Project Manager	2
2.2.2	Eeschema	3
2.2.3	CvPcb	3
2.2.4	Pcbnew	3
2.2.5	GerberView	3
2.2.6	Bitmap2Component	3
2.2.7	PCB Calculator	4
2.2	DesignSpark PCB	4
2.3	Express PCB	5
3	Comparison of Features Among Three PCB Design Tools	5
3.1	KiCad Features	5
3.2	DesignSpark Features	6
3.3	ExpressPCB Features	6
4	KiCad as PCB Design Tool	6
4.1	Introduction	6
4.2	Basic Workflow	7
4.3	Eeschema	7
4.4	PcbNew	8
5	PCB Development for an IoT system prototype	9
5.1	Introduction	
5.2	Design Framework	10
5.2.1	Schematic Design	11
5.2.1.1	Creating a new Library	12
5.2.1.2	Creating Photonh	16
5.2.2	Layout Design	20
5.2.2.1	Creating a New Footprint and New Footprint Library	20
5.2.2.2	Editing Pad Properties	21
5.2.2.3	Creating Array	23
5.3	Creating Amplifier (INA118P)	26
5.4	Creating Connector and Resistor Footprints	29
5.4.1	Creating connector footprints	29
5.4.2	Creating Resistor Footprints	30
6	Conclusion	33
	Reference	34

List of Abbreviations

IoT Internet of things

PCB Printed Circuit Board

RFID Radio frequency identification

NFC Near field communication

WSANs Wireless sensor and actuators networks

THT Through hole technology

SMD Surface Mounted Device

1 Introduction

IoT consists of a wide range of communication standards, protocols and data formats so that the IoT environment is heterogeneous, decentralized and complex. It consists of technology such as near field communication (NFC), radio frequency identification (RFID), wireless sensor and actuators networks (WSANs). Basically, it can be said to be a system where a network of physical devices, vehicles, home appliances are embedded with software, electronics, actuators, which enable these things to connect, collect and exchange data. [1.]

An IoT ecosystem can be defined as a collection of smart devices connected in Internet network which use embedded processors, sensors and hardware to collect and send data so that they can interact. These devices also communicate with other related devices and act on the information they get from one another. Figure 1 shows the most common IoT applications in everyday life. [1.]

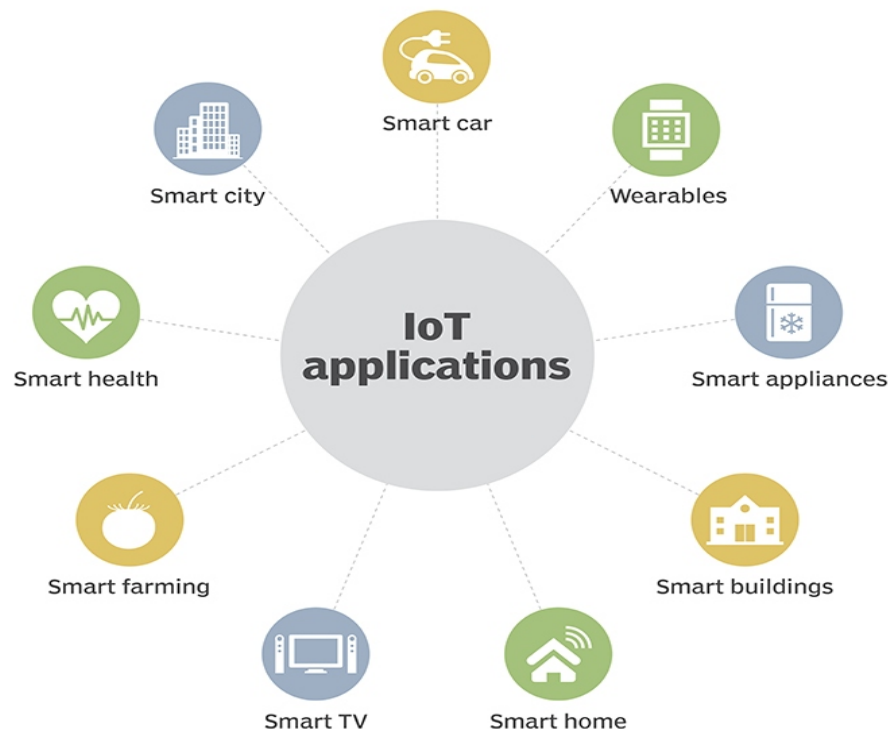


Figure1: IoT uses and applications [1.]

2 Free Design Tools for Designing Printed Circuit Board(PCB)

There are plenty of free design tools on the Internet for designing a PCB. The selection of software depends on characteristics of the project. If the project is simple and does not require a lot of library component features, it can be carried out using a light software which is easy to download, install and operate and such software does not occupy a lot of space on the computer disk. Some smaller projects can also be carried out online, without even bothering to download and install on the computer. [7.]

Among almost ten free PCB design tools, below are listed top 3 best free ones:

- KiCad
- DesignSpark PCB
- ExpressPCB

2.1 KiCad

KiCAD is an open-source software tool for the creation of electronic schematic diagrams and PCB artwork. Beneath its singular surface, it incorporates an elegant ensemble of components such as eeschema, PCB calculator, GerbView, PI editor and so on these are briefly discussed below:

2.1.1 KiCad Project Manager

This is the main component of the software tool, which is a project manager. It contains .pro extension and the file contains few parameters for the current project, including the component library list. [4.]

2.1.2 Eeschema

Eeschema is the component used for schematic file editor and it contains various file types, namely .sch, .lib, .net. .sch files contain schematic files, which do not contain the components themselves. .lib files contain schematic component library files, containing the component descriptions: graphic shape, pins, fields. And .net files contain netlist file created by the schematic and are read by the board editor. This file is associated with the .cmp file, for users who prefer a separate file for the component/footprint association. [4.]

2.1.3 CvPcb

This tool allows matching the components from the schematic file to the footprint components while designing the printed circuit board. The files stored on the footprint library are listed here so that the corresponding components from the schematic diagram are matched to perform the desired function in a specific design. This association is added to the netlist file created by the schematic capture program eeschema. It is simply called a footprint selector and has .net file extension. [4.]

2.1.4 PcbNew

This tool is for editing the circuit board. The actual design is made on this tool which contains all information about layout and the file format is .kicad_pcb. [4.]

2.1.5 GerbView

GerbView is used to view gerber files, which allow the file for fabrication. It has the file format .gbr. Gerber is also the standard image input format for all bare board fabrication equipment needing image data, such as photoplotters, legend printers, direct imagers or automated optical inspection (AOI) machines and for viewing reference images in different departments. [4.]

2.1.6 Bitmap2Component

This tool helps to convert bitmap images to components or footprints. This file contains the format .lib, .kicad_mod, .kicad_wks. This is a utility tool, which allows a user to

create a component for logos. It creates a schematic component or a footprint from a bitmap picture. [4.]

2.1.7 Pcb Calculator

This tool is a basic calculator for component, which allows measuring track width versus current, electrical spacing, color code, transmission lines. This tool also comes under utility tools. [4.]

2.1.8 PI Editor

This is a utility tool used for page layout editor. It contains page layout description files, for people who want a worksheet with a custom look and the file extension is .kicad_wks. [4.]

2.2 DesignSpark PCB

DesignSpark PCB is one of the most popular and easily accessible electronics design software tools. The program is easy to use and learn. DesignSpark has so many features so it can capture schematics easily, it can design PCB board, improve layout, and so on with the same program. The tools are free to use for all designers. [5.]

Unlike KiCad, DesignSpark has various components, which allow the user to work perfectly giving a lot of free features. This also has a schematic editor where a user can draw diagrams and connections with multiple schematic sheets combining together to form the complete design. Moreover, it allows a user to add third-party libraries needed for the design work. Similarly, this also has autorouting features that automatically places routes between components on a layout and produces Gerber and excellon drill files. [3.]

2.3 ExpressPCB

This is also a similar kind of software as KiCad and Designspark. However, differs a little with the design rules, graphics and available component and component libraries. This also has two tools for designing such as expressSCH for drawing schematic and ex-pressPCB for designing circuit boards. Most of the drawing techniques are almost similar. [6.]

3 Comparison Of Features Among Three PCB Design Tools

Even though top three free design tools are not much different but they still stand out among each other with some major important features.

3.1 KiCad Features

- This is an open source program which allows user edit file formats and library elements
- This tool provides rule checking such as the ability to set and pin properties
- Component library links with community maintained footprints and symbols
- Allows viewing the design in 3D model
- Contains simple, complex and flat hierarchy schemes
- Generates Gerber files for fabrication
- No need of registration for free use [8.]

3.2 DesignSpark Features

- No size limitation for design

- Generates Gerber and ODB++ files for fabrication
- Wide range of components around only 32 countries around world and 2,500 manufacturer
- Library connected to RS components
- Needs registration although it is free
- Unlike KiCad, this tool allows to create own components and footprints. [3.]

3.3 ExpressPCB Features

- Some features are outdated and cause a problem while importing components from schematic
- Only file format supported for fabrication is proprietary with no Gerber files which creates vendor lock-in and costly boards
- Mechanical features are limited such as maximum hole size is 0.25 inches and no support for non-plated holes in boards interior
- This is unable to create internal slots and openings. [9.]

4 KiCad as PCB Design Tool

4.1 Introduction to KiCad

Among three listed software, comparing all those listed features, KiCad was selected best for this project as this is open source software and has a wide range of libraries. The components which are not available in the library can be created easily. Moreover, there are plenty of free tutorials available for the user guide, which explain even the basic steps of operation. [4.]

Kicad uses two types of library symbols .lib and .pretty for footprints. Symbols are used to draw schematics and after creating symbols, the proper footprints, which suit the design are assigned. [4.]

4.2 Basic Workflow

The basic workflow in KiCad is explained below:

Step 1: Create Project

Step 2: Create Schematic on eeschema. If the required component symbol does not exist on the library, create a component from library editor.

Step 3: Assign Footprints to the symbols used in a schematic. If the required footprint does not exist then create a new footprint from footprint editor.

Step 4: Create netlist, annotate components and associated components with related footprints.

Step 5: Create the design board by reading netlist from pcbnew window, importing netlist from eeschema.

Step 6: Test the design using a design rule check and generate production files. [4.]

4.3 Eeschema

Eeschema is a schematic capture software distributed as a part of KiCad. Eeschema allows a user to draw, control, manage library and access to the PCB design software. Eeschema cooperates with PcbNew, which is KiCad's printed circuit design software. This can also export netlist files, which list all the electrical connections, for other packages. Eeschema includes a symbol library editor, which can create and edit symbols and manage libraries. Figure 2 show below is the window to show a complete schematic circuit. [4.]

