EasyEDA Tutorial

2022.03.02

v6.4.32

EasyEDA Editor：https://easyeda.com/editor

EasyEDA Desktop Client：https://easyeda.com/page/download

Remark

- This document will be updated as the new features of the editor are updated.
- The latest revison please refer at [EasyEDA Tutorials.pdf]()

FAQ

Editor FAQ

Please spend a few minutes reading this FAQ, it will save you lots of time getting started with EasyEDA.

Tutorial

Download for PDF

EasyEDA-Tutorials.pdf

Video Tutorials

Youtube - EasyEDA

Ask for Help

Contact Us

Update Records

Update Records

Schematic

If I update the schematic, how do I then update the PCB?

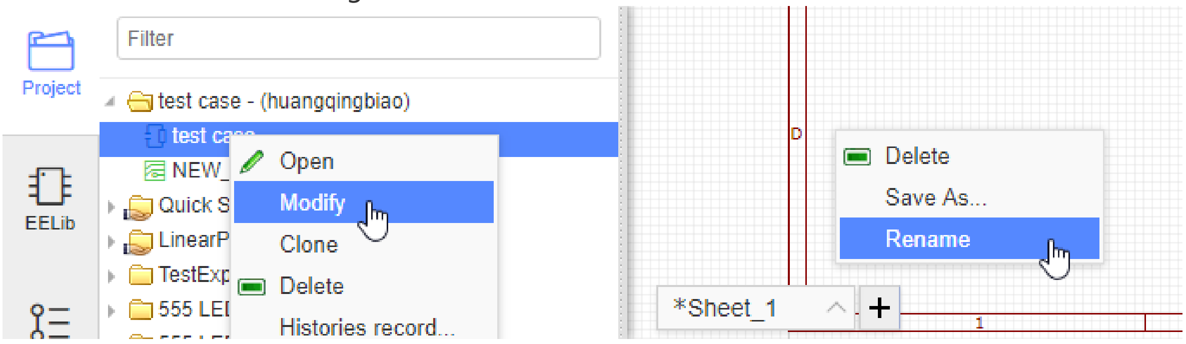
Using: "Menu - Design - Update PCB".

Alternatively, you can import changes from the schematic from within the PCB Editor:

https://docs.easyeda.com/en/PCB/Import-Changes/index.html

How to rename a Sheet/Page or modify description.

In this menu, there is a `Modify` option, so you can rename your files. Double click or right-click the sheet tab can change the sheet title too.



What is the unit of the schematic sheet? How to change schematic unit?

The basic unit of the schematic sheet is the pixel. 1 pixel is about 10mil (0.001 inch) but please note that this use of the pixels as a unit in a schematic is just for reference.

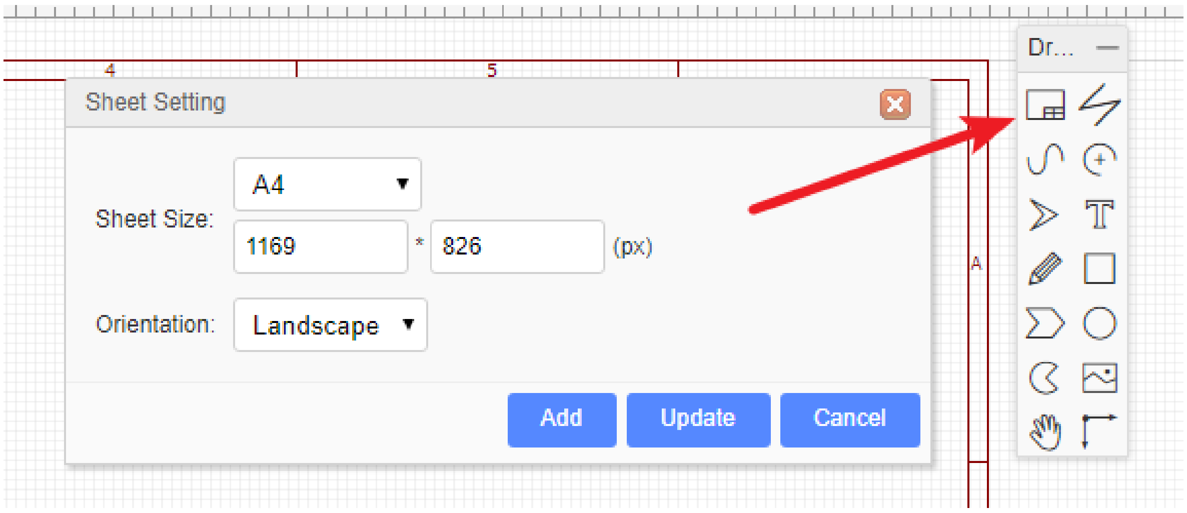
For a complex project, I want to split the schematic over several sheets. Does EasyEDA support hierarchy?

EasyEDA don't support hierarchy, but support multi-sheets。
Please check out this link https://docs.easyeda.com/en/Schematic/Multi-Sheet/index.html

How to change the sheet size and modify the design information.

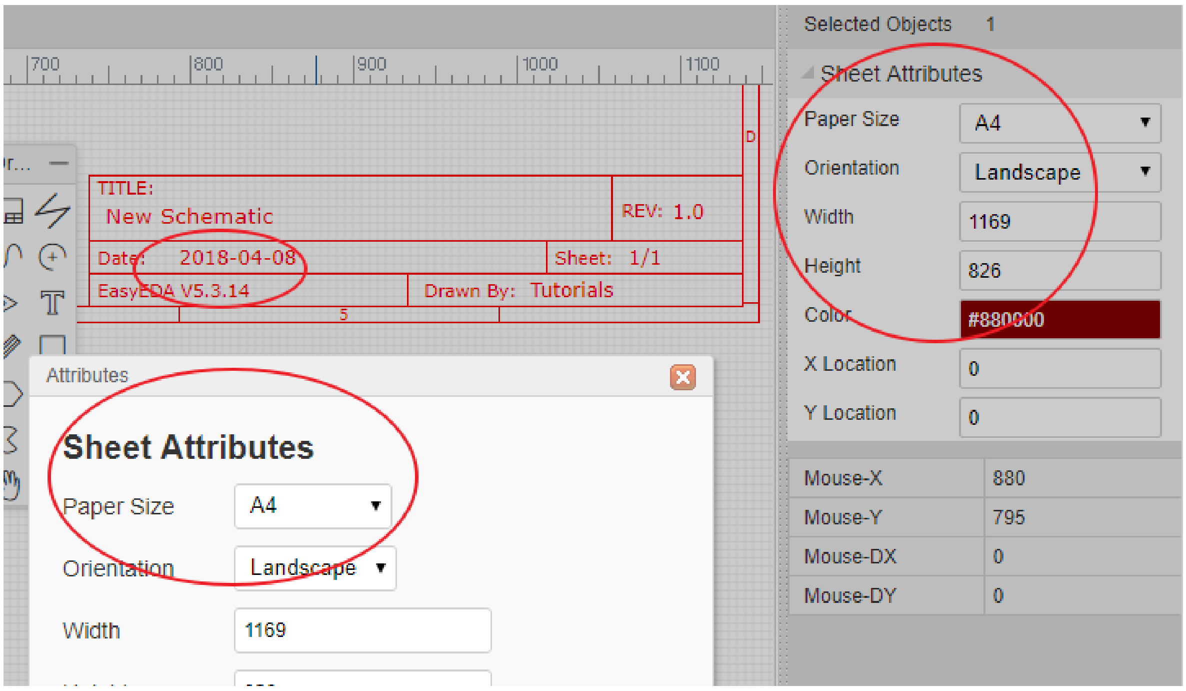
To change the sheet size, move the mouse anywhere over the lower right area of the drawing border or frame until the whole border highlights red and then right-click on it. Paper size and orientation can then be changed in `Sheet Attributes` in the right hand panel.

To modify the design information, left-click on the relevant blue text in the lower right area of the drawing border or frame to change it in `Text Attributes` in the right hand panel. Double left-clicking the blue text will allow you to type new information directly into the field.


Sheet Setting

Sheet Size: A4, 1169 * 826 (px)

Orientation: Landscape



TITLE: New Schematic | REV: 1.0
Date: 2018-04-08 | Sheet: 1/1
EasyEDA V5.3.14 | Drawn By: Tutorials

Attributes

Sheet Attributes

Paper Size	A4
Orientation	Landscape
Width	1169

Selected Objects 1

Sheet Attributes

Paper Size	A4
Orientation	Landscape
Width	1169
Height	826
Color	#880000
X Location	0
Y Location	0

Mouse-X	880
Mouse-Y	795
Mouse-DX	0
Mouse-DY	0

How to indicate low electronic level in the Schematic Pin or Netlabel

You can add a `#` characater in the pin name/netlabel last text. You can use symbols that you are familiar with. You do not have to add a line above the netlabel name.

I can't convert schematic to PCB. Why is this?

1. You have not set the right footprints for your components.
2. Prefix Conflict Error
3. Invalid footprints

PCB

How to solve the PCB very slow issue

Because of v6.3 add two features:

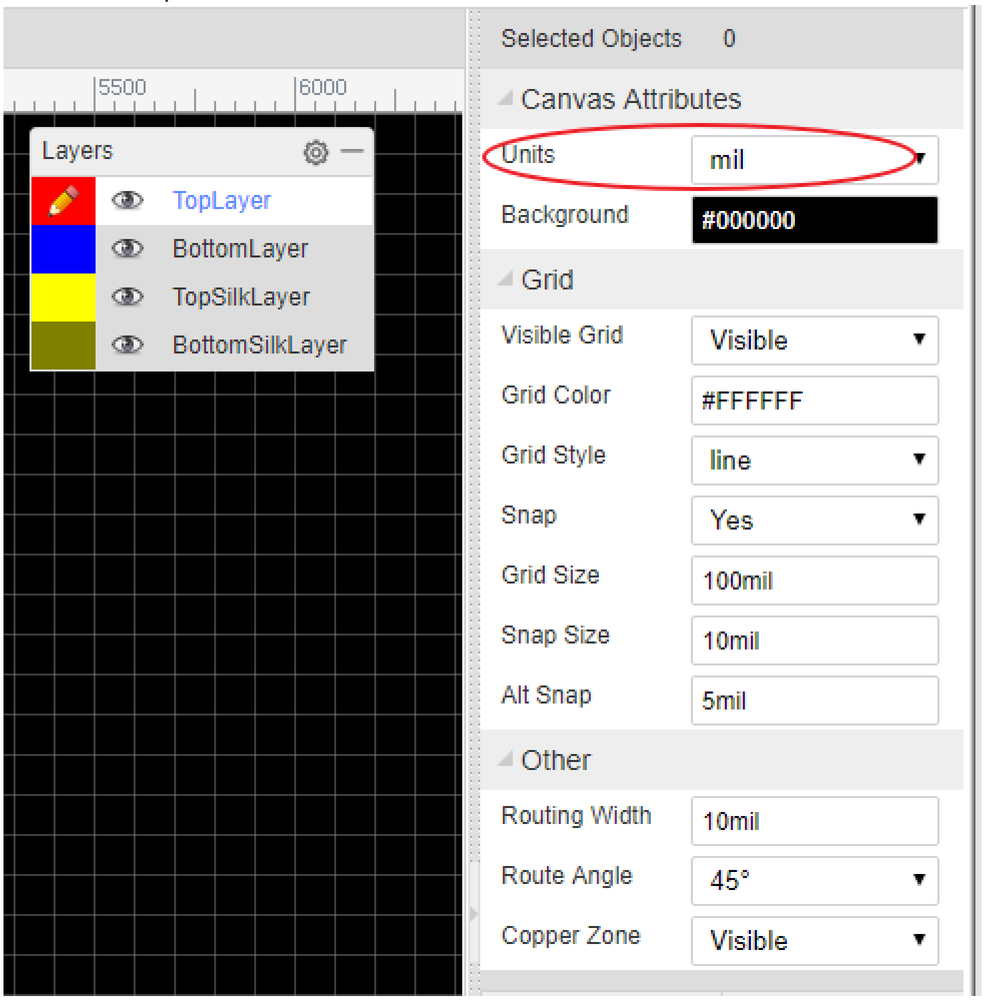
1. All objects support show the ratlines which are containing nets;
2. Solve bottom layer pads will cover top copper issue.
   These features will make editor slower than before.