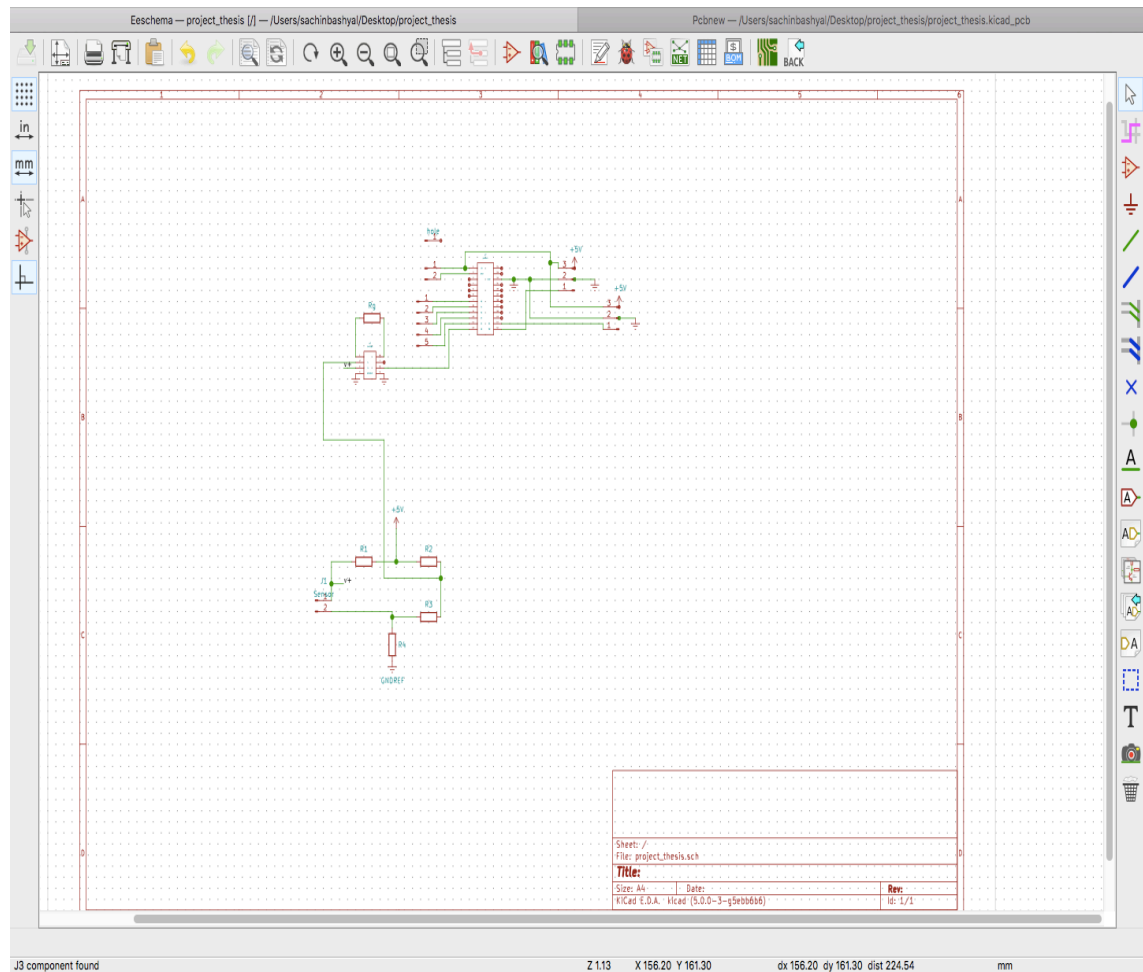


Figure 2: Eeschema window with a functional schematic

4.4 PCBnew

This is the tool where the actual design of a board is performed. This is used in association with eeschema and schematic design to create a printed circuit board. This manages the libraries of footprints. Footprints can be assigned to the components corresponding to the schematic created in eeschema, either from the existing library footprints or by creating new footprints, using footprint editor. The required footprints are automatically listed during a reading of netlist, and the components from schematic have to be matched with respective footprint components, in order to draw the circuit. Footprints can be changed at any time of design and updated by regenerating netlist and re-reading it from PCBnew. The figure 3 below is the window of PCB layout, which shows complete routed circuit design for a PCB board. [4.]

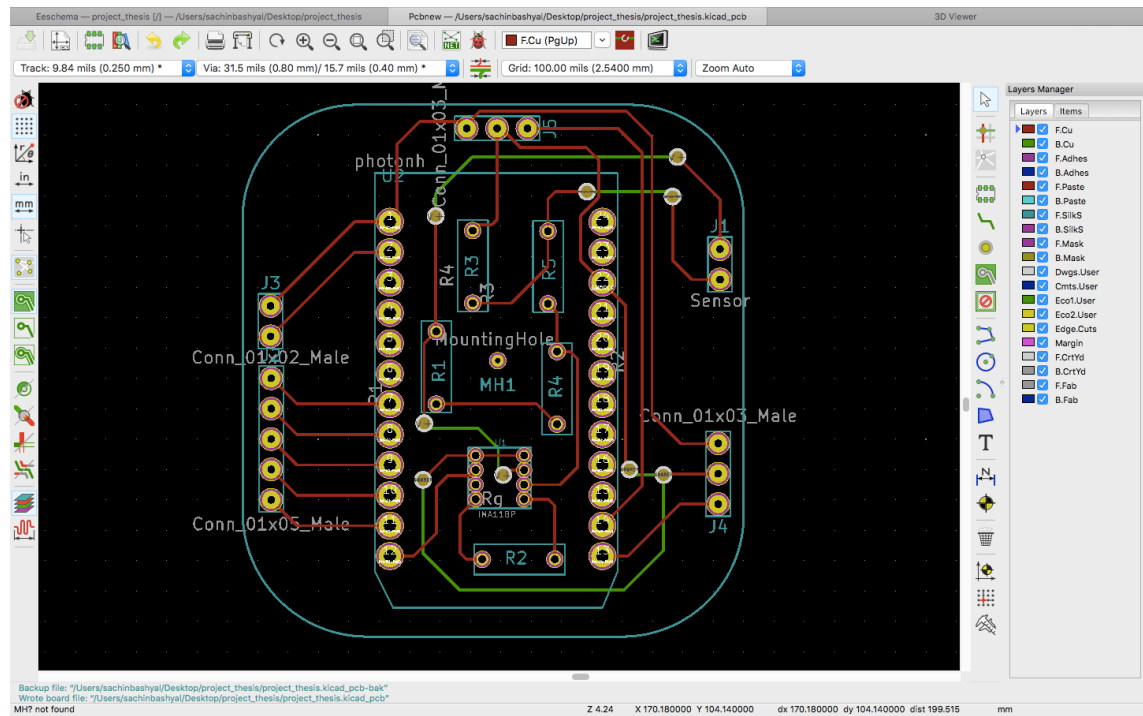


Fig 3: A PCBnew window with a complete design board

5 PCB Development for an IoT system prototype

5.1 Introduction to the project

As the world is growing rapidly with innovation and technology, every aspect of human life system is getting advanced. Among them, one astonishing concept is smart home. A smart home consists of every home appliances working as smart device and connected to the Internet creating a network in order to exchange data, widely known as IoT (Internet of Things). This project is a part of a smart home concept where an old technology of lighting system is converted into an IoT system so that it can be connected and controlled via internet on various basis. The following figure 4 shows an old light holder, which is upgraded to the IoT system without changing the design of the light holder.

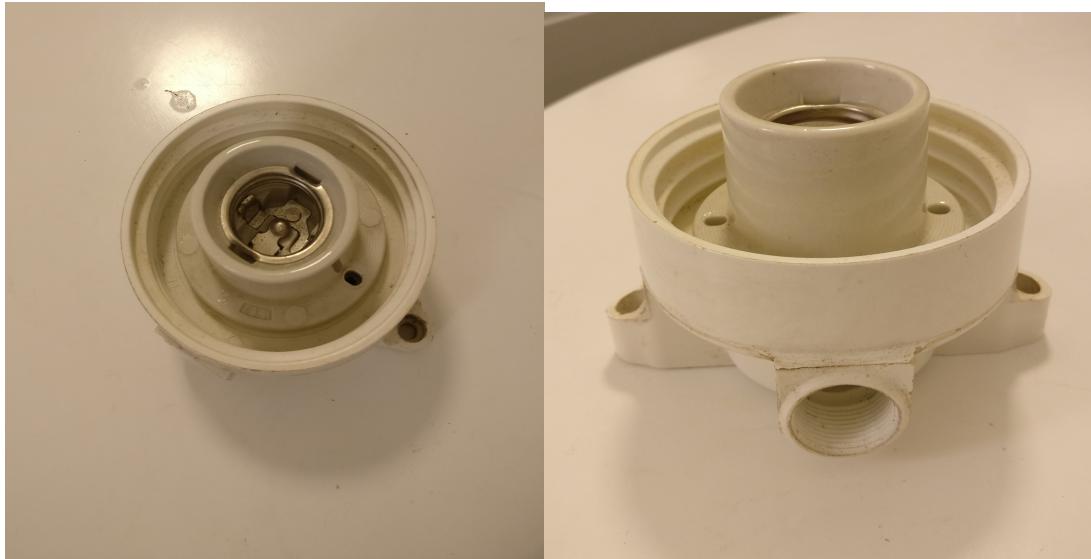


Fig 4: An old model of a light holder to be changed into IoT system

As a part of the project, a printed circuit board is designed and fitted into the hollow space existing on the old design of the holder. The required component information for the design is provided and all the required components on design are to be created and customized personally on the software library. The project consists only of the hardware development and the software is not included as part of the project.

5.2 Design Framework

The design has to be carried out according to the given information for schematic and most importantly the dimension of the given old light system holder, so that the designed board fits into the system physically as well as functionally. The figure 5 shows the light holder and the hollow semicircular space where the designed PCB board has to be fitted and fixed to make it functional.



Fig 5: Picture showing the space on the old light system to fit the PCB

The design framework mainly consists of two parts namely schematic design and layout design.

5.2.1 Schematic Design

The window for designing schematic in KiCad is termed as Eeschema. Eeschema allows creating the graphics-based design required for the project. In this project, the circuit contained limited components such as photon microcontroller, resistors, amplifier and sensors. Although the tool provides wide range of component library the components such as photonh, amplifier and connectors were custom created using 'create delete, edit symbol', according to the requirements for the project. Schematic design is the graphical representation of the circuit where all the components are connected in order to create the actual design on the layout. Before discussing creating new components, creating a new library in order to save the custom components, is explained below.

5.2.1.1 Creating a New Library

Before creating any components in a schematic, a new library has to be created in order to save the custom created components. Although the newly created components can also be saved in an existing library, it is much more convenient to access the newly created components from the user's own library, which saves time and lowers the chances of getting errors in a circuit. Below is the description, how a new library was created in this project. Figure 6 shows the main window of KiCad software from where all the functions such as eeschema, PCBnew, create new component, and so on can be accessed.

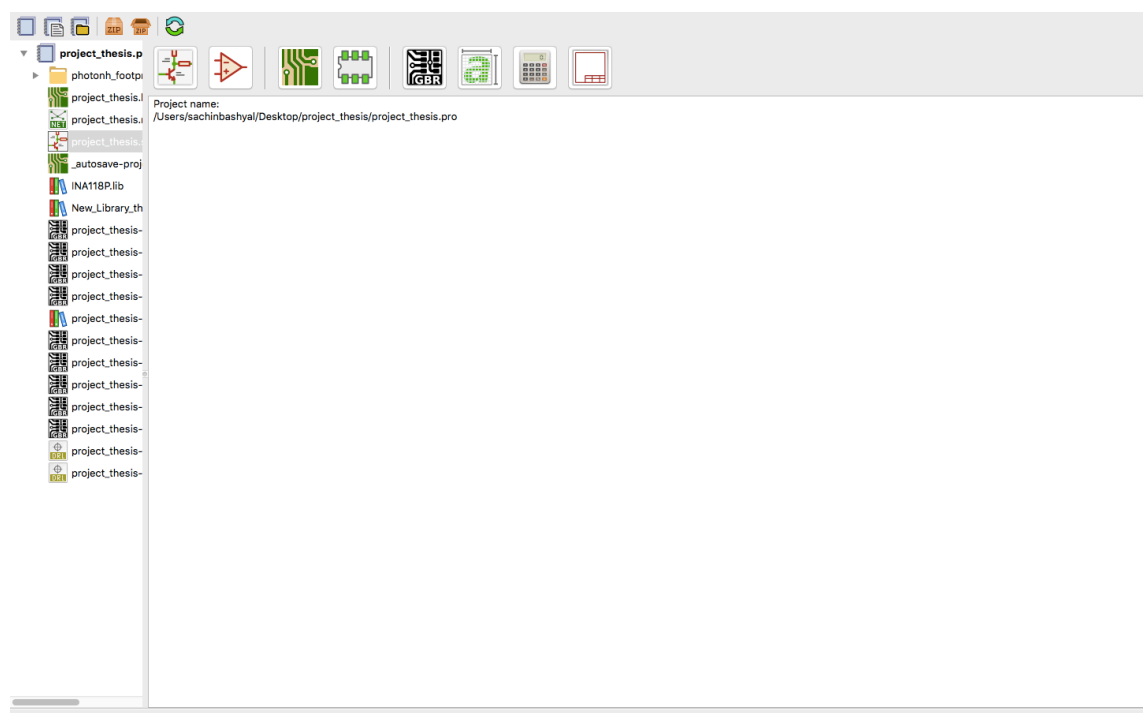


Fig 6: Main project window

Selecting the 'symbol library editor' option from above window, the below window will appear. Figure 7 shown below is the library window to create new components in eeschema.