There are some methods to try to improve：

1. Via Tools - Design Rule, disable real-time DRC
2. Via Settings - PCB settings, disable Add Teardrop Automatically
3. Via Settings - PCB settings, disable Net Highlighting While Coursor Hover the Track
4. Via Settings - PCB settings, disable The Track's Routing Follows Component's Rotation
5. Via Settings - PCB settings, disable Rebuild Plane Automatically, and using copper area instead of inner plane layer
6. Via Settings - PCB settings, set Canvas Zoom Effect as Speed Priority
7. Hide ratline layer before move footprints at PCB
8. Check footprints which have a lot pad's, and the pads are polygon type, please modify them as Retangle or Oval
9. Suggest PCB doesn't over 300 footprints, Pads and Vias no more than 1500

How to change the Units of PCB from mil to mm or inch.

There is an option for that in PCB canvas attributes:



Selected Objects 0

Canvas Attributes

Units	mil
Background	#000000

Grid

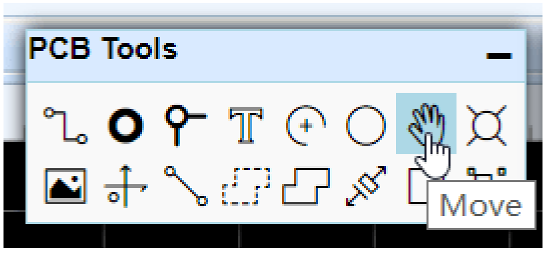
Visible Grid	Visible
Grid Color	#FFFFFF
Grid Style	line
Snap	Yes
Grid Size	100mil
Snap Size	10mil
Alt Snap	5mil

Other

Routing Width	10mil
Route Angle	45°
Copper Zone	Visible

How to pick and move the components on the PCB canvas quickly.

Before routing the PCB, the components need to be positioned in suitable places on the PCB. In the PCB Editor, it can sometimes be quite difficult to select components by clicking on the silkscreen outline or the pads. To select and move them more easily, please use drag mode (Hot Key `D`) or click the `Move` icon in the PCB Tools toolbar:



How to add test point in schematic or PCB?

Schematic: You can place a single pin connector from EElib, and then update its footprint.

PCB: You can place a top/bottom layer pad , and then route it with track.

Can I create a PCB without creating schematic?

Yes but for any but the simplest PCBs, please see:

[Layout PCB Without Schematic]()

How to add more fonts for PCB.

You can refer to Text of PCB section.

How to insert an Image/Logo to PCB.

You can refer to Image of PCB section.

How to insert a DXF as board outline.

You can refer to Import DXF File of Import section.

How to create non rectangular pcb outline such as round?

You can import a DXF file for the board outline. For a round board outline, you can use an arc to do that, you just need to change to the board outline layer, then draw 1 arc like in the image below (need to adjust a bit later), you can use lines and arcs to create complex board outlines.

How to add a slot and cut out.

Please use solid region https://docs.easyeda.com/en/PCB/PCB-Tools/index.html#Solid-Region

Or draw a track and right-click it, use the "Convert to Board Cutout" option.

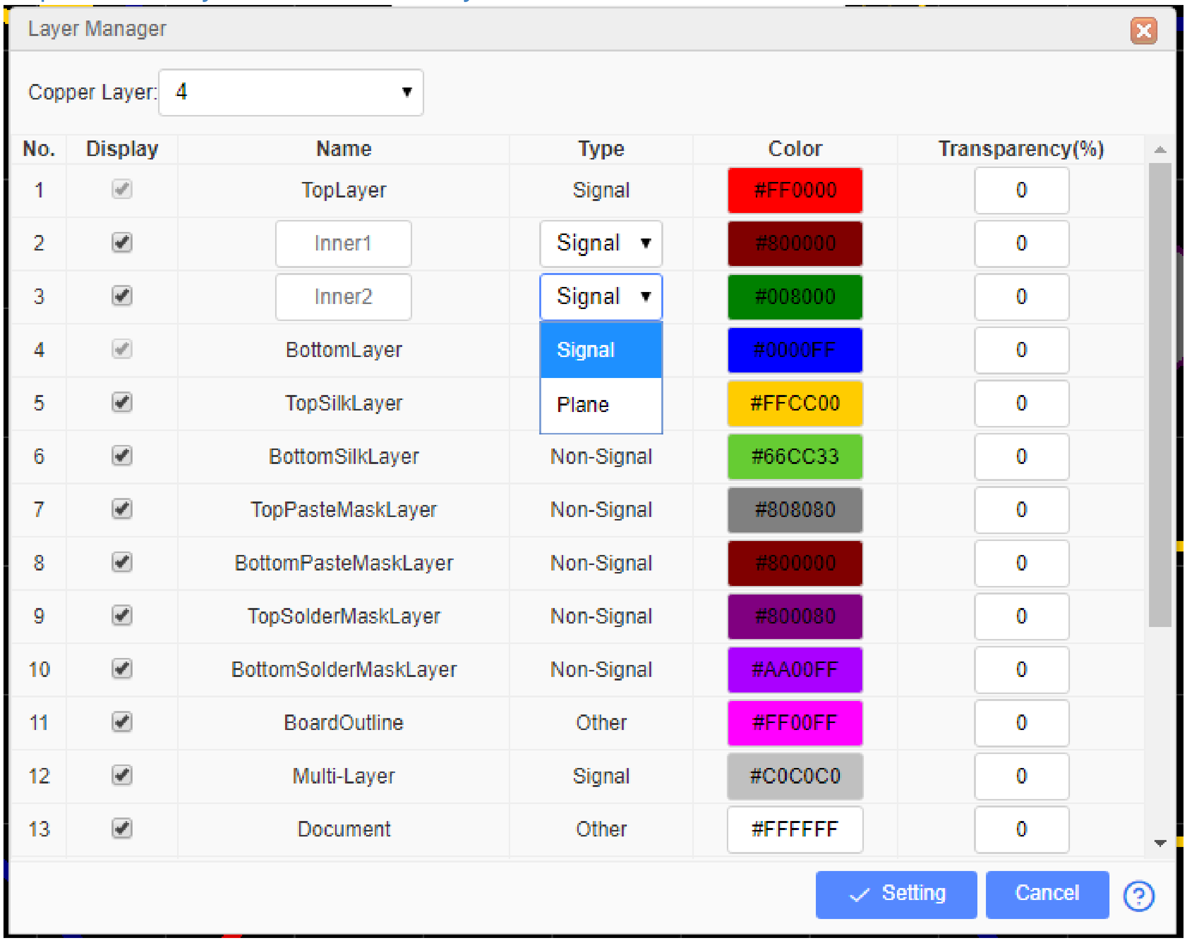
How to measure dimensions on a PCB.

https://docs.easyeda.com/en/PCB/PCB-Tools/index.html#Measure-Dimension

use Hotkey M

How to add more layers.

Click the layer options button, then tick the extra layers in the dialog that opens.
https://docs.easyeda.com/en/PCB/Layers-Tool/index.html


Layer Manager

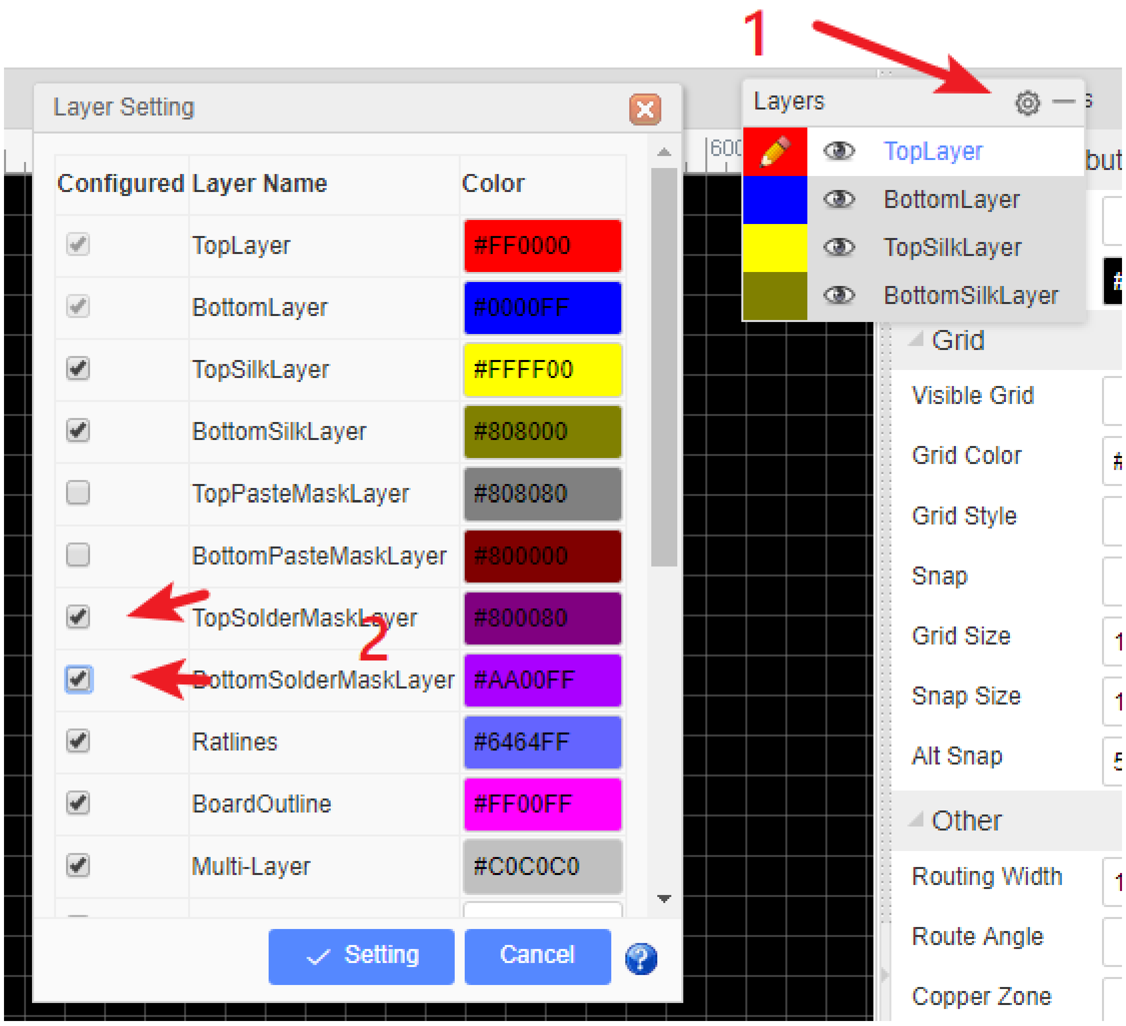
Copper Layer: 4

No.	Display	Name	Type	Color	Transparency(%)
1	✓	TopLayer	Signal	#FF0000	0
2	✓	Inner1	Signal	#800000	0
3	✓	Inner2	Signal	#008000	0
4	✓	BottomLayer		#0000FF	0
5	✓	TopSilkLayer		#FFCC00	0
6	✓	BottomSilkLayer	Non-Signal	#66CC33	0
7	✓	TopPasteMaskLayer	Non-Signal	#808080	0
8	✓	BottomPasteMaskLayer	Non-Signal	#800000	0
9	✓	TopSolderMaskLayer	Non-Signal	#800080	0
10	✓	BottomSolderMaskLayer	Non-Signal	#AA00FF	0
11	✓	BoardOutline	Other	#FF00FF	0
12	✓	Multi-Layer	Signal	#C0C0C0	0
13	✓	Document	Other	#FFFFFF	0

How to add solder mask aperture.

It is possible to get boards with the copper exposed so that you can apply a layer of solder over those tracks to further increase their current carrying capacity. In this case, you need to add solder mask over a copper (copper area, track, solid region).
EasyEDA will add solder mask for pads automatically. Sometimes however, you may need to add an aperture in the solder mask to expose and area of copper.