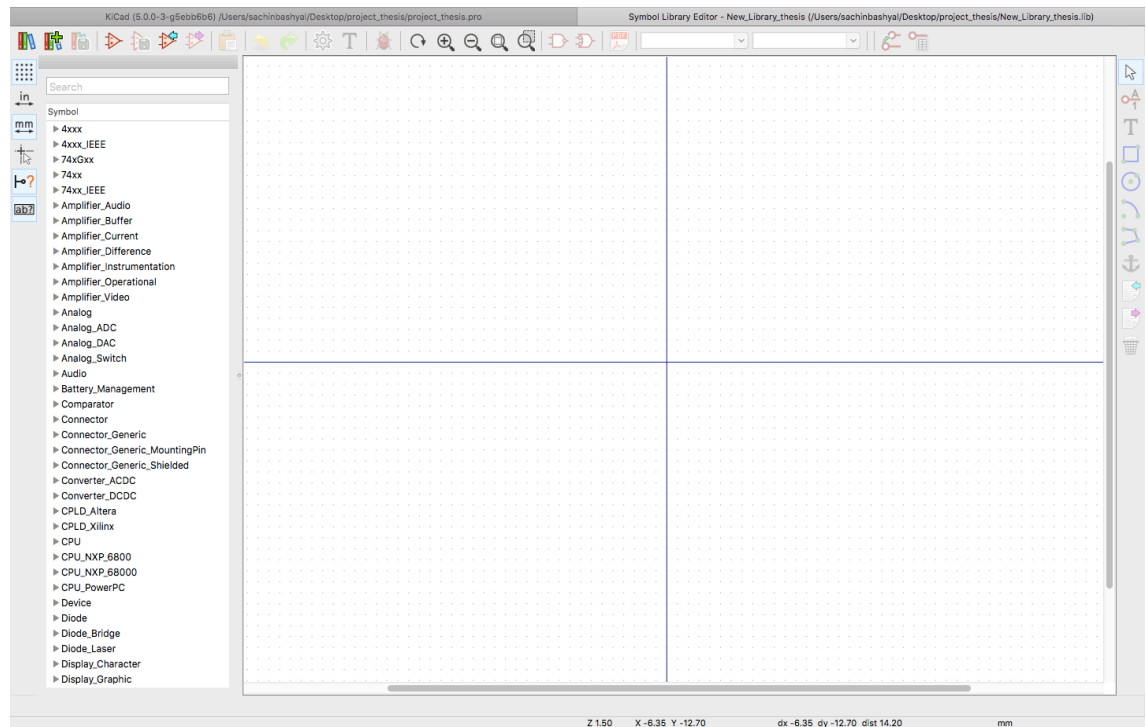


Fig 7: Library Editor Window

The green button on the top left corner of the window called 'create a new library' is selected, which shows the following dialog box. Figure 8 below shows the library naming dialog box.

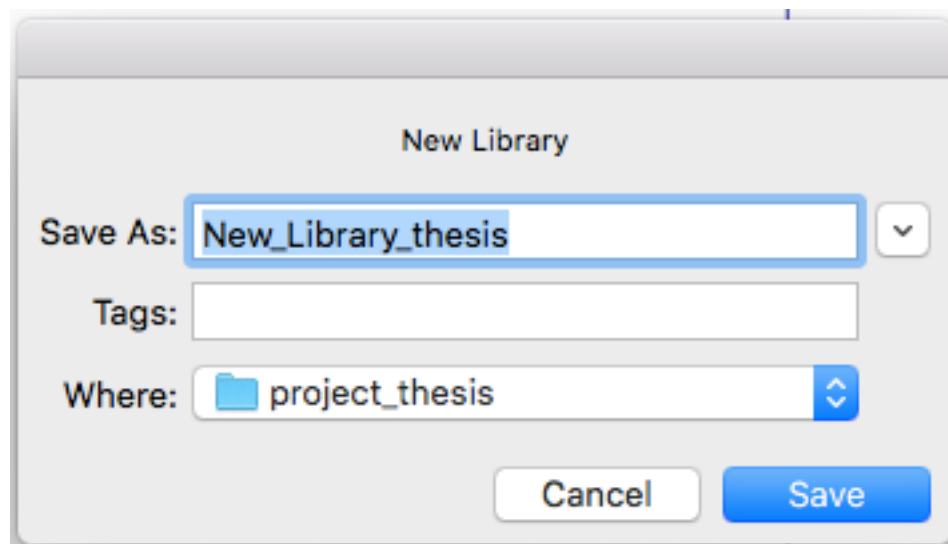


Fig 8: Library naming Dialog Box

The new library name is given as 'New_library_thesis' on 'save as' option and the preferred location of the library can be selected from 'where' option as project thesis, which is the folder created and saved on a desktop for this project. The new library name saved and create a new symbol option is selected which displays the following dialog box. Figure 9 below shows the library selection window.

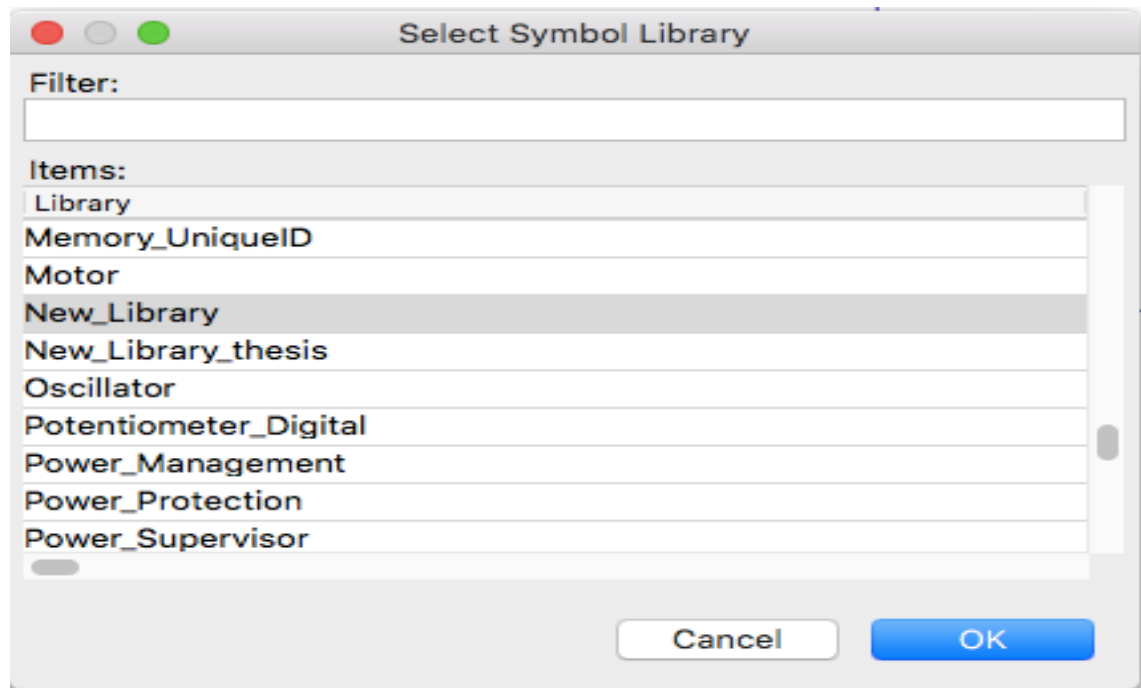


Fig 9: Library Symbol Select Window

From the 'filter' option the name of the library is selected as 'New_library_thesis' where the new component will be saved and 'OK', and the following symbol properties dialog box will appear from where we input symbol properties such as symbol name. Figure 10 below shows window, which allows to input and edit symbol properties.

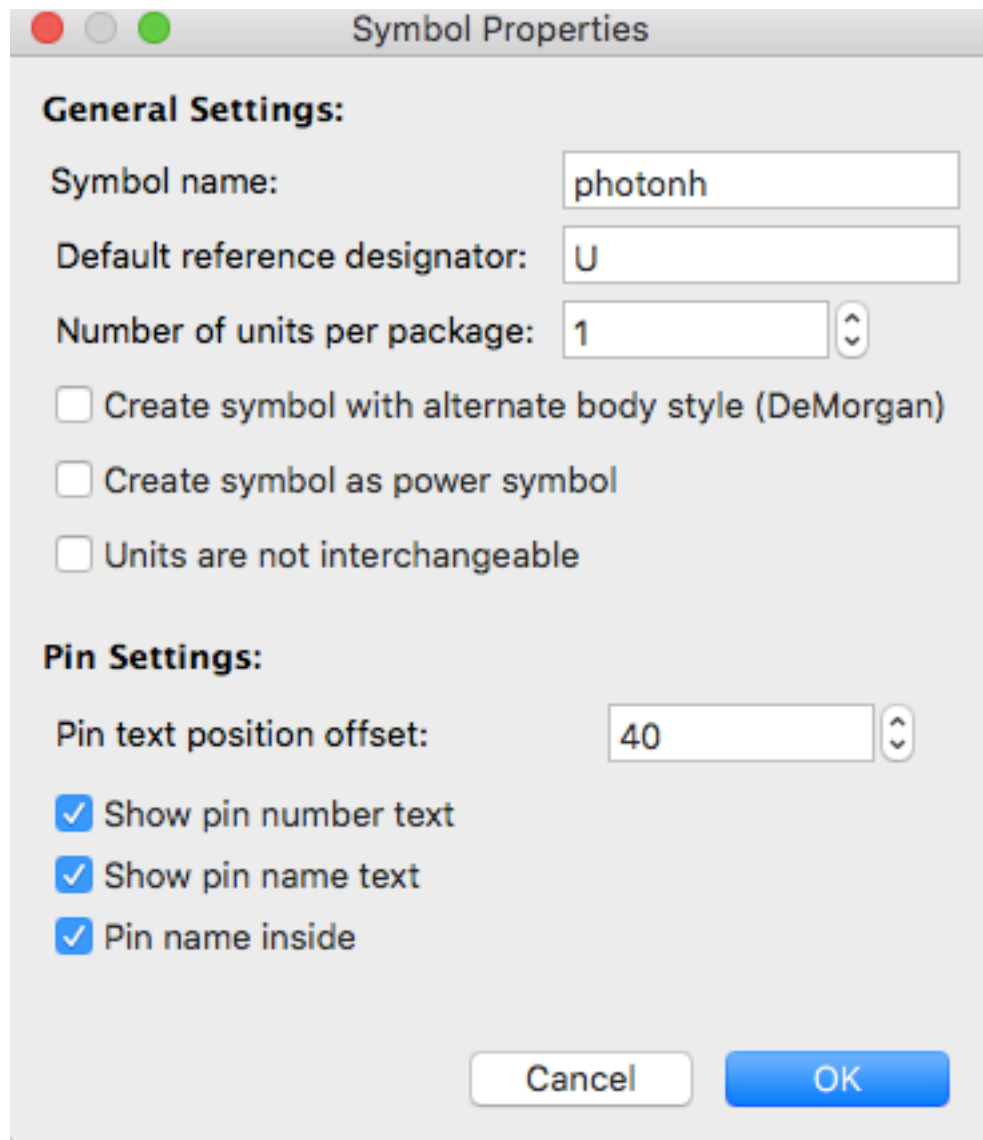


Fig 10: Symbol Properties Window

After giving the component name as 'photonh', 'OK' button is selected and the following window will appear where new component is created. Figure 11 below shows symbol library editor window, which allows to create a new part.