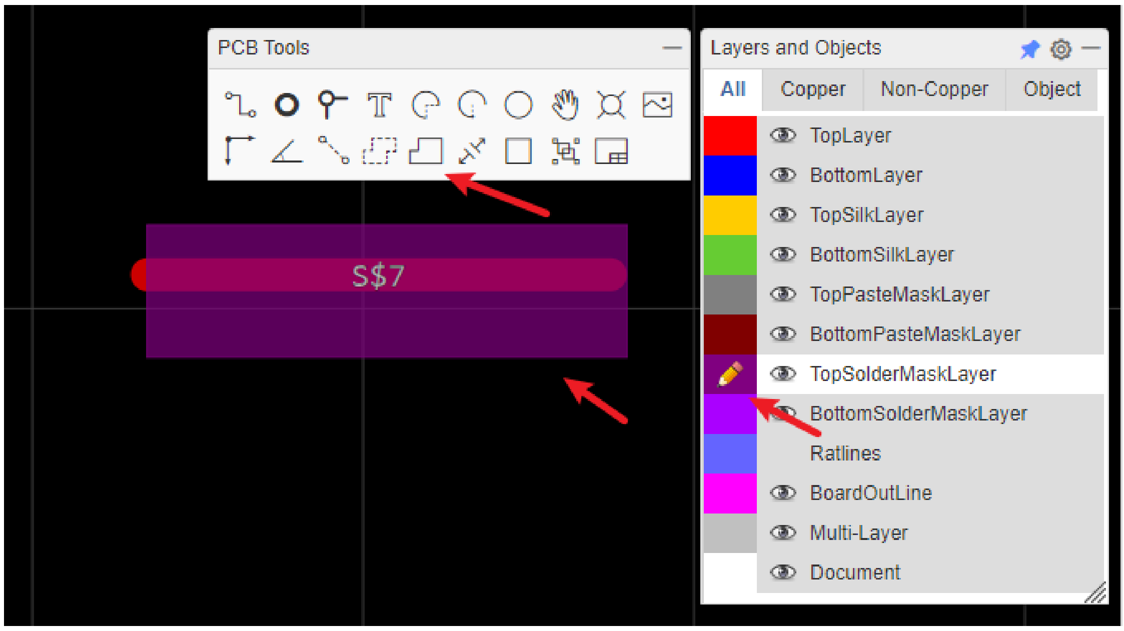
1. First, add a top or bottom solder mask layer, as required.



Layer Setting

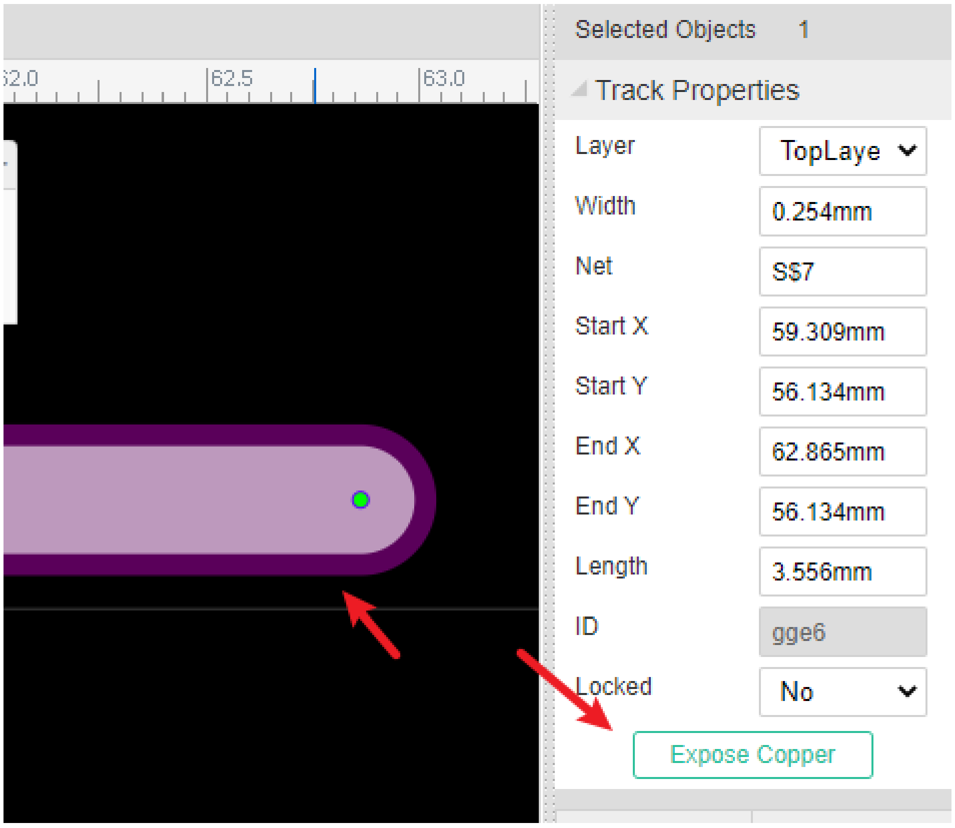
Configured	Layer Name	Color
✓	TopLayer	#FF0000
✓	BottomLayer	#0000FF
✓	TopSilkLayer	#FFFF00
✓	BottomSilkLayer	#808000
☐	TopPasteMaskLayer	#808080
☐	BottomPasteMaskLayer	#800000
✓	TopSolderMaskLayer	#800080
✓	BottomSolderMaskLayer	#AA00FF
✓	Ratlines	#6464FF
✓	BoardOutline	#FF00FF
✓	Multi-Layer	#C0C0C0

2. Next, draw a region in the solder mask layer over a copper item as illustrated in the image below. This in effect draws an aperture in the solder mask so that the copper item inside the region, in this case the track, will be not be covered by the green film of solder mask.



A common mistake is to just draw a solder mask, without a copper area, like the track pointed to by the yellow arrow. That is incorrect and does not produce the desired result.

Or you can click the track, and then click the `Expose Copper` button at the right-hand panel.



Selected Objects 1

Track Properties

Layer	TopLaye
Width	0.254mm
Net	S$7
Start X	59.309mm
Start Y	56.134mm
End X	62.865mm
End Y	56.134mm
Length	3.556mm
ID	gge6
Locked	No

How do I set the dimensions of my PCB in the layout?

PCB's dimension/size depends on the board outline, you can create your board outline, please refer to Board Outline of the PCB section.

My PCB is complex, how can I be sure that I have routed all of the tracks?

Please refer to Design Manager of PCB section.

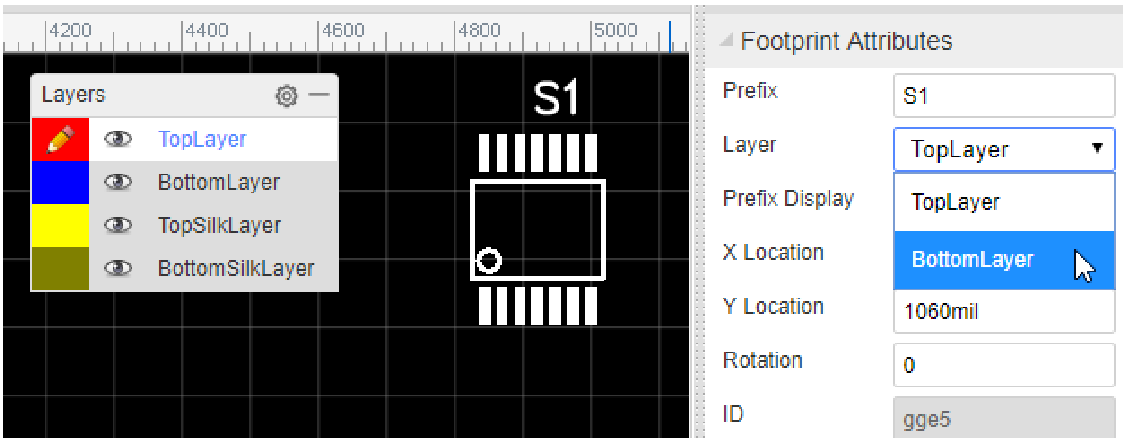
I need to start my layout again, how can I remove all of the tracks?

You can use "Menu - Route - Unroute All" and "Menu - Edit - Global Delete".

How to put a component on the bottom layer?

There are two ways to do this.

1. If your active layer is the bottom layer, then every component you place will be placed on the bottom layer automatically.
2. You can place a component then select it and change its layer attribute to `Bottom layer` in the right hand panel.



How to panelize the PCB

Please refer at [PCB: Panelize]()

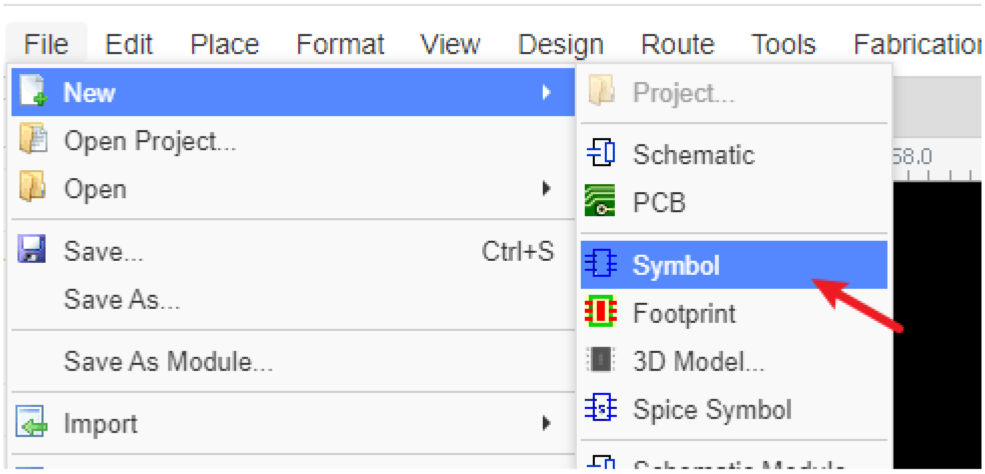
What does Warning copper area do not allow self intersection

Please refer at [Forum: What does Warning copper area do not allow self intersection]()

Library and Parts

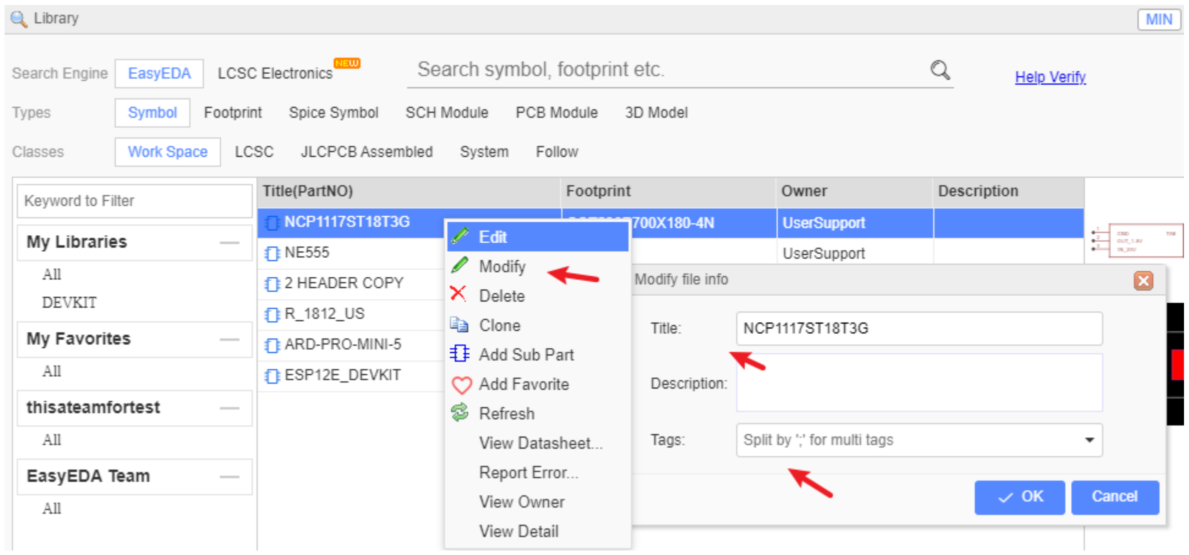
How to create a schematic symbol library.

File > New > Symbol



How to tag my schematic library symbol.