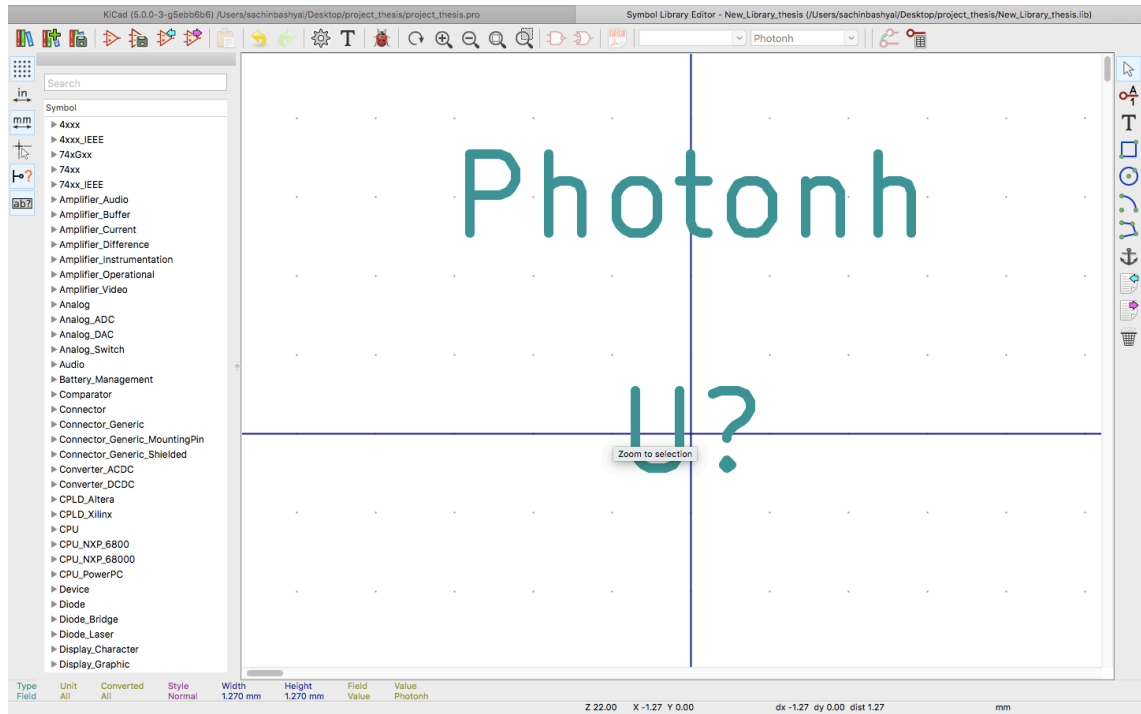


Fig 11: Symbol Library Editor Window

5.2.1.2 Creating Photonh

The above toolbar on figure 11 will allow a user to draw the part as required if its rectangular or circular components and allowing more options such as adding pins, texts, lines, arcs, and texts, which are located on the right-hand side of the window.

For the required component photon 24 pins are to be added, which can be selected from the right-hand side of the toolbar and the pin properties can be defined. Pin properties can be edit and input from the window shown below in figure 12.

Pin Properties

Pin name: Vin

Pin number: 1

Orientation: Right

Electrical type: Input

Graphic Style: Line

Name text size: 0.279 millimeters

Number text size: 0.279 millimeters

Length: 2.540 millimeters

Pin Pos X: -7.620 millimeters

Pin Pos Y: 10.160 millimeters

Sharing:

Common to all units in symbol

Common to all body styles (DeMorgan)

Schematic Properties:

Visible

Cancel OK

Fig 12: Pin Properties Window

The window shows the pin properties that can be edited as a number, pin name, electrical type, graphical style and the dimensions of text, number, unit and alternate representation membership and pin position. The first pin is named as Vin and pin number is given 1. Similarly, the remaining 23 pins are added and stated the properties according to the given datasheet. The graphical style for this component is selected as 'line'. Also, it is possible to edit the length and size of text and number parameters by clicking on the text and number of pins. Along with the pin numbering and naming, pin graphical styles can also be selected from the window as shown below in figure 13.

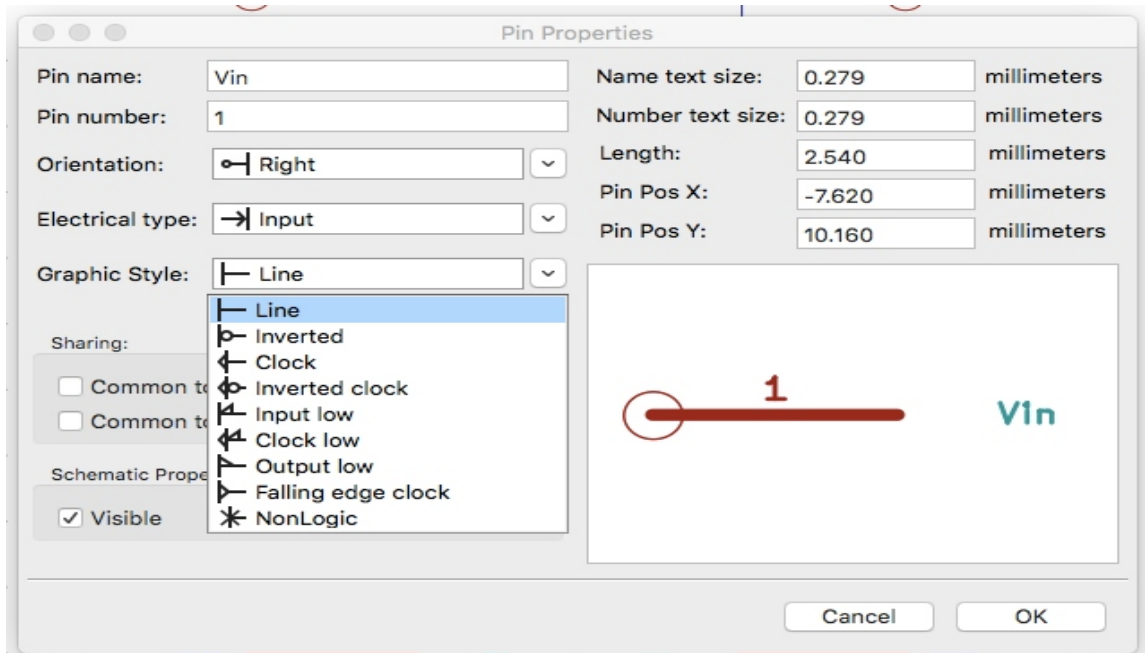


Fig 13: Pin Properties with graphical style Window

After adding 24 pins, naming the pins, numbering them and selecting pin graphical styles, the graphic rectangle is added to the symbol body, which will give the following new symbol photonh as shown in figure 14.

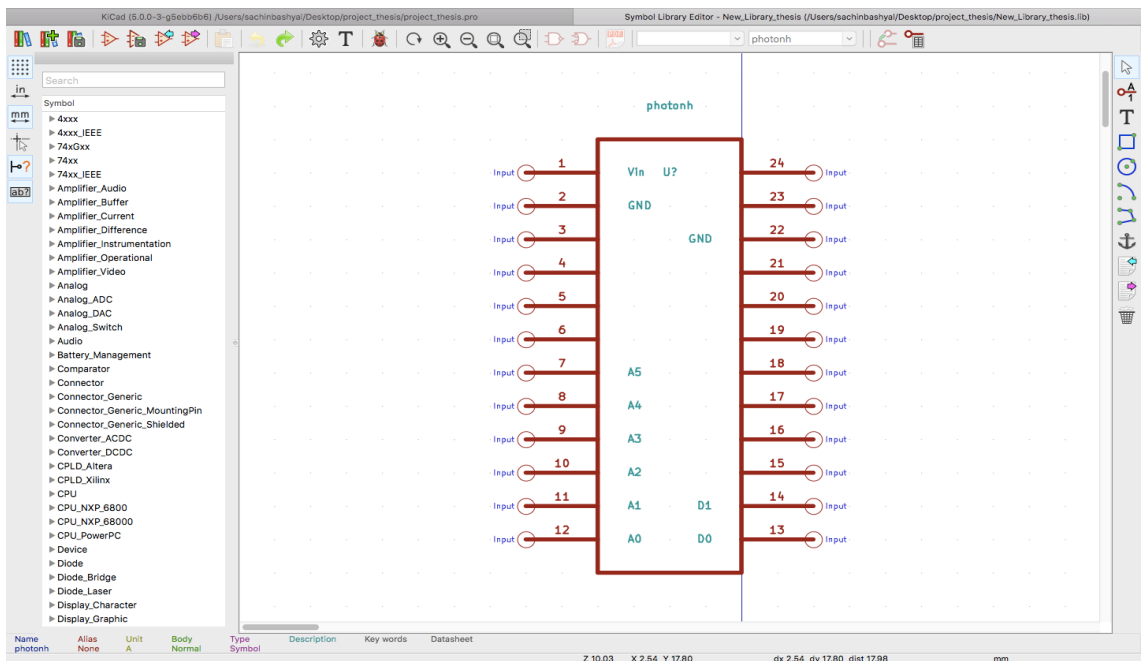


Fig 14: Newly created component photonh

Finally, the component is created and before saving the component, check duplicate and off-grid pins can be performed from the button on the top which looks like a red bug so that there will be no error on the component. The component is saved using 'save current symbol' and it will be available in the eeschema during creating schematic the diagram.

The required amplifier INA118P was created in a similar process as described above. After the required components are placed in eeschema, the following schematic was created. Figure 15 shows the complete schematic diagram of the circuit needed to create for the project.

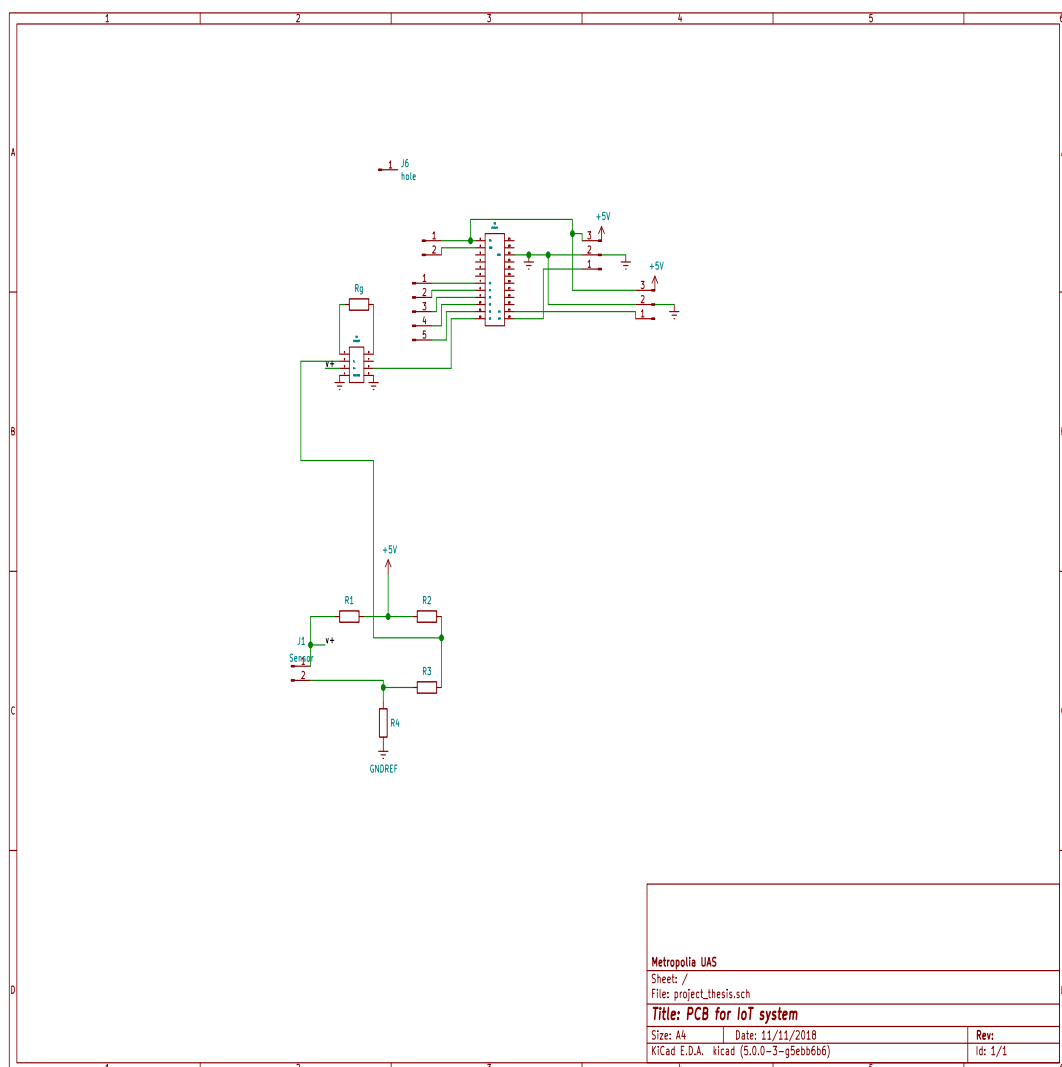


Fig 15: A complete diagram in Eeschema

5.2.2 Layout Design

The actual board is designed on a layout window, which is called PCBnew in KiCad. Before performing any design in PCBnew it is necessary to perform annotation, generate a netlist and matching the footprints and components from the schematic in eeschema. After reading the netlist from PCBnew window the actual circuit design is fetched on PCB new window, where we place the components in such order that it gives the best professional design using minimum vias and creating the route tracks. Even before matching the components from schematic to the PCBnew, the corresponding footprints have to exist on footprint library otherwise it needed to be created. In this project, every footprint components were custom created. Below is the description for creating footprints.

5.2.2.1 Creating a New Footprint and Footprint Library

If a new footprint is created it is always safe to create a new library where the footprints can be stored and saved in order to fetch easily while matching the parts from schematic with footprints. Newly created footprints can also be saved in existing libraries. It might not be so easy to find the components if the library has a lot of component libraries so it is always safe and easy to create own customized library in order to access new footprints easily.

New footprint can be created through the footprint library editor. Regarding this project creation of Photonh footprint is discussed. Selecting the 'footprint library editor' and 'new footprint' option the following dialog box to name new footprint will appear as figure 16.

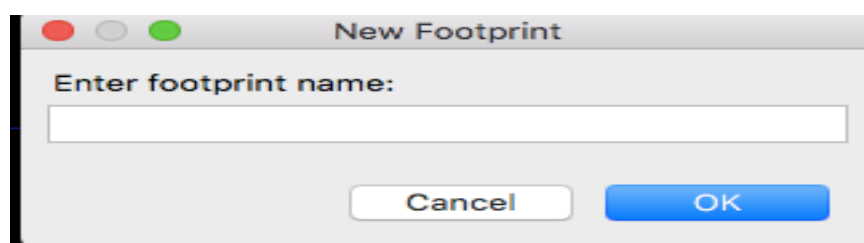


Fig 16: Footprint naming window

Selecting the footprint name as photonh and pressing 'OK', the following window will appear. The following figure 17 shows the window to create a new footprint.

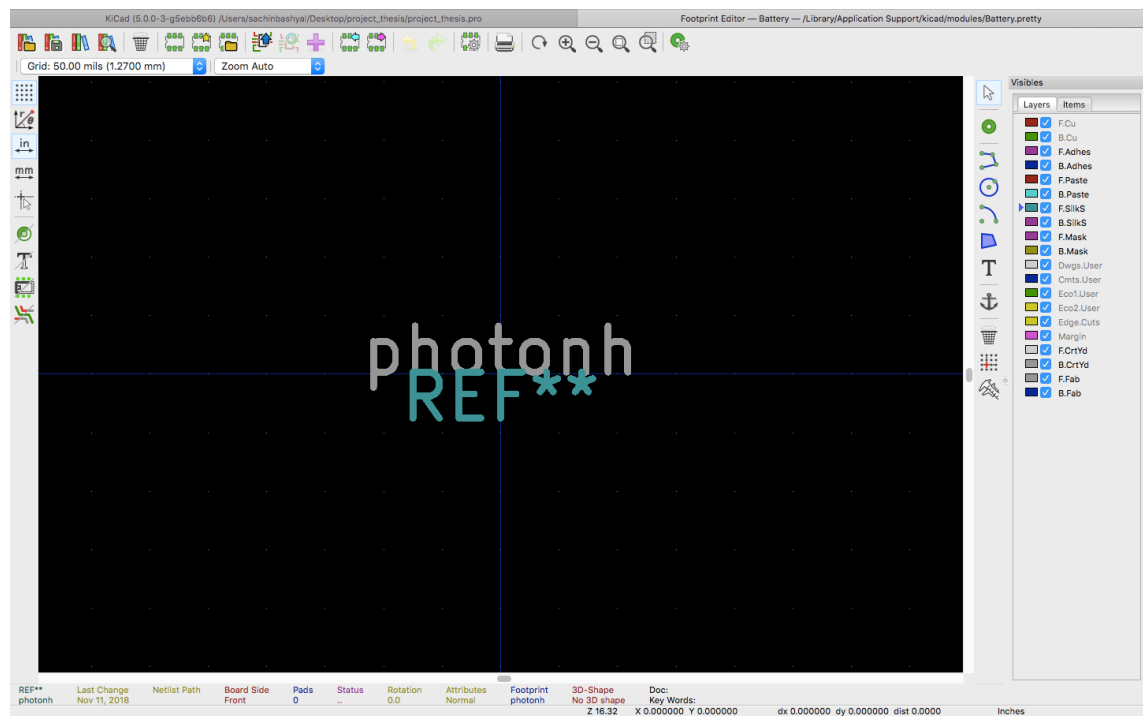


Fig 17: Footprint library editor window

The buttons on the right hand side of window provides options such as adding pad, graphics line, graphics circle, graphics arc, graphics polygon, texts, anchors and tool for measurement whereas left sidebar provides menu such as show or hide grid, polar coordinates, units either millimeters or inches, cursor shapes and so on.

5.2.2.2 Editing Pad Properties

One of the most important parts for designing footprint is selecting pad properties such as shape and size of the pad, size of a hole in case of selecting through hole along with the dimension of a component. Pad type can be selected either through hole, SMD or connector but in addition, it also gives the option to select NPTH, mechanical. Pad shape has various options such as circular, rectangular, oval, trapezoidal or rounded rectangle. Other options such as hole shape, hole size in x and y coordinates, the position of a pad in x and y-axis, pad numbering, a net name can also be customized. Selection of pad-type depends on the component itself. For instance, a through hole pad

is used to design a photonh. Figure 18 shows a window with a single through hole pad to create a required component for this project, which is photonh. [4.]

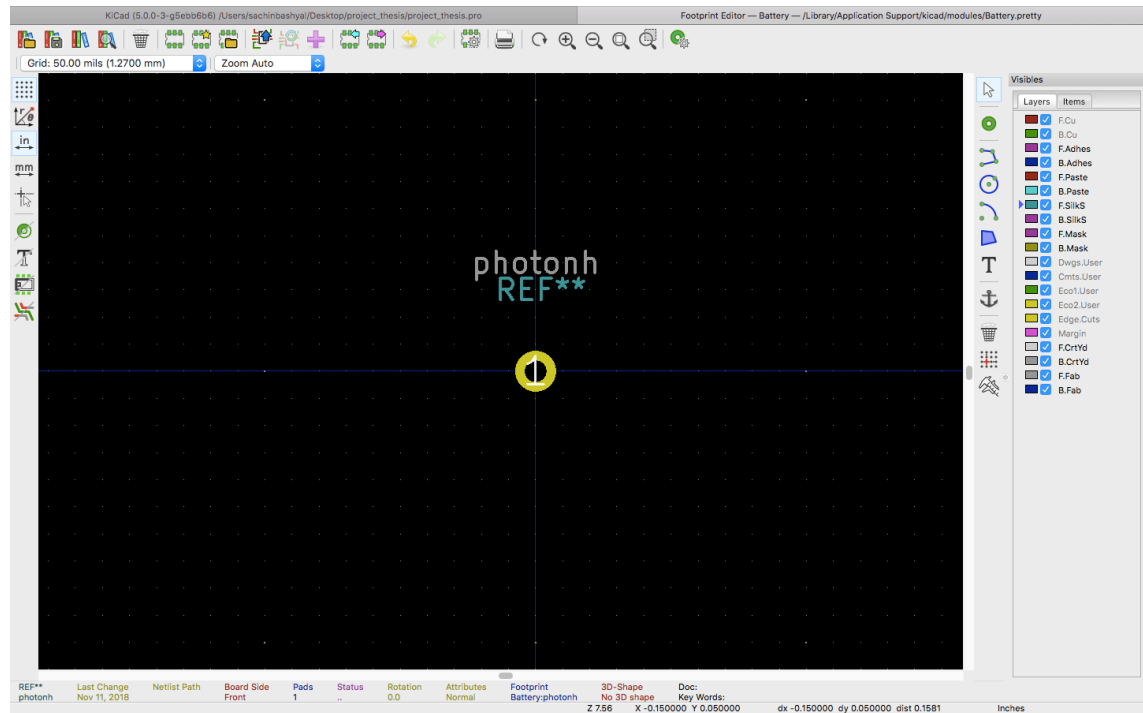


Fig 18: Footprint editor window with a single pad

To create a photonh first a through hole pad is selected. And the pad properties are edited by either double-click or right-click and selecting the properties. The following pad properties window will appear where the properties for the pad are set as pad type-through hole, shape-circular, hole shape-circular and dimensions as size x 0.074 and hole size 0.04. The dimensions are given on the datasheet and the measurement units are on inches. Figure 19 shows the window to edit and input pad properties.

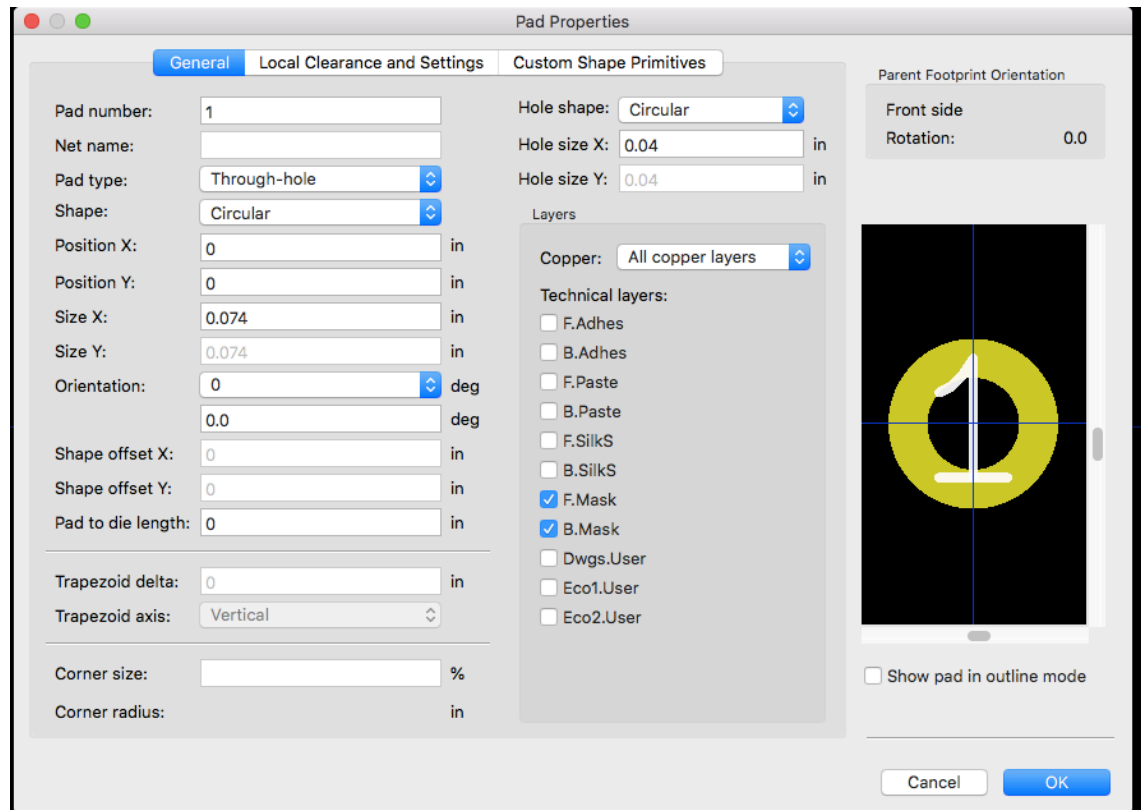


Fig 19: Pad properties window

5.2.2.3 Creating Array

Once a pad is created with the required category and dimensions, multiple pads of similar properties can be created easily using create array option since most of the components required multiple pads to complete a design. Creating array for multiple pads saves time and eases the design process having less chance of making an error. As the dimensions and properties for the pad are already set, it creates same pads during creating the array but the most important part of creating an array is deciding space between two pads which can be found in the datasheet of the required component. Apart from that, create array window also provides multiple options such as array type: grid or circular, vertical and horizontal count, vertical and horizontal spacing, vertical and horizontal offset, pad numbering direction and start position, pad numbering scheme and so on. Figure 20 shows the window to create array.