After creating the library and saving, you can add a tag for it, and you can add and edit the tag at "Library":



Title(PartNO)	Footprint	Owner	Description
NCP1117ST18T3G	700X180-4N	UserSupport	
NE555		UserSupport	
2 HEADER COPY			
R_1812_US			
ARD-PRO-MINI-5			
ESP12E_DEVKIT			

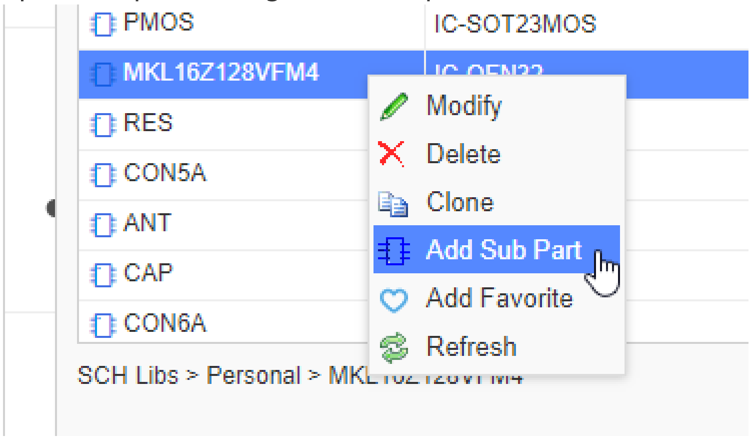
Title: NCP1117ST18T3G

Description:

Tags: Split by ';' for multi tags

How to create sub parts for multi-part components.

In personal part list, Right click the part then select **Add Sub Part** from the menu that opens:



PMOS	IC-SOT23MOS
MKL16Z128VFM4	IC-QFN32
RES	
CON5A	
ANT	
CAP	
CON6A	

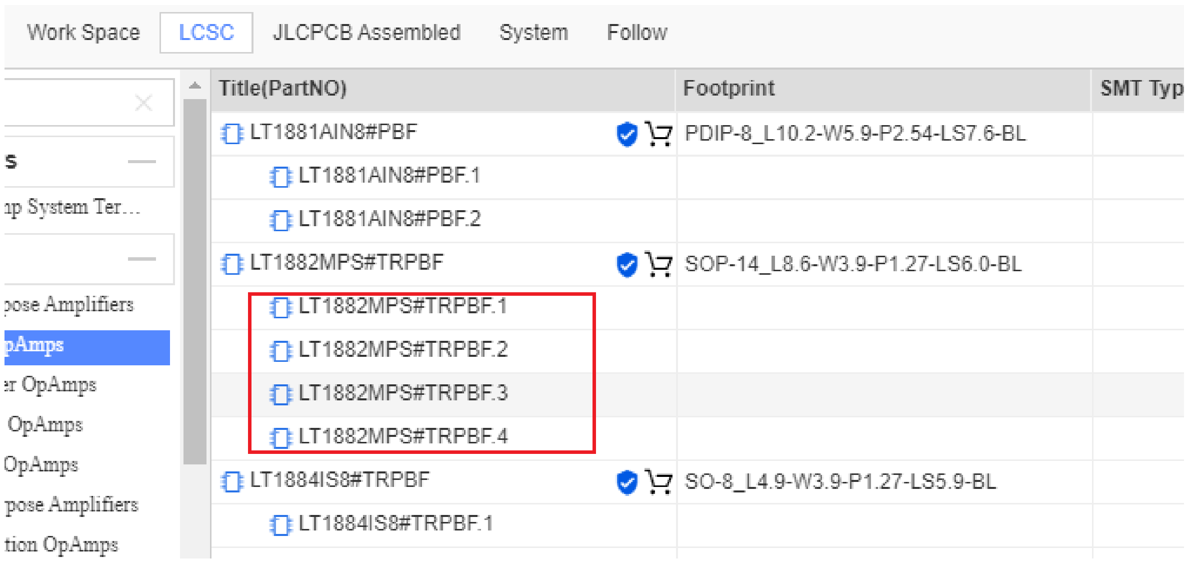
SCH Libs > Personal > MKL16Z128VFM4

How to change the footprint for a component.

https://docs.easyeda.com/en/Schematic/Footprint-Manager/index.html

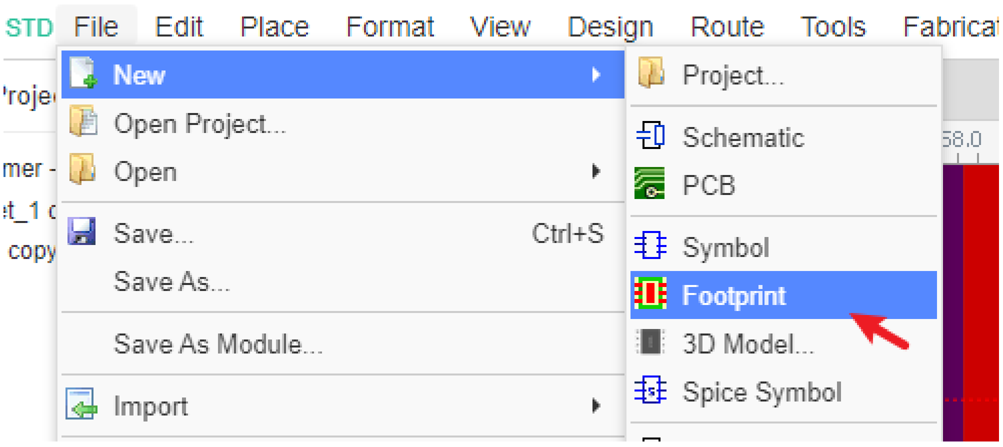
How to add sub parts to a schematic.

You can add sub parts to a schematic one by one but please note that the sub parts prefix must be in the form of U1.1 U1.2 etc, and not U1.A U1.B.



Title(PartNO)	Footprint	SMT Typ
LT1881AIN8#PBF	PDIP-8_L10.2-W5.9-P2.54-LS7.6-BL	
LT1881AIN8#PBF.1		
LT1881AIN8#PBF.2		
LT1882MPS#TRPBF	SOP-14_L8.6-W3.9-P1.27-LS6.0-BL	
LT1882MPS#TRPBF.1		
LT1882MPS#TRPBF.2		
LT1882MPS#TRPBF.3		
LT1882MPS#TRPBF.4		
LT1884IS8#TRPBF	SO-8_L4.9-W3.9-P1.27-LS5.9-BL	
LT1884IS8#TRPBF.1		

How to create a PCB footprint.

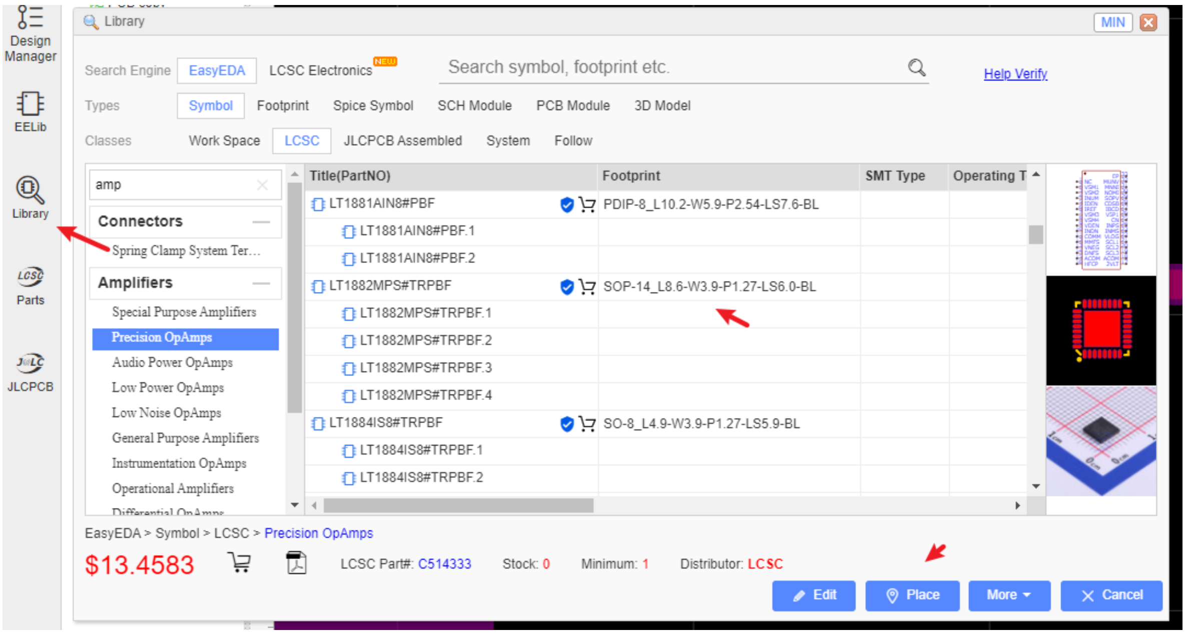


How to change a component's footprint?

please refer at [Footprint Manager]()

How to find components/parts/libraries?

The component search function has been significantly improved to make finding part symbols and footprints quicker and easier. Press `SHIFT+F` or click on the `Libraries` icon on the left navigation panel:



Title(PartNO)	Footprint	SMT Type	Operating T
LT1881AIN8#PBF	PDIP-8_L10.2-W5.9-P2.54-LS7.6-BL		
LT1881AIN8#PBF.1			
LT1881AIN8#PBF.2			
LT1882MPS#TRPBF	SOP-14_L8.6-W3.9-P1.27-LS6.0-BL		
LT1882MPS#TRPBF.1			
LT1882MPS#TRPBF.2			
LT1882MPS#TRPBF.3			
LT1882MPS#TRPBF.4			
LT1884IS8#TRPBF	SO-8_L4.9-W3.9-P1.27-LS5.9-BL		
LT1884IS8#TRPBF.1			
LT1884IS8#TRPBF.2			

In the new components dialog, it is easy to select the right components via tags and you can set tags for your own components.

Fabrication and Order

How to Order PCB

1. Before ordering, please check this Gerber first: https://docs.easyeda.com/en/PCB/Gerber-Generate/index.html#Gerber-View

2. Visit and login at https://jlcpcb.com/quote
3. Add this Gerber file(compressed file) on the page and type the order options
4. Save to Cart, and then submit the payment

If you want to combine the components order with the PCB order at https://lcsc.com , please refer:
https://support.lcsc.com/article/24-do-you-offer-combine-shipment-with-pcbs

please refer at Order PCB

How export BOM and order parts?

please refer at Export BOM

Import and Export

How to import Alitum/Eagle/Kicad File?

Please refer at Import Altium Designer

Can I export my design?

Yes, you can export your design as EasyEDA format or Alitum Designer format. Please refer at:

Export EasyEDA format

Export Alitum Designer Format

How to export or print the schematic or PCB?

please refer at Export

Save and Backup

Where are my files?

Your files are stored on EasyEDA servers, so you can access them anywhere and share them with your partners.

if you using EasyEA desktop client, you can set the runing mode as "Project Offline Mode", it will save your project to local.

How to save my file to the local?

You can download the project via :

- Right-click project and download;
- Download the EasyEDA Source
- Export to Altium file Export: Export Altium

How to recover the deleted file?

1. Check "Recycle Bin" at the editor bottom-left icon, find and recovery. When using desktop client "Project Offline" mode doesn't support the Recycle Bin.
2. Find it back at "Document Recovery" function. Via: Top Menu - Advanced - Document Recovery.

I don't like others seeing my design. How can I stop that happening?

Set your project as Private. For extra security you can even save your work locally.

as above to save your file to locad as EasyEDA format.

Is EasyEDA safe?

There are no absolutely secure things in the world but even if you have the misfortune - as happened to one of our team - of losing one laptop and having two hard drives break, EasyEDA will try to protect your designs in following ways:

1. We utilize SSL throughout the entire domain EasyEDA.com. Secure Socket Layer (SSL) technology encrypts all data transferred between your computer and our servers. Your data is for your eyes only.
2. You can save your files locally.
3. Multiple copies of every file are saved in your local database.
4. EasyEDA servers backup your designs frequently.

What if EasyEDA cannot become self sustaining and has to close down?

We promise to do our best to ensure that neither of these things will happen; we have spent so much of our time to get to this point. We promise that if we cannot make enough money out of EasyEDA to keep it alive or to fund further development, we will not simply abandon our baby or our community but we will consider donating the code to the Open Source Community to let them build on our efforts. There are no companies who can stay forever, so if a time comes when we have to close down, we will follow the steps below:

1. Give our users six months warning prior to closure;
2. Ensure all our users can backup their designs;
3. Ensure that user's designs can be exported to some other EDA tools, such as Kicad, Altium Designer and others.
4. Package our codes, so that users can install an EasyEDA in their own OS (Windows, Linux, Mac). Users can then build their own cloud EDA.
5. Upload our codes to github.com and make them open source.

So, nothing will be lost and our users can continue to enjoy an awesome web based EDA tool that lets them stay in charge of their designs: anywhere, anytime and on any OS.

How to backup my project?

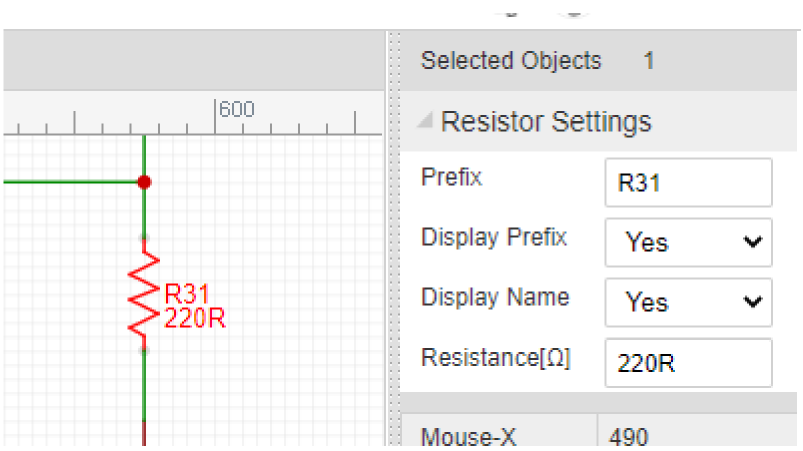
Please refer at: Saving Your Work Locally

Spice Simulation FAQ

EasyEDA's main target is schematic and PCB, not simulation. EasyEDA only support simple schematics simulation.

How to set the resistance of a resistor

You can use the name attribute. Just set the name or double click the value text.

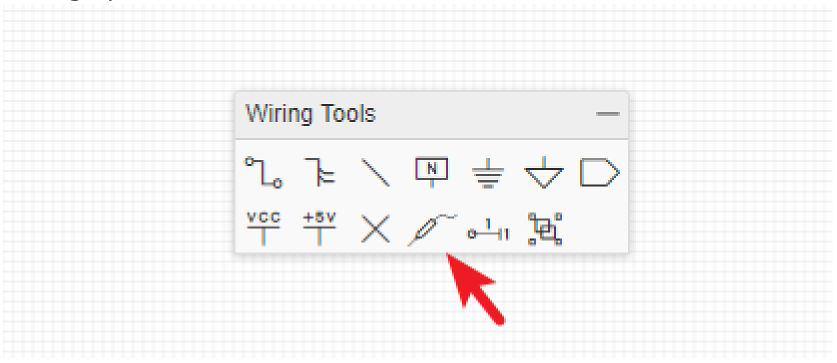

Selected Objects 1

Resistor Settings

Prefix	R31
Display Prefix	Yes
Display Name	Yes
Resistance[Ω]	220R
Mouse-X	490

Where Can I find the Probe/voltage probe?

Voltage probe



Why I can't simulate my schematic

EasyEDA only has very few simulation models, EasyEDA is powered by LTSpice, please check LTspice to know what can be simulated.

How to ask for help and get an answer

[Must read] How to ask for help and get an answer

Others

Can I use EasyEDA in my company?

You are free to use EasyEDA for individuals, business and education.
If you add our Logo and link on your PCB/Video we will appreciate.

What happens if EasyEDA service is offline for some reason?

EasyEDA can be run as an offline application. You can export your design first, when the service back, you can import the design and save to EasyEDA server.

Or you can use EasyEDA desktop client "Project Offline" mode.

How to find the list of hotkeys.

Please refer at Introduction: Shortcut Keys

Why does EasyEDA focus on Cloud based EDA?

EasyEDA is built for people who like to work anywhere, who like to build projects together with other team members, who like to share their projects, who like something that operates like a github for hardware design. The only way to meet these needs is to build a Cloud version EDA.

How can I work if there is no internet?

Although most of the time there are ways to access the internet easily and cheaply there may be times when, for whatever the reason, internet access is simply not possible. For times like this, EasyEDA is working to provide a desktop client soon.

Does EasyEDA have a desktop version?

Yes, please refer at: https://easyeda.com/page/download

Which Browser is best for EasyEDA?

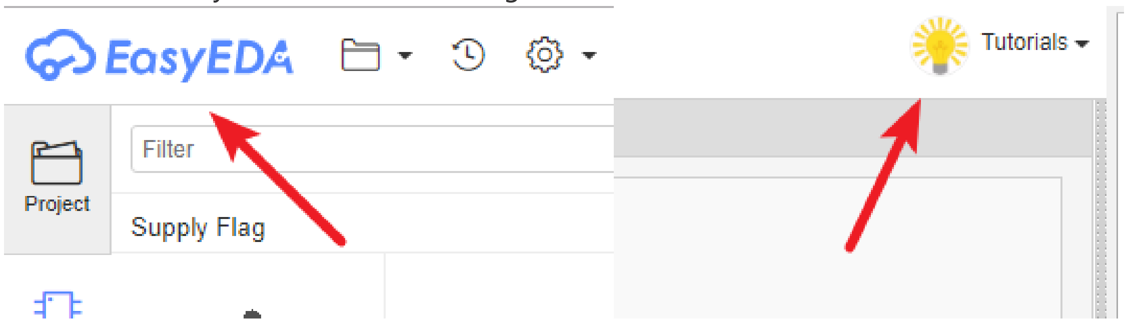
The latest **Chrome** and **Firefox**. If you are restricted to using other browsers, it would be better to download the EasyEDA desktop client.

How to go to your dashboard.

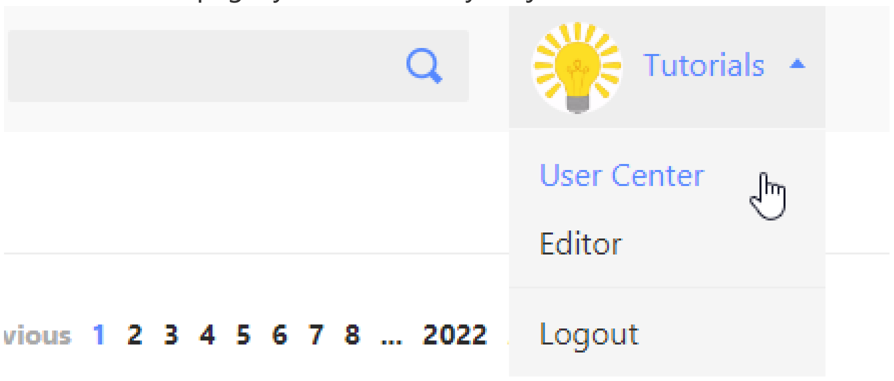
In the User Center, you can check all your Projects, Modules, Libraries and Friends, Messages etc.

There are two ways to arrive there.

1. From the Editor, you can click on user logo:

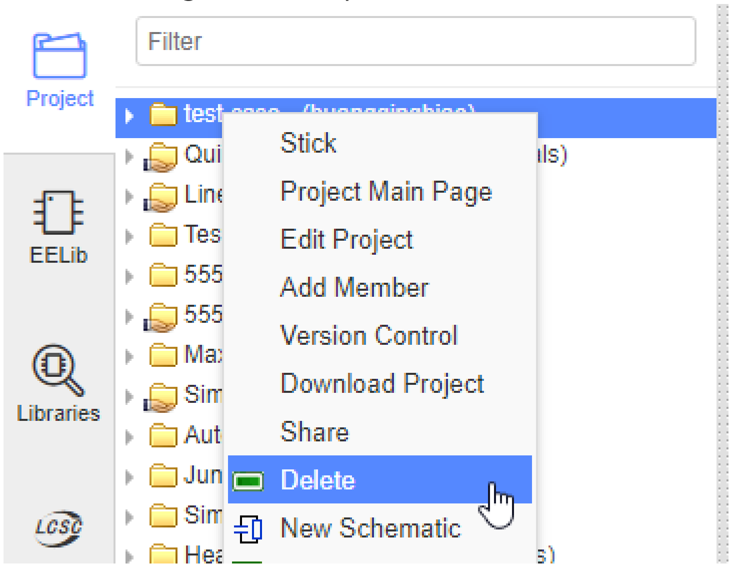


2. From the homepage, you can click My Projects:



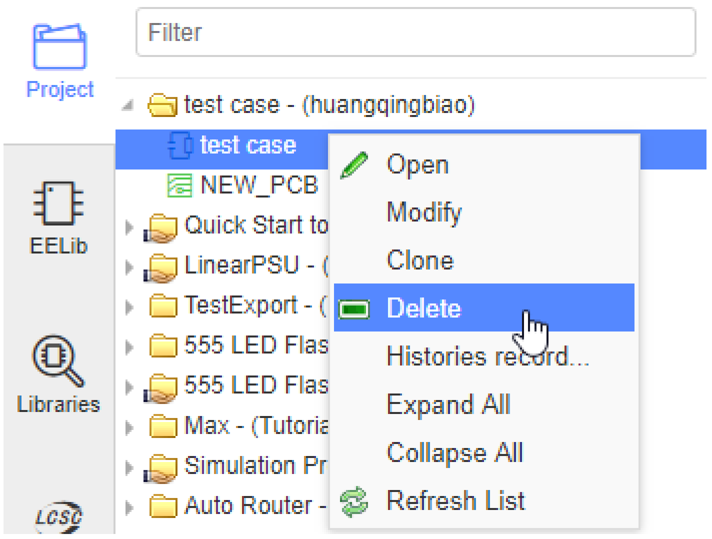
How to delete a project.

Select it and right click to open a context menu, like the image below.



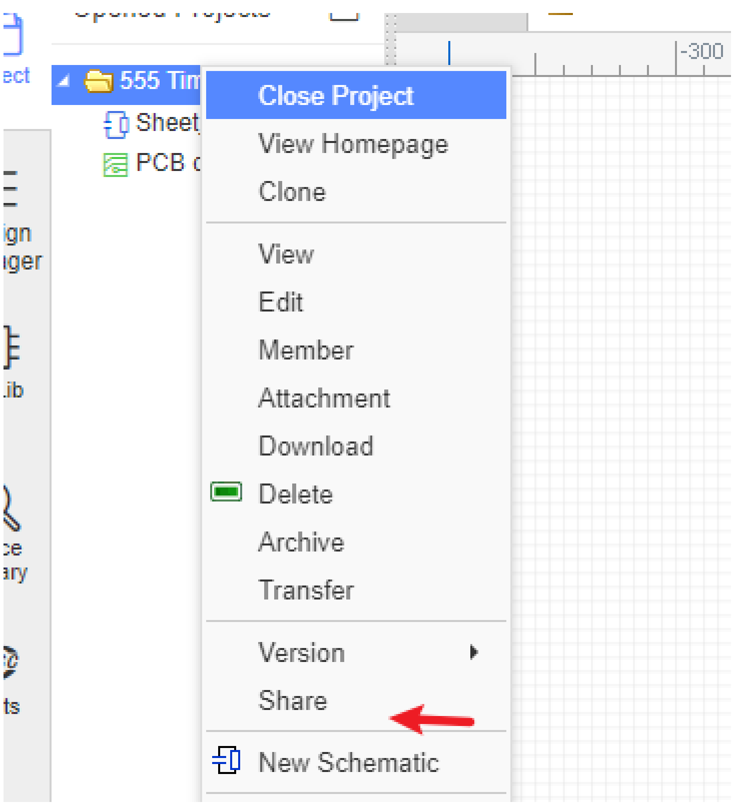
How to delete a schematic or PCB.

Select it and right click to open a context menu, like the image below.



How to share a project with others.

1. Make your project public.



2. To share a project privately with only selected collaborators via: [Add Member](#)

How to find the version history of schematics and PCBs.

The version history of your EasyEDA schematics and PCBs can be accessed by right-clicking on the file you wish to query to open the context menu as shown in the image below:

Then click on the version number that you wish to view.