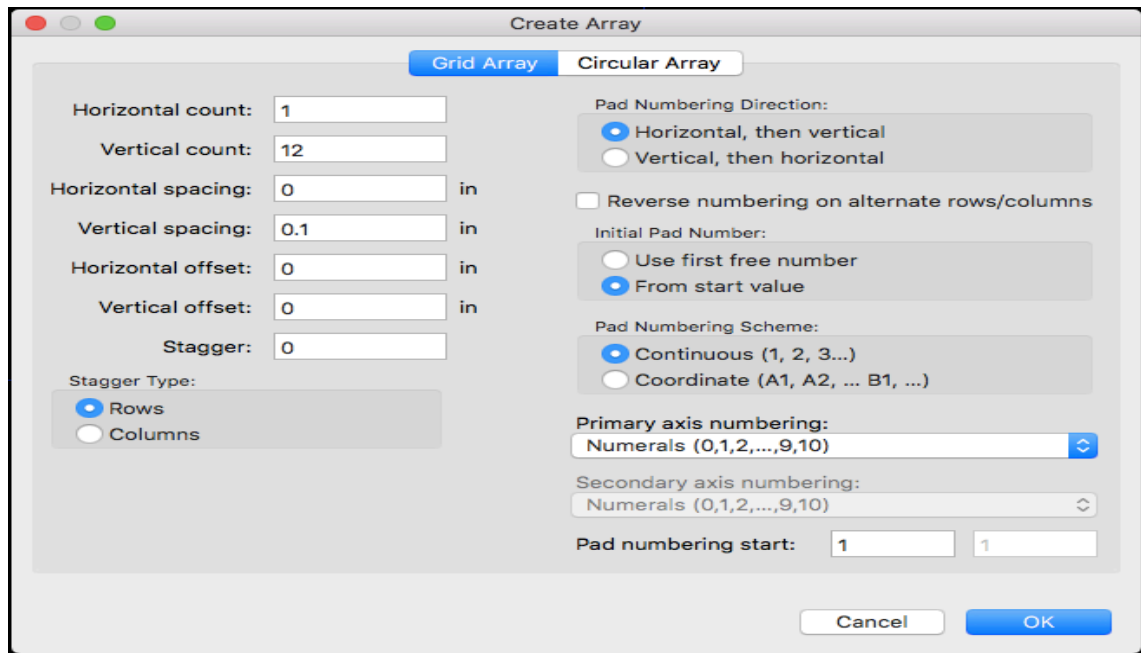


Fig 20: Create array window

Considering the following design, the component photonh is 24 pinned so we have to create the array first for 12 pins downward vertical and 2 pins horizontal and 12 pins up-ward vertical. So the pin count for creating the first array is vertical pin 12 and horizontal 1, vertical spacing is 0.10 inch (from datasheet), horizontal spacing is 0, and 'pin numbering start' is 1. Figure 21 shows the window after creating 12 pads using 'create array' function.



Fig 21: Creating 12 vertical pads using create array function

To create the horizontal pads from 12th pin, the values change from horizontal count to 2, vertical count 1, pad numbering starts from 12 and the horizontal spacing 0.7 inches (from datasheet), vertical spacing is 0. Figure 22 shows creating 13th pad using array.

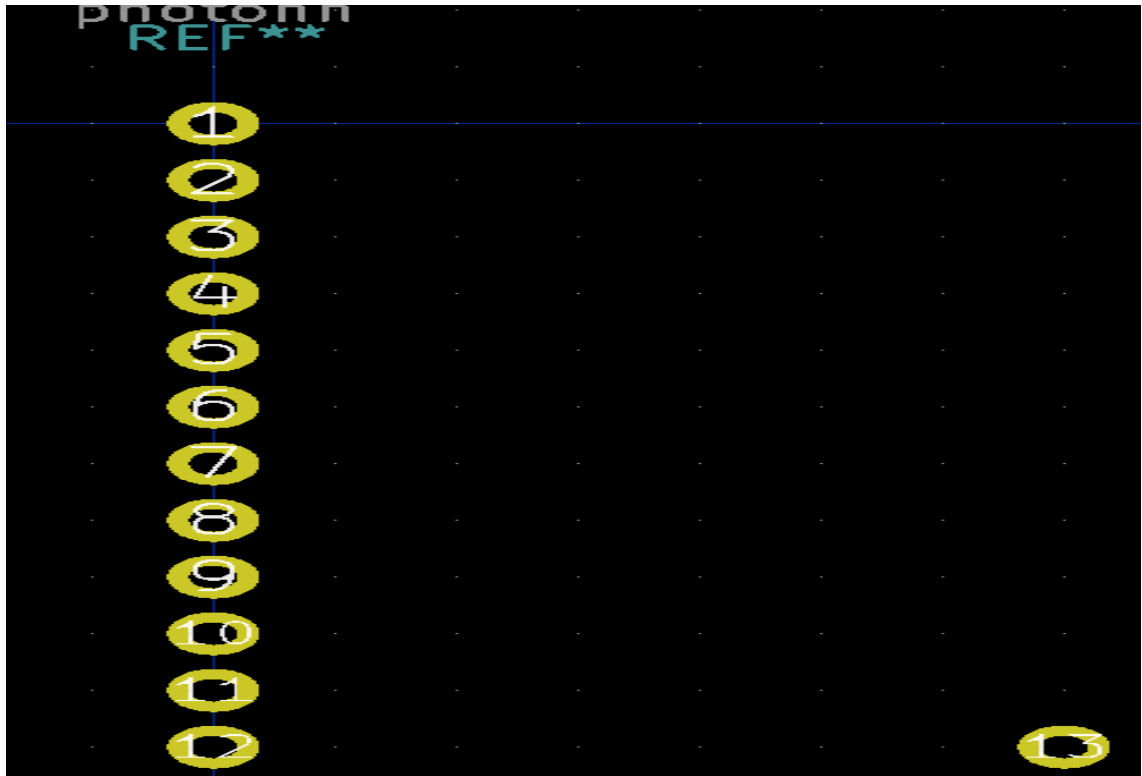


Fig 22: Creating 13th pad using create array function

Finally, rest of pads are placed by creating an array from 13th pad to upward vertical direction where the value changes as vertical count as -0.10 since the pad position moves to upward which is opposite direction in the y-axis and all other dimensions remain same except pin count from position 13. After adding the graphic lines which are 0.8 inches horizontal and vertical length of 1.44 inch as given in datasheet, the component will appear as the following picture. Figure 23 shows a complete component or a footprint.

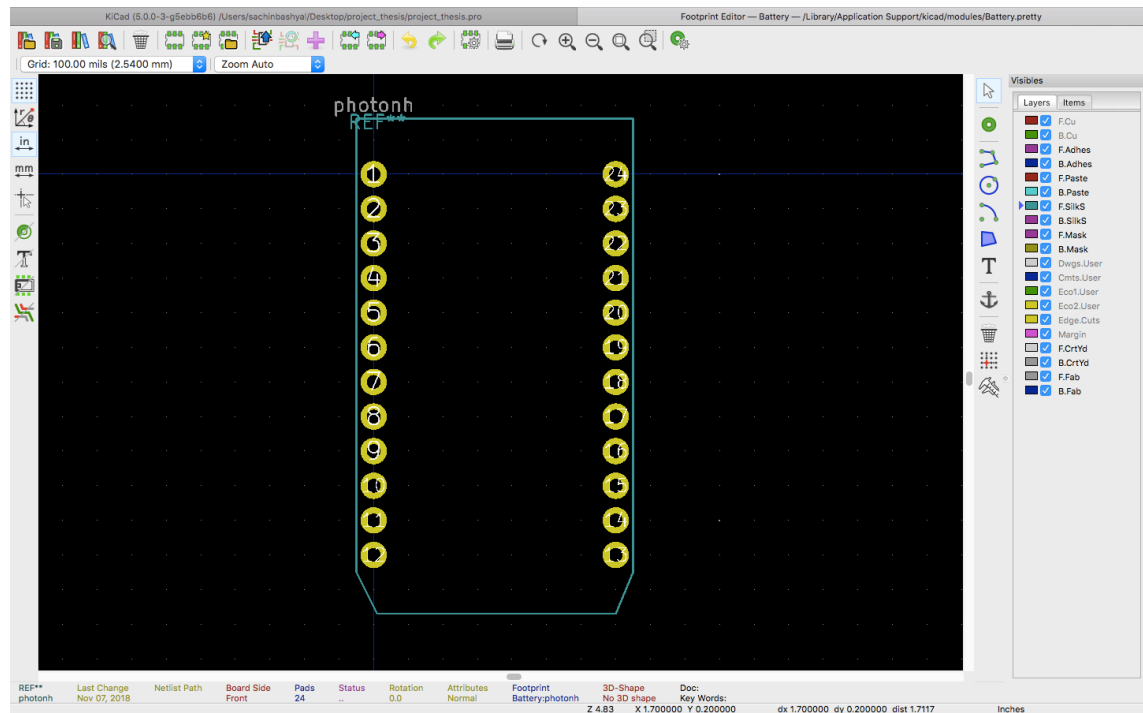


Fig 23: Final footprint layout of component photonh

Lastly, the component has to be saved in new footprint library which can be created by selecting 'create a new library and save current footprint' option or in the existing library by selecting 'save footprint in active library' option so that the component is available during parts matching.

5.3 Creating Amplifier (INA118P) Footprint:

Creating INA118P amplifier footprint has the similar procedure as the photonh however it does not contain the through hole pads instead it has THT (through hole technology) pads. Selecting the 'new footprint' option and giving a name for the new footprint, we select the new pad as through hole as shown below. Figure 24 shows a window to create an amplifier footprint.

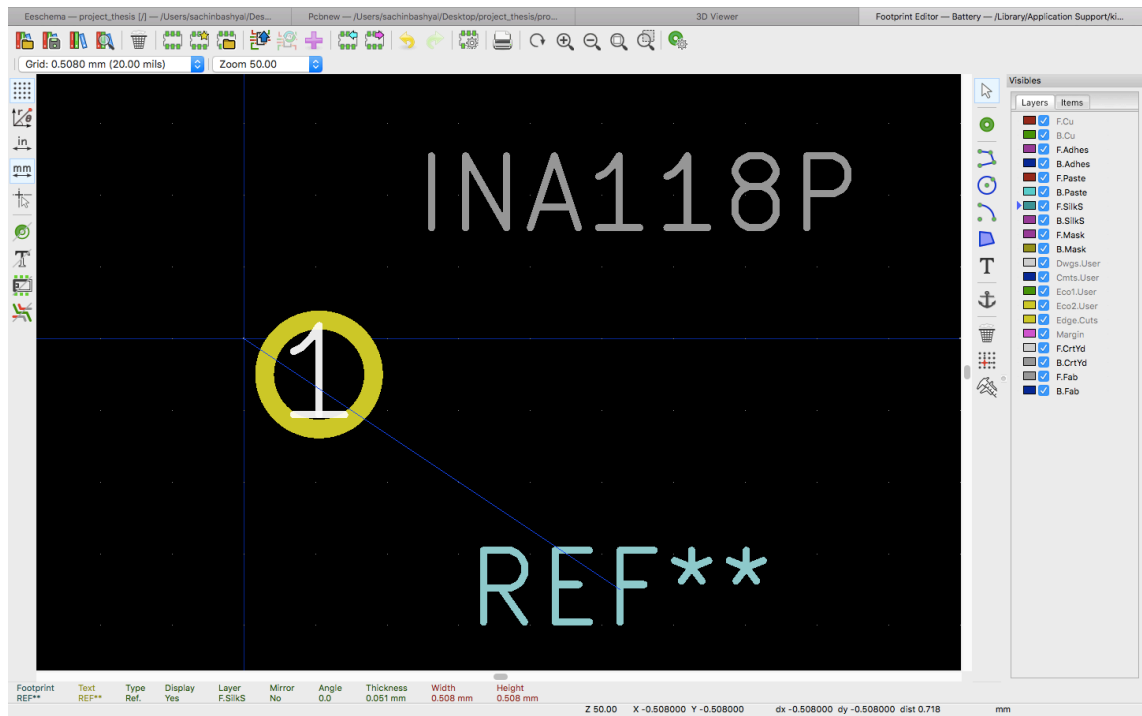


Fig 24: Footprint editor with through-hole pad

The pad properties are set as pad type through hole, shape rectangular and dimensions as size x 0.041 and size y 0.016 and units are inches.