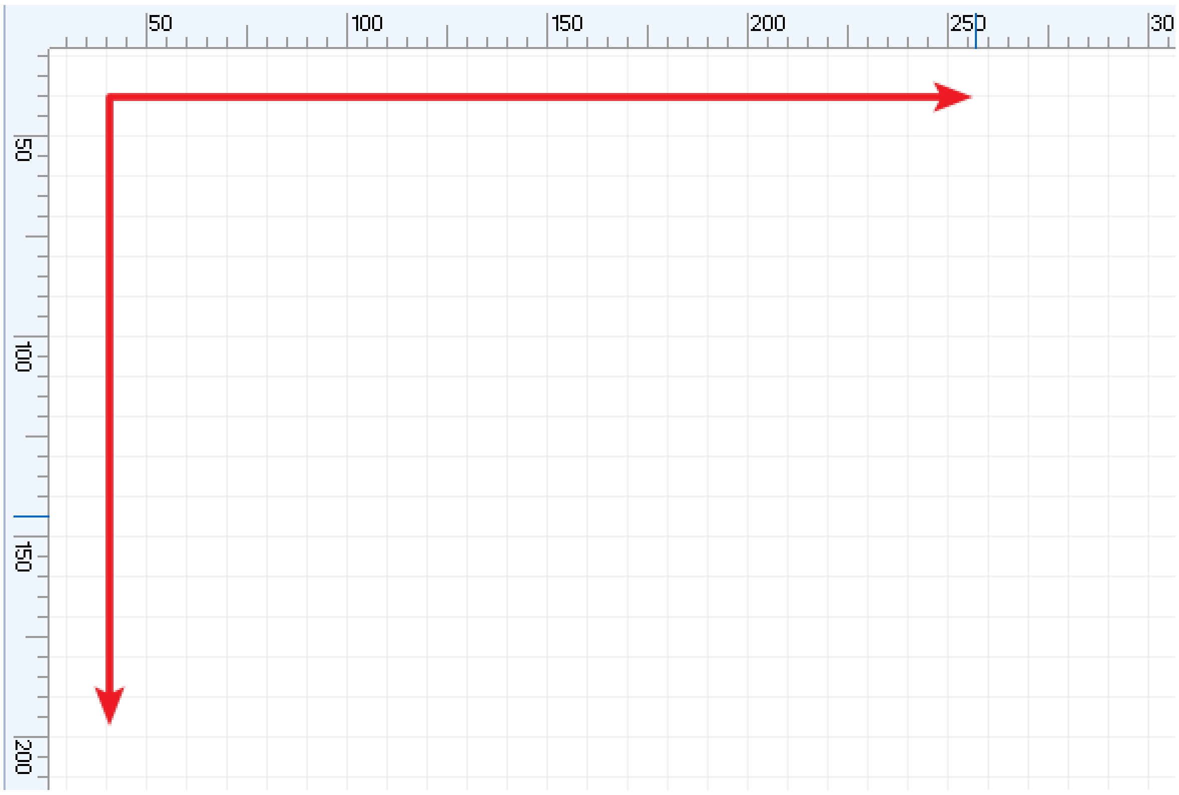
Note: *saving a previous version will restore that version to being the current version of the file.*

Does EasyEDA canvas use the Cartesian coordinate system?

Yes and no.

It uses X and Y coordinates where the horizontal X coordinate is positive to the right of the origin and negative to the left but the vertical Y coordinate is positive **below** the origin and negative above it.

Actually, we think our coordinate system is not very good but it is hard to change.



How to update editor to latest version and how to remove editor cache?

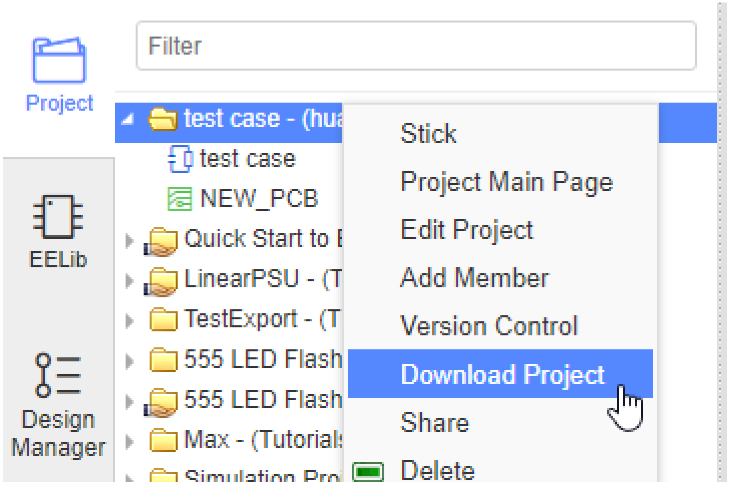
please refer at: How to Update

Essential checks before placing a PCB order

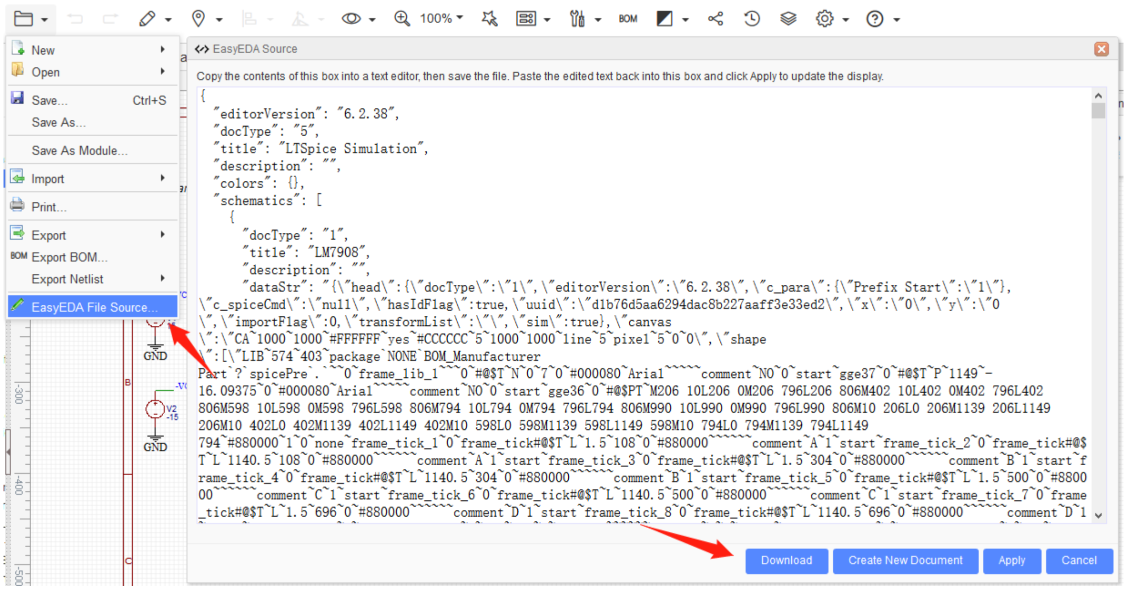
Please refer Essential checks before placing a PCB order

Keep in Mind

1. After the first save of any file, EasyEDA will back up all saved files automatically under the Version Control. If you want to back up your files locally, you can download a copy of the whole project or of individual files in a project in EasyEDA Source (JSON) format:



and **File > EasyEDA File Source > Download**



Copy the contents of this box into a text editor, then save the file. Paste the edited text back into this box and click Apply to update the display.

```
{
  "editorVersion": "6.2.38",
  "docType": "5",
  "title": "LTSpice Simulation",
  "description": "",
  "colors": {},
  "schematics": [
    {
      "docType": "1",
      "title": "LM7908",
      "description": "",
      "dataStr": "{\"head\":{\"docType\":\"1\",\"editorVersion\":\"6.2.38\",\"c_para\":{\"Prefix Start\":\"1\"},
\"c_spiceCmd\":\"null\",\"hasIdFlag\":true,\"uuid\":\"d1b76d5aa6294dac8b227aaff3e33ed2\",\"x\":\"0\",\"y\":\"0
\",\"importFlag\":0,\"transformList\":\"\",\"sim\":true},\"canvas
\":\"CA~1000~1000~#FFFFFF~yes~#CCCCCC~5~1000~1000~line~5~pixel~5~0~0\",\"shape
\":[\"LIB~574~403~package`NONE`BOM_Manufacturer
Part`?`spicePre`.`~~0~frame_lib_1~~~0~#@$T~N~0~7~0~#000080~Arial~~~~~comment~NO~0~start~gge37~0~#@$T~P~1149~-
16.09375~0~#000080~Arial~~~~~comment~NO~0~start~gge36~0~#@$PT~M206 10L206 0M206 796L206 806M402 10L402 0M402 796L402
806M598 10L598 0M598 796L598 806M794 10L794 0M794 796L794 806M990 10L990 0M990 796L990 806M10 206L0 206M1139 206L1149
206M10 402L0 402M1139 402L1149 402M10 598L0 598M1139 598L1149 598M10 794L0 794M1139 794L1149
794~#880000~1~0~none~frame_tick_1~0~frame_tick#@$T~L~1.5~108~0~#880000~~~~~~comment~A~1~start~frame_tick_2~0~frame_tick#@$
T~L~1140.5~108~0~#880000~~~~~~comment~A~1~start~frame_tick_3~0~frame_tick#@$T~L~1.5~304~0~#880000~~~~~~comment~B~1~start~f
rame_tick_4~0~frame_tick#@$T~L~1140.5~304~0~#880000~~~~~~comment~B~1~start~frame_tick_5~0~frame_tick#@$T~L~1.5~500~0~#8800
00~~~~~~comment~C~1~start~frame_tick_6~0~frame_tick#@$T~L~1140.5~500~0~#880000~~~~~~comment~C~1~start~frame_tick_7~0~frame
_tick#@$T~L~1.5~696~0~#880000~~~~~~comment~D~1~start~frame_tick_8~0~frame_tick#@$T~L~1140.5~696~0~#880000~~~~~~comment~D~1
```

2. If you need help, you can contact us email or ask via our Support Forum; we will respond ASAP.

support@easyeda.com

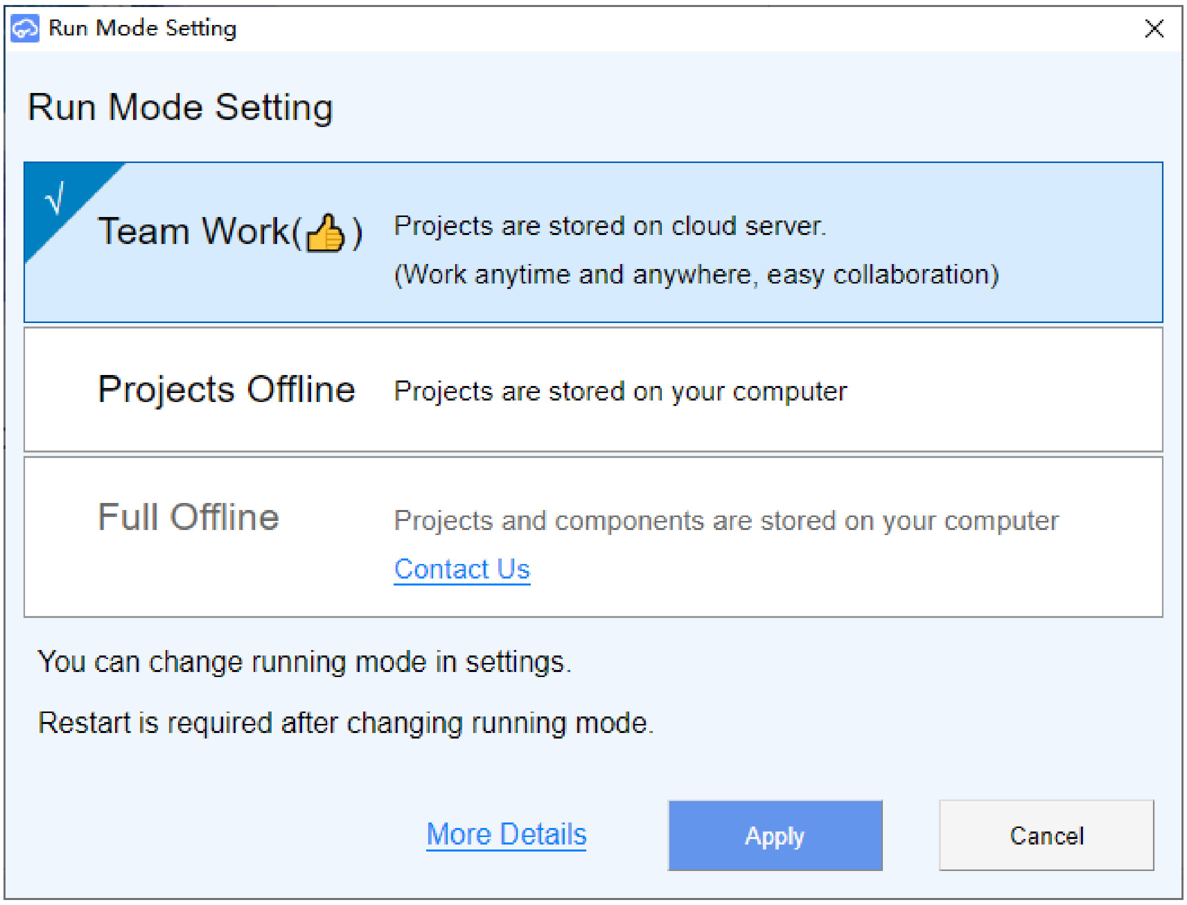
EasyEDA Desktop CLient

Download

Download address: https://easyeda.com/page/download

Client Running Mode

When you install the client first time, you can set the runing mode:


Run Mode Setting

Run Mode Setting

√ Team Work(👍) Projects are stored on cloud server. (Work anytime and anywhere, easy collaboration)

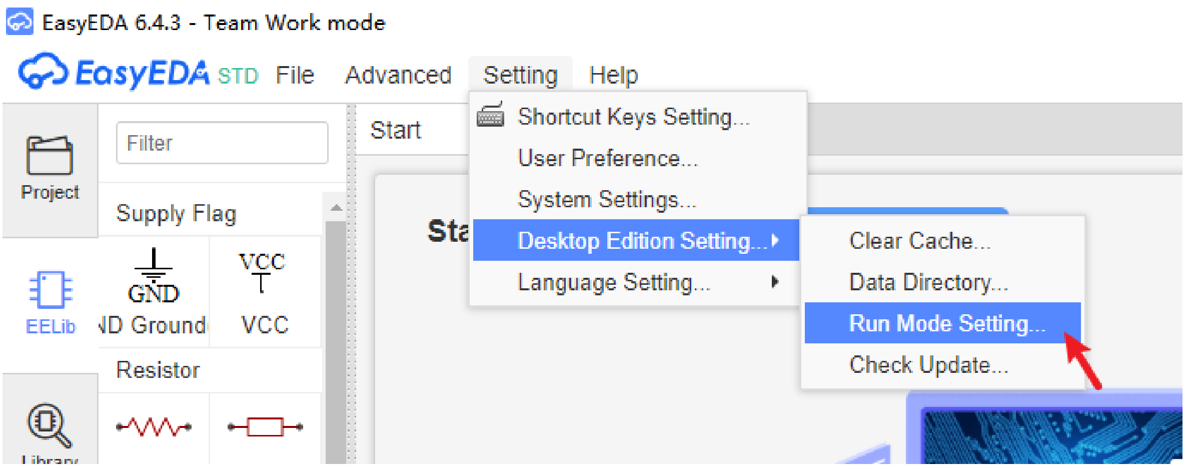
Projects Offline Projects are stored on your computer

Full Offline Projects and components are stored on your computer Contact Us

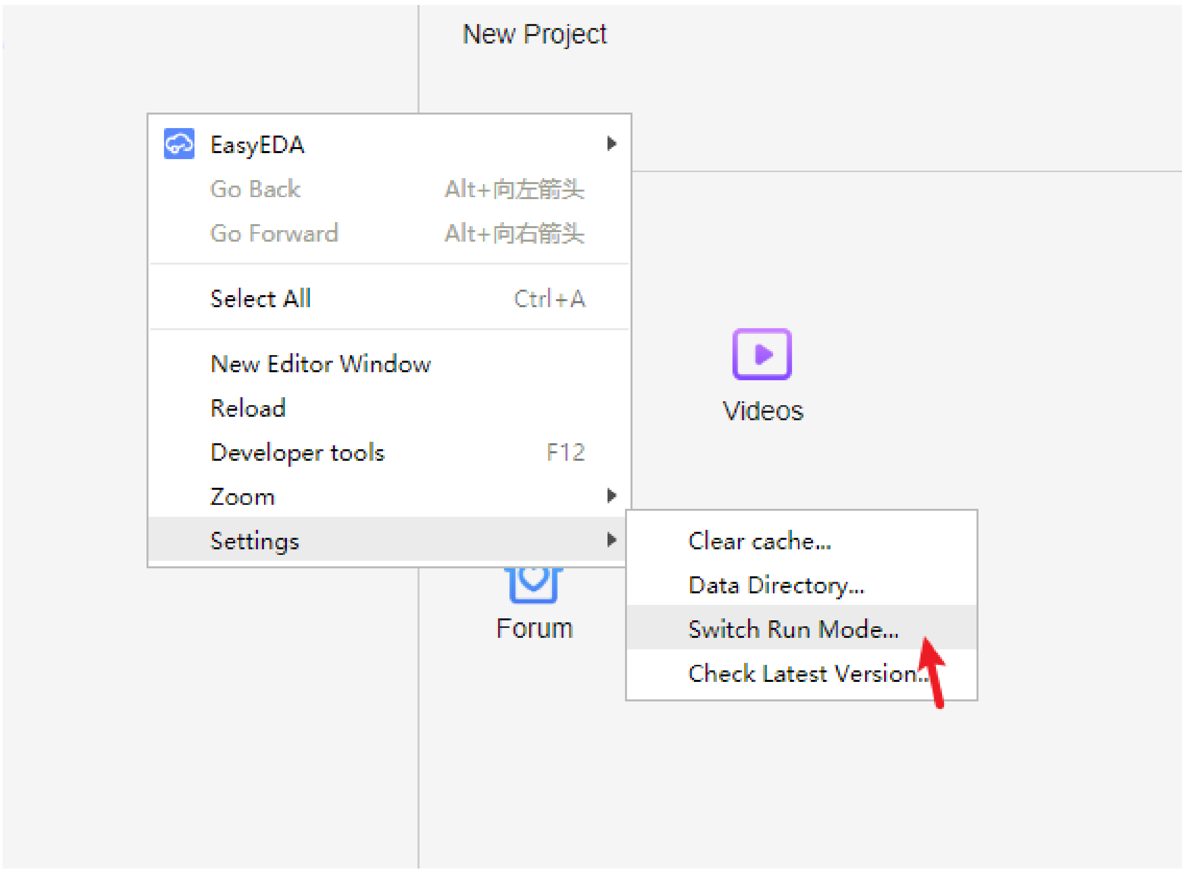
You can change running mode in settings.

Restart is required after changing running mode.

if you want to change the runnig mode after installation, you can via: Top Menu - Setting - Desktop Client Setting - Running Mode Setting



or right-click the start page, via: Setting - Switch Run Mode



Team Work Mode

This version is full function, such as team work, work any time any where.
Project and library are saving at cloud server.

Projects Offline Mode

Project save at local, the library save at the cloud. Only few option needing internet, such as: library searching, library saving, schematic convert to PCB, import changes, etc.

Full Offline Mode

Doesn't provide yet.

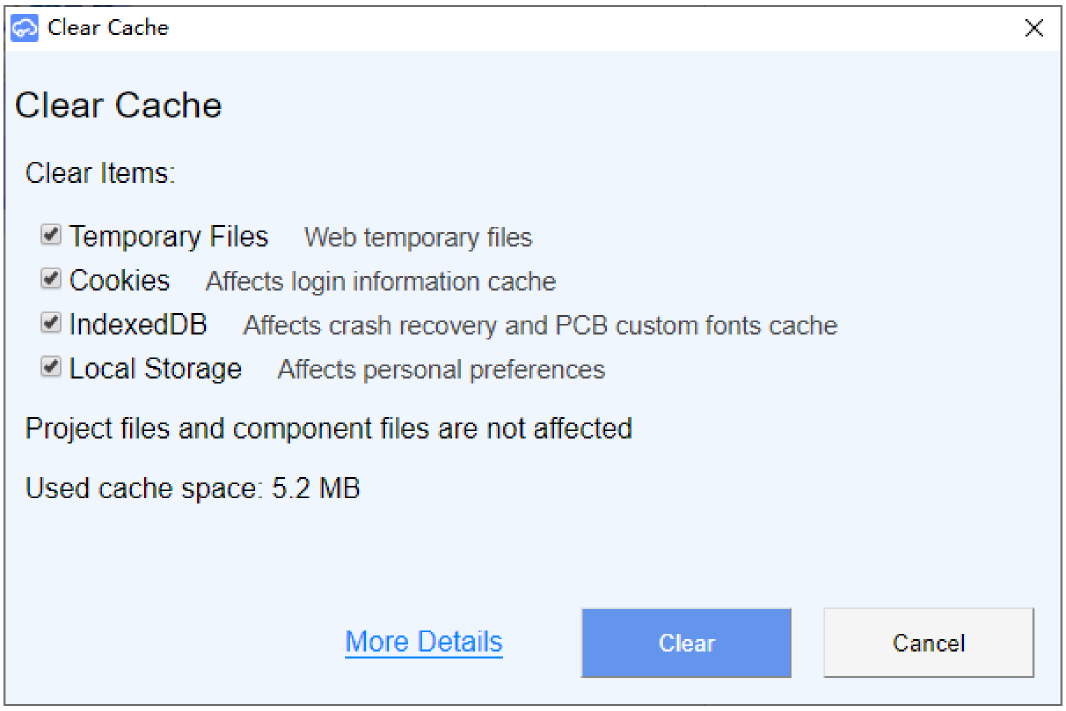
Projects and libraries are saving at local. It is only provide for company. That will take some cost.

Client Setting

Right-click the start page, visit the setting menu. or at： Top Menu - Setting - Desktop Client Setting

- **New Editor Window**: Create a new editor window.
- **Reload**: You can reload the editor.
- **Zoom**: Zoom in or zoom out the editor windows.
- **Setting**:

- **Remove Cache**: Clear editor cache.



Clear Cache

Clear Cache

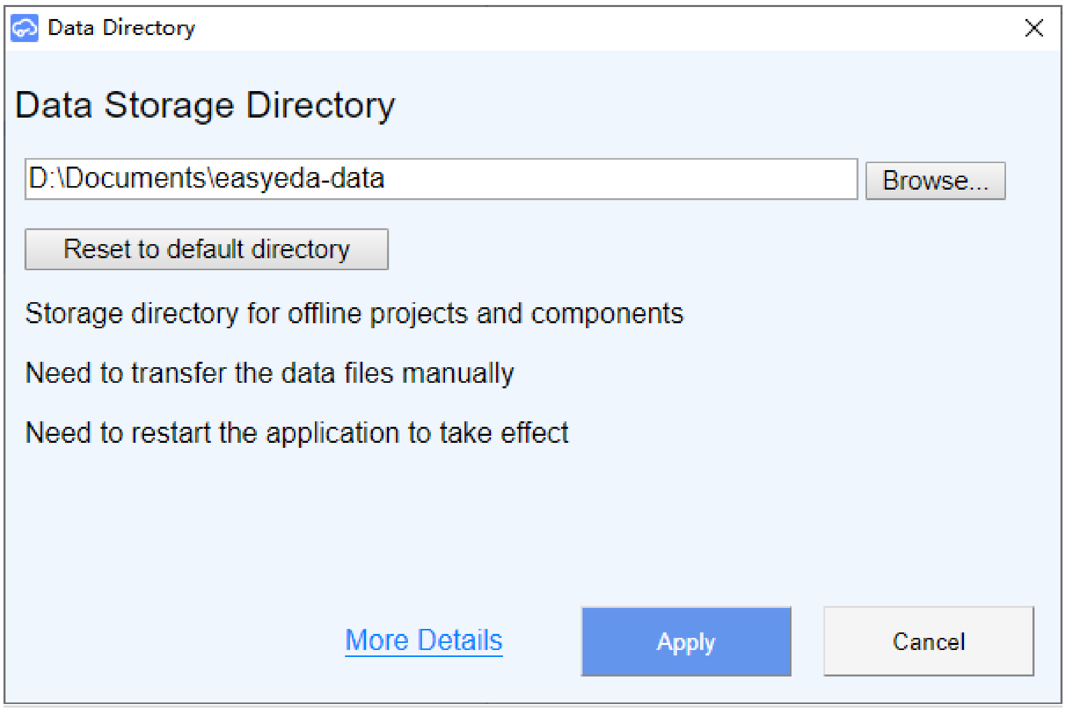
Clear Items:

- ✓ Temporary Files Web temporary files
- ✓ Cookies Affects login information cache
- ✓ IndexedDB Affects crash recovery and PCB custom fonts cache
- ✓ Local Storage Affects personal preferences

Project files and component files are not affected

Used cache space: 5.2 MB

- **Data Saving Directory**: Including offline projects and auto-backup projects saving directory. When you using "Team Work Version", the client will auto-backup your file to this directory, which is named "projects_backup", each signle file you saved will be saving in this directory, if you want to recovery the file from this directory, you can open the backup file at the editor. Or you can use editor Document Recovery function too.