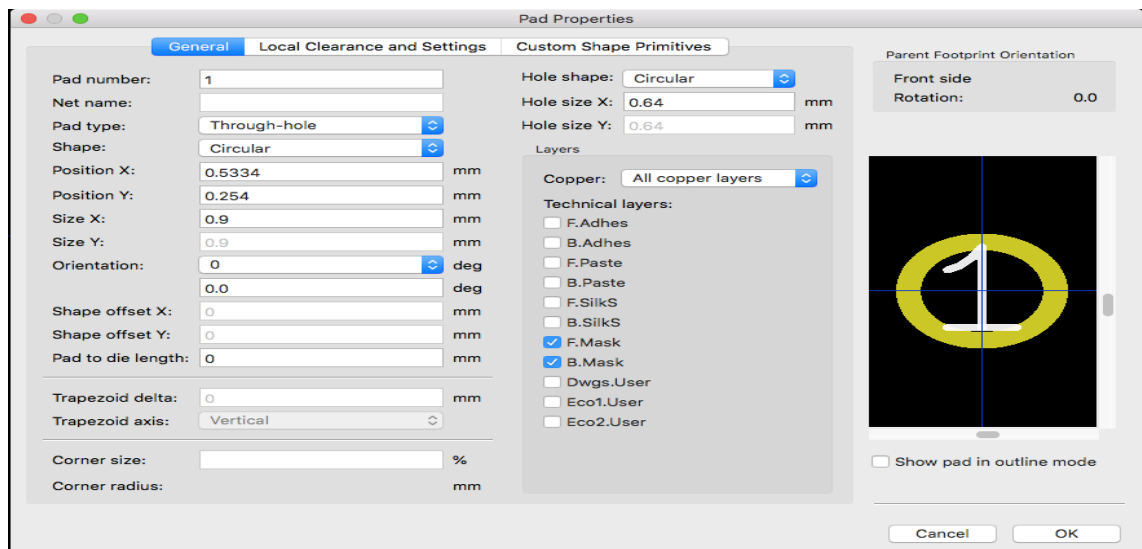


Fig 25: Pad properties window for through hole pad

Similarly using create array properties three similar pads are created vertical downwards by the difference of 0.05 inches between to pads. Also creating the array bet-

ween two pads horizontally, the distance between two pads is 0.198 inches and creating array vertically upwards for further 4 more pads as the same dimensions of the difference of 0.05 inches between two pads. Figure 26 shows eight pads, created using array function, required for INA118P component as a footprint.

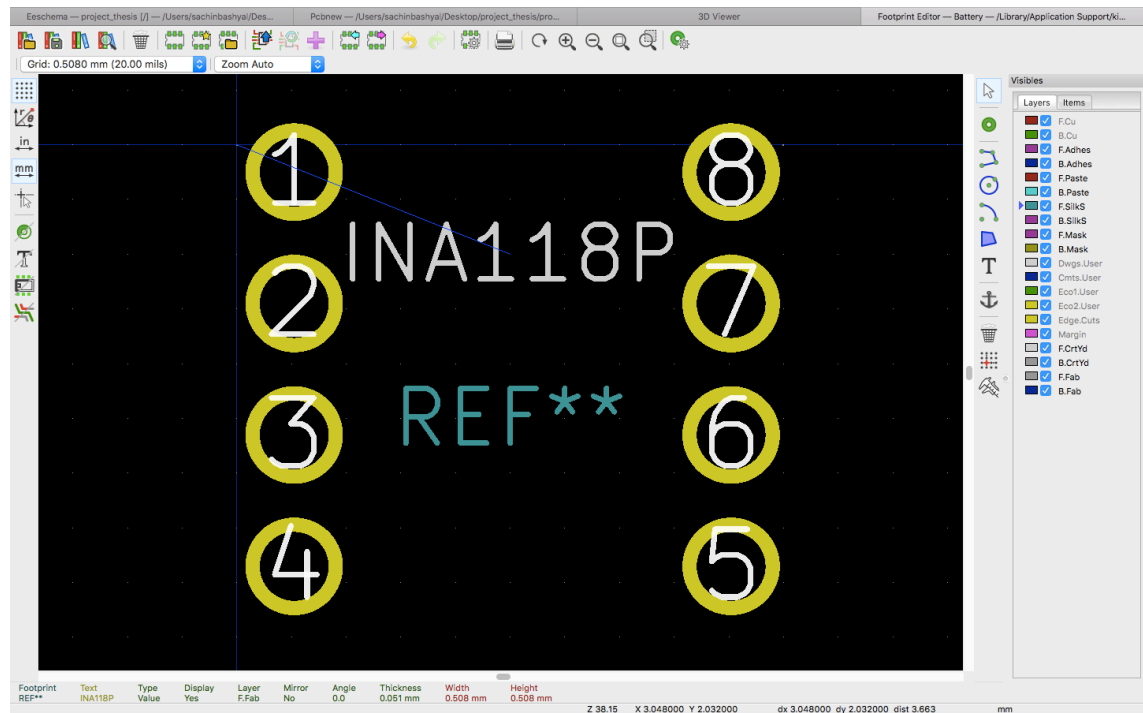


Fig 26: Through hole pads created by using array functions

Finally, adding the graphic lines between these pads as the vertical length 0.197 inches and horizontal length 0.157 inches the following footprint symbol is ready as an amplifier INA118P. Figure 27 shows a complete footprint of required component INA118P amplifier.

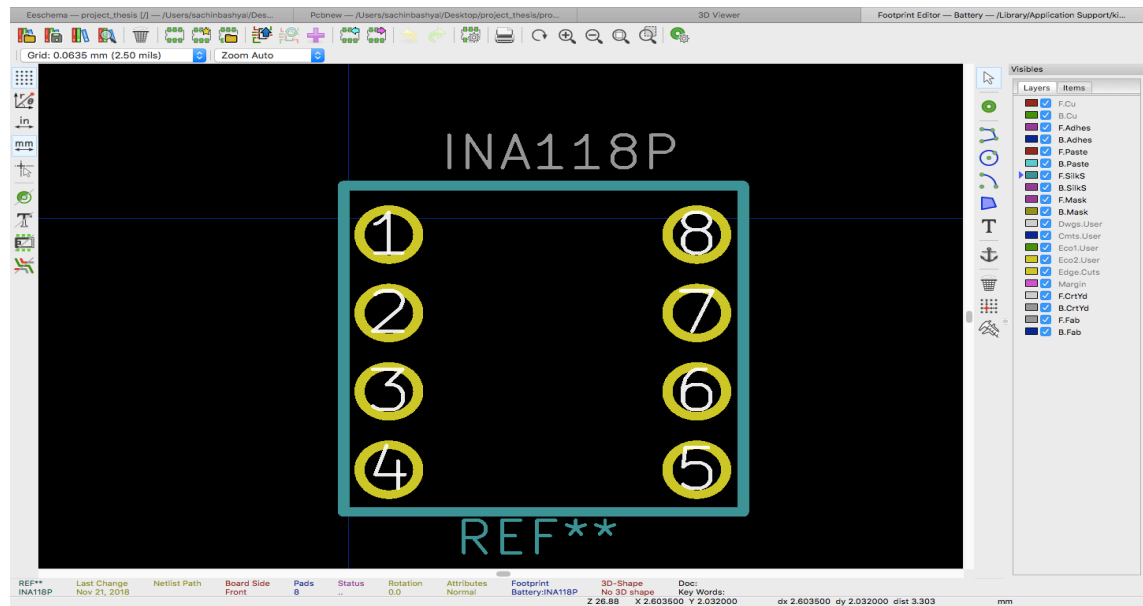


Fig 27: Newly created amplifier footprint

5.4 Creating Connector and Resistor Footprints

5.4.1 Creating Connector Footprints

Creating a connector footprint is pretty simple. The only dimension we need to consider is hole diameter and the distance between two pads. We place the through hole pads with hole size 0.64 mm and the distance between two pads is 2.54 mm. The types of connectors used for this project were 1*2, 1*3, 1*5. Removing 3 and 2 pads consecutively from a 1*5 pin connector and taking the same dimensions for hole size and pad distance will create 1*2 and 1*3 connectors. The following figure 28 shows footprint of 1*5 female connector type.

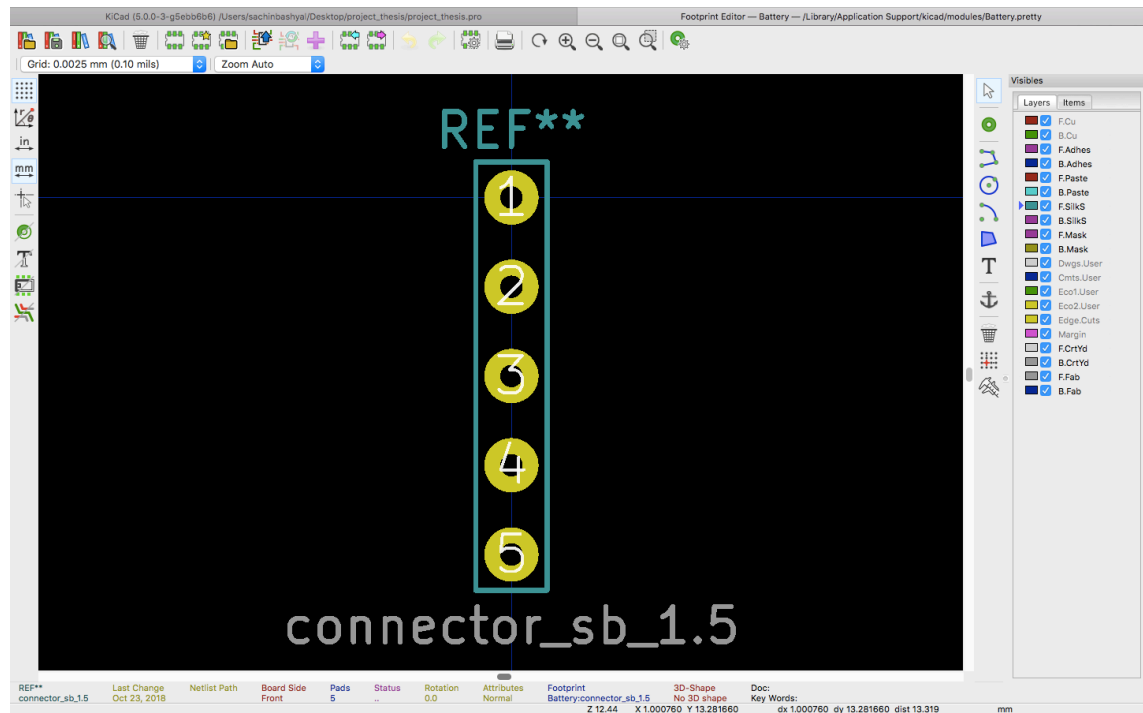


Fig 28: Five pin connector footprint

5.4.2 Creating Resistor Footprints

The resistor used for this project was THT type. Creating THT resistor is also quite simple. The resistor dimension used for distance between two through hole was 6.5 mm and the hole size was used was 1.22 mm. Total five resistors were used of same dimensions. The following figure 29 shows through-hole resistor footprint.

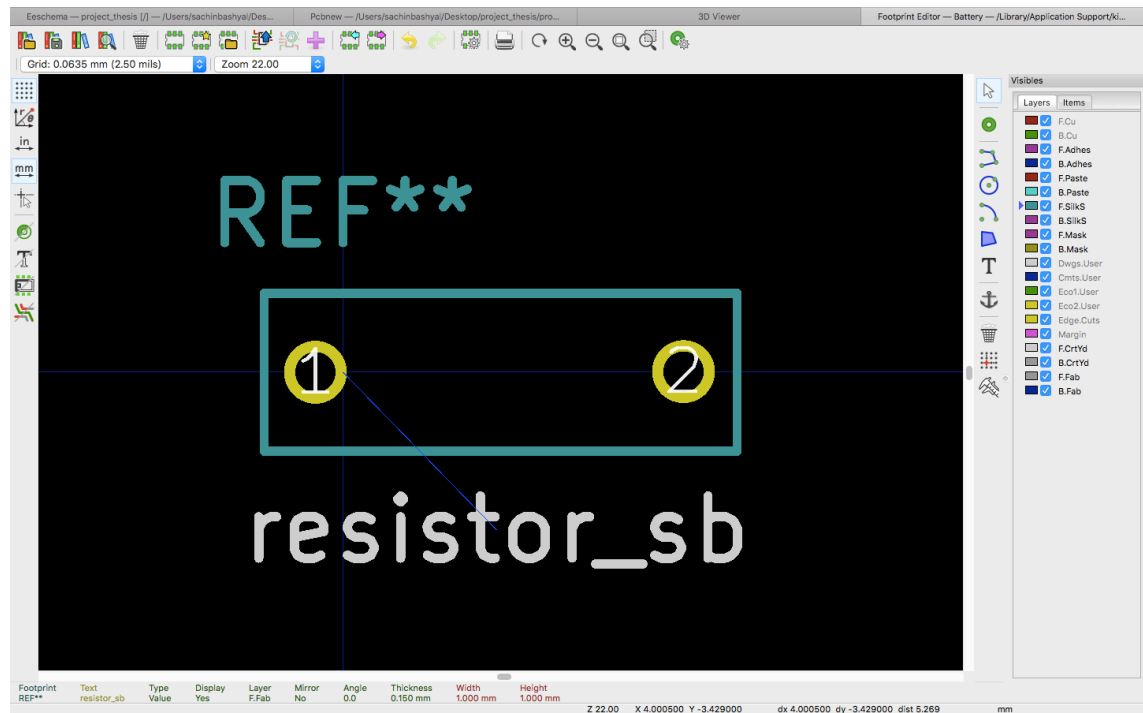


Fig 29: THT Resistor Footprint

Lastly, the layout consists of one mounted hole used to fix the board to the holder so that it can stay still. The mounting-hole dimensions were taken by measuring the hole in the holder where the board is supposed to be attached. The measured dimension for the mounting hole was 0.45 mm. Figure 30 shows the footprint for mounting-hole, which is used to mount the PCB board on light holder.

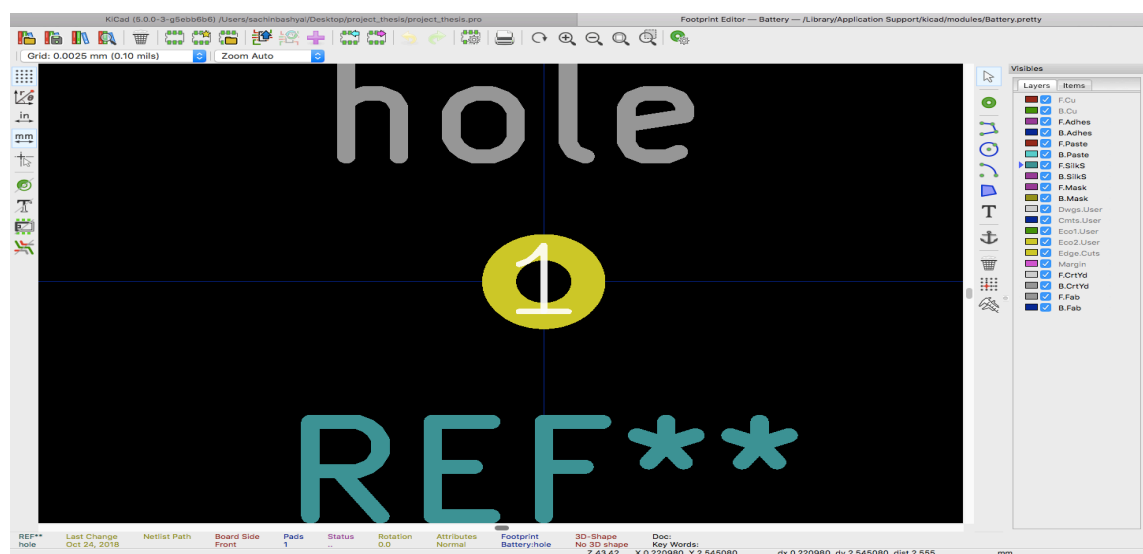


Fig 30: Mounting hole footprint

Matching all these newly created footprint components to the parts in a schematic diagram, creating the netlist, annotating and finally reading netlist from the PCBnew window, a diagram is appeared where components need to be placed, in order and routed so that the final design will look like below. Figure 31 shows the complete arranged and routed components for a PCB design required for this project.

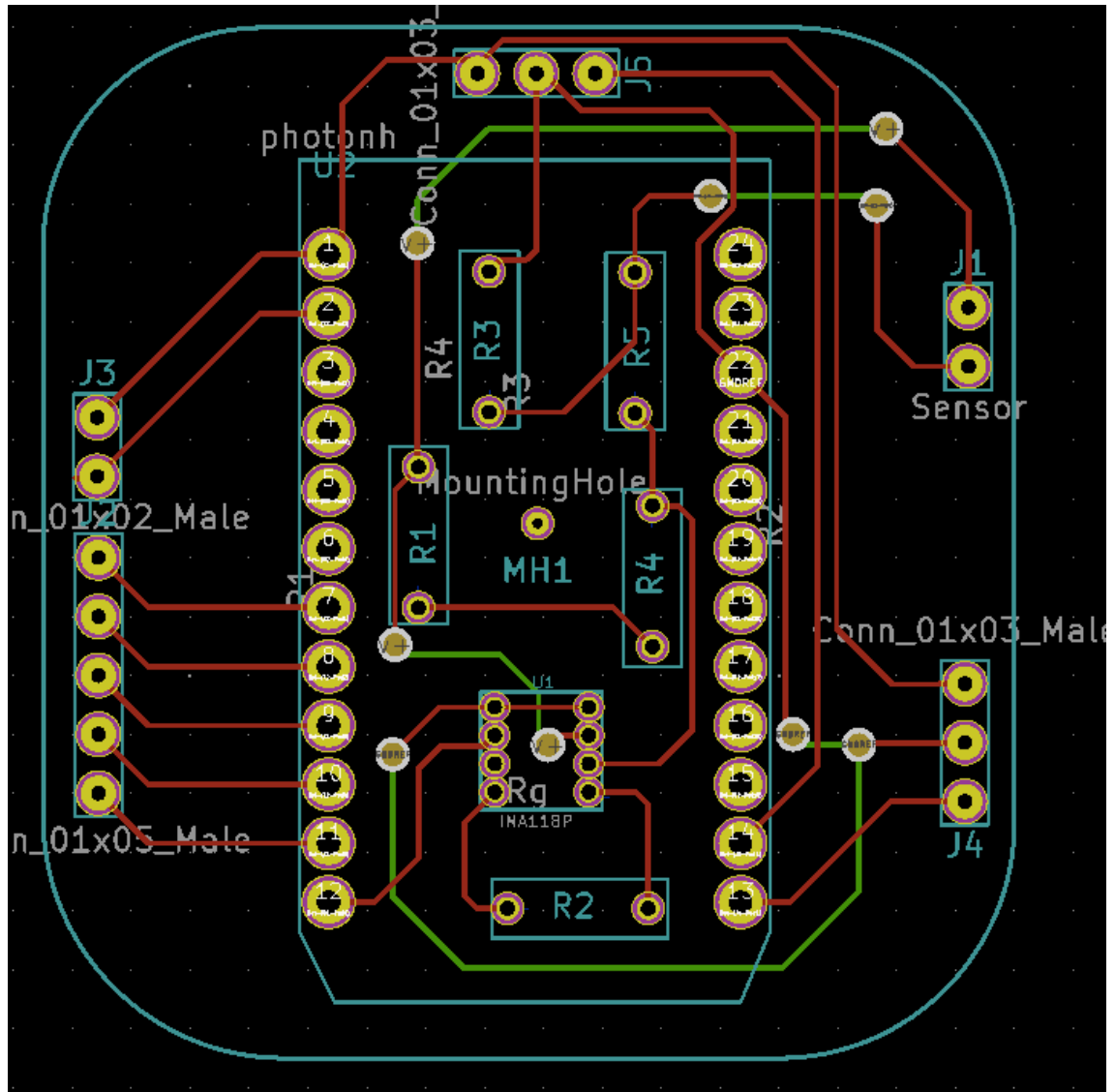


Fig 31: A complete design layout

The PCBnew window also allows the user to view the 3D version of the designed board. When the board will be printed using a milling machine it will look like the picture shown below. Figure 32 shows three dimensions PCB board which looks similar after the fabrication process of the created design.

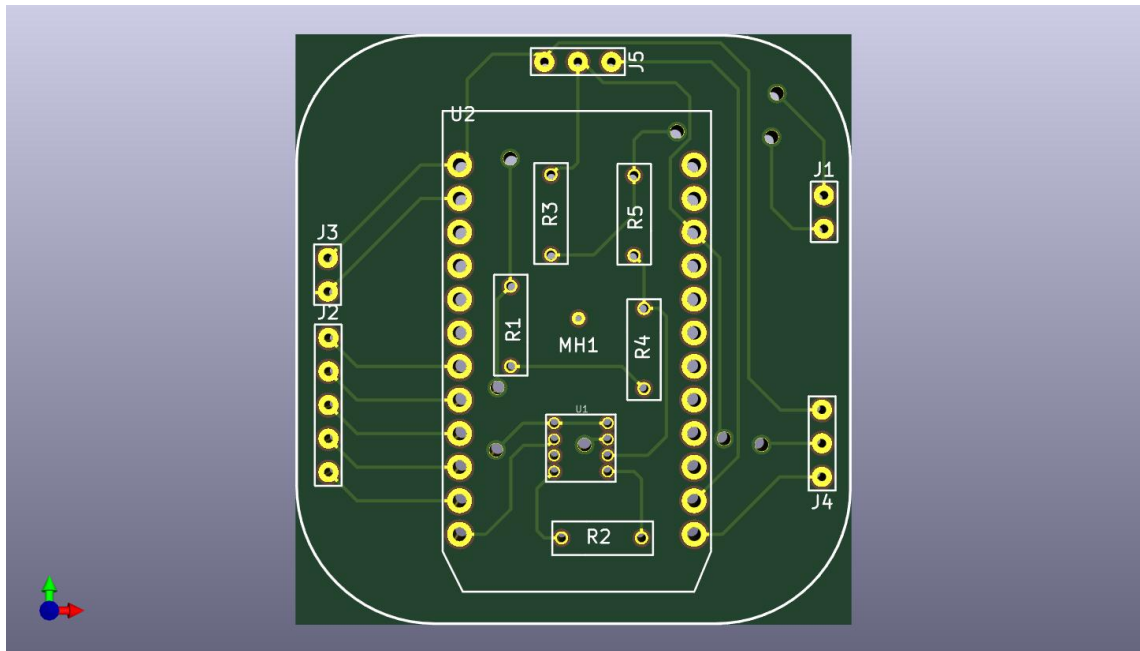


Fig 32: 3D version of design layout

6 Conclusion

Hence the project was carried out using free design tools and according to the requirements for the IoT system. The project is aimed to develop a part of a smart home project where the lighting systems will be functioning as an IoT system. Although KiCad was selected as a design tool various other software could also be used. According to the preference, KiCad was emphasized because of its open source property, user-friendly graphics and abundant free resources to learn. During the selection of design tools PADS Logic and PADS Layout were the best ones but due to the previous experience on those tools, KiCad was selected in order to get more knowledge and experience on the tool that has not been operated.

This project contributes a small part in a smart home project. Having designed an IoT system for the home light it brings ease and comfort in user operation and at the same time, it also contributes to modern technology. The designed board has to be printed and tested to be functioning before placing the components and feeding programme to be fully functioning. However, due to the delay of installed lab appliances in the new University building, the projected is limited only to designing circuit.

References

1. Rouse M. TechTarget- IoT Agenda [online]
URL: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> Accessed 10 October 2018.
2. Electrocomponents- Introduction to Design Spark [online]
URL: <https://www.electrocomponents.com/media/videolibrary/introduction-designspark> Accessed 11 October 2018.
3. Designspark guidebook pdf- DesignsparkPCB [online]
URL: <https://designspark.zendesk.com/hc/en-us/articles/115004904329-Is-there-a-book-about-DesignSpark-Mechanical->
Accessed 11 October 2018.
4. KiCad-PCBorg – Documentation [online]
URL: <http://kicad-pcb.org/help/documentation/> Accessed 14 October 2018.
5. Designspark wiki- [online]
URL: https://en.wikipedia.org/wiki/DesignSpark_PCB
Accessed 11 October 2018.
6. ExpressPCB -[online]
URL: https://www.expresspcb.com/wp-content/uploads/formidable/ExpressPCB_7_1_Quick_Start.pdf
Accessed 20 October 2018.
7. Marian P. Electroschematics- Best of free PCB design Softwares [online]
URL: <https://www.electroschematics.com/2249/pcb-design-software/>
Accessed 5 October 2018.
8. Sanfrancisco School - PCB School – The Ultimate PCB design software comparasion guide [online]
URL: <https://www.sfcircuits.com/pcb-school/pcb-design-software-comparison-guide#eagle-overview>

Accessed 19 November 2018

9. PCBShopper – expresspcbreviews - [online]

URL: <https://pcbshopper.com/expresspcb-reviews/> Accessed 19 November 2018.