Data Directory

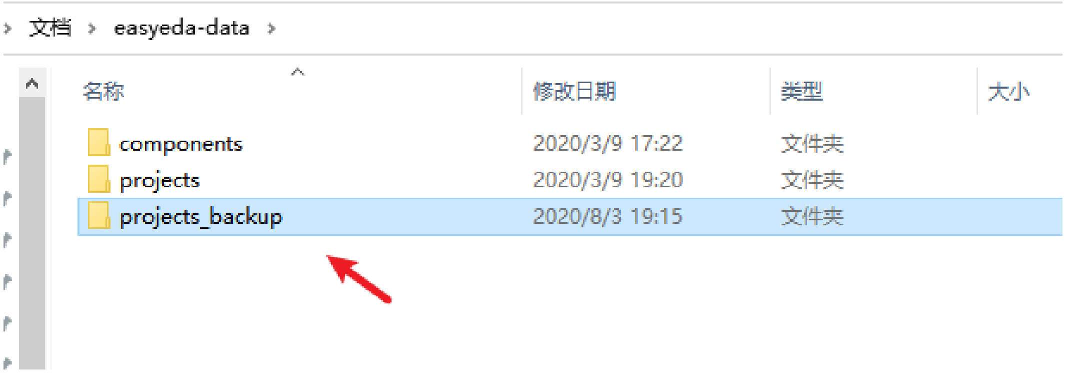
Data Storage Directory

D:\Documents\easyeda-data

Storage directory for offline projects and components

Need to transfer the data files manually

Need to restart the application to take effect

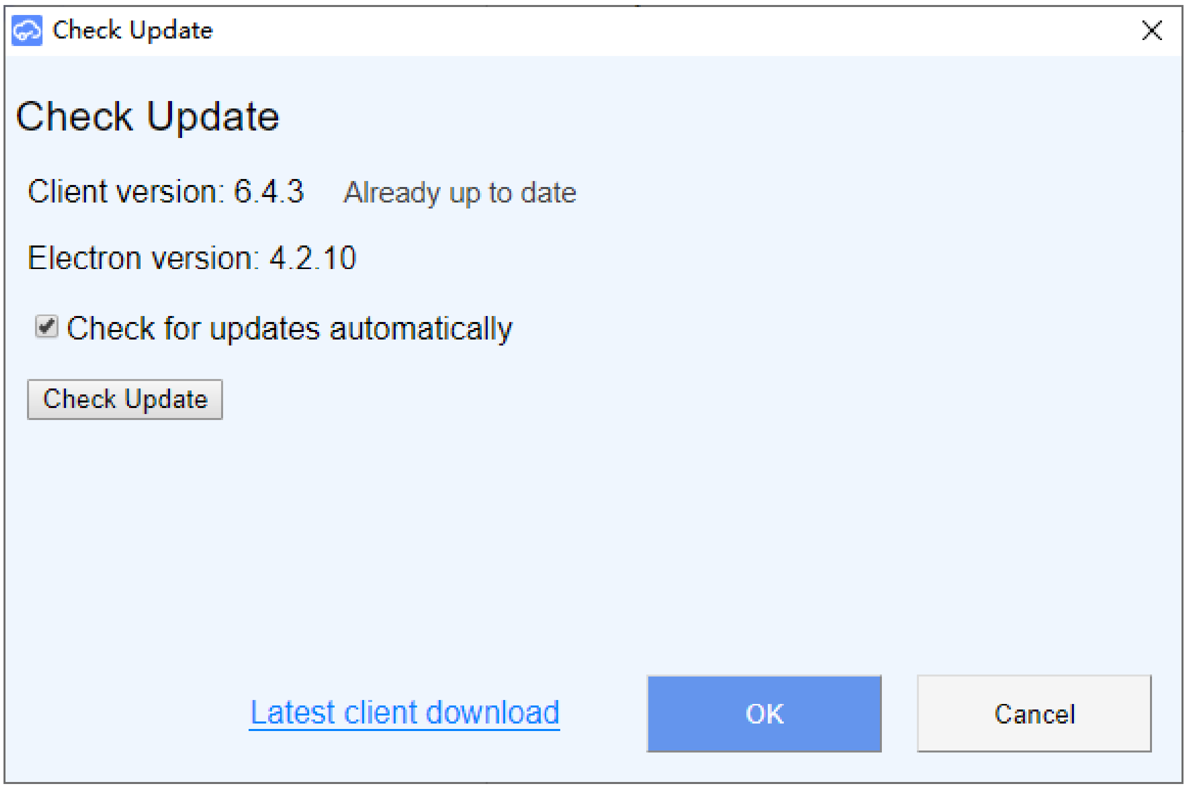


› 文档 › easyeda-data ›

名称	修改日期	类型	大小
components	2020/3/9 17:22	文件夹	
projects	2020/3/9 19:20	文件夹	
projects_backup	2020/8/3 19:15	文件夹	

```
-   **Version Setting**: Modify the running mode you need.

-   **Check Update**: Check the client version.
```


Check Update

Check Update

Client version: 6.4.3 Already up to date

Electron version: 4.2.10

[✓] Check for updates automatically

How to import online project into project offline version in batch?

1, first download project backup to local: backup project

2, after downloading and then decompression, to get the projects separate compression files, each compression file separately decompression in a folder.

3, copy the unzipped project folder to the offline project save directory.

4, then open the client, the client will automatically scan the newly added directory to generate a list of projects.

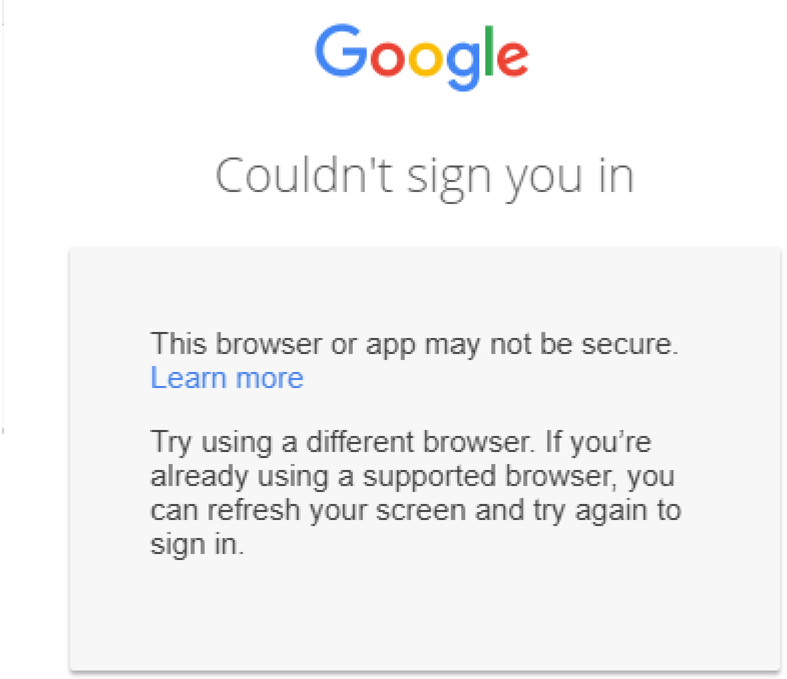
Note:

- *Please do not directly modify the name of the document in the folder in explorer, and do not directly copy and paste the new document in the folder, otherwise the editor will not be able to properly recognize the newly added document. Please go through the editor " - File - EasyEDA File Source..." Proceed to add new documents into the project.*
- *The too old client version doesn't allow to use, the dialog will tell you the version is expired, please download the new version to install.*
- *Doesn't support to upgrade automatically yet, please download and install manually.*

Known Issue

- When client running mode is "Project offline", it doesn't support to open the public project at Explore, and can not open the cloud project too.
- If you delete the file or project on "Project offline", it will be deleted directly, and can't not be undone! You only can find it back at "project_backup" directory

- If you login with Google account, it will show the client is not secure, please refer at this post [Can't login via Google Accout]



Google

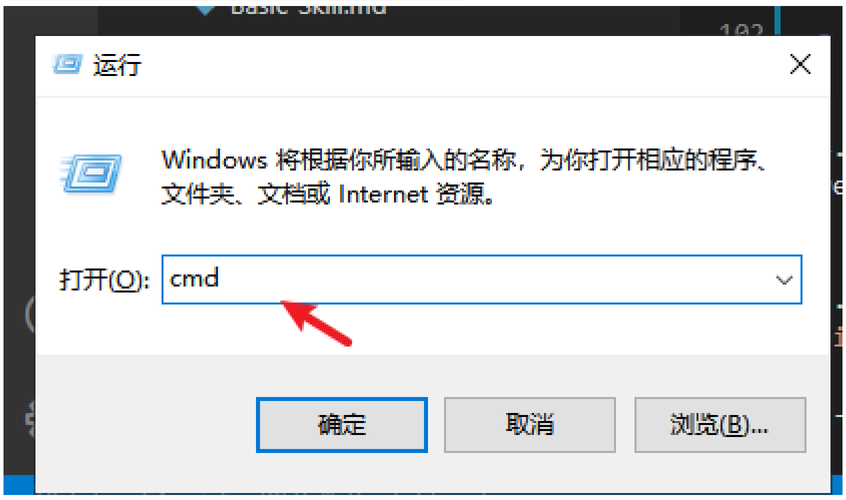
Couldn't sign you in

This browser or app may not be secure. Learn more

Try using a different browser. If you're already using a supported browser, you can refresh your screen and try again to sign in.

Please reset your password to get the password [Reset Password]

  - 1.Hit password reset link above
  - 2.enter gmail email address and hit reset, keep track of your new password
  - 3.now log in "normally", typing in your gmail address and password, not hitting the "login with Gmail" button.
  - 4.this issue is Google block other browsers, you can search this issue at Google
- Windows: Some windows systems can not run EasyEDA client well, or some PCs need some times to loading the login page,
if you met the dialog blank screen all the time when open the client, please try below steps:
  - Close client
  - 1.Open CMD window dialog by administrator
`WIN + R , input cmd, then enter.`

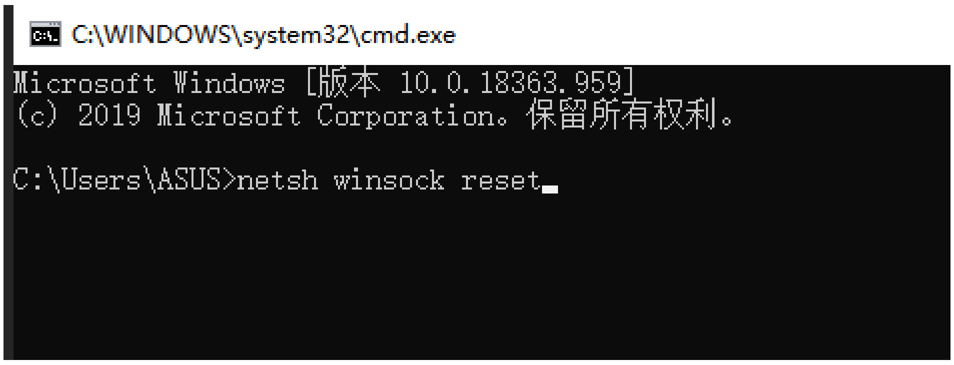


运行

Windows 将根据你所输入的名称，为你打开相应的程序、文件夹、文档或 Internet 资源。

打开(O): cmd

- 2.Input this at cmd window: `netsh winsock reset`

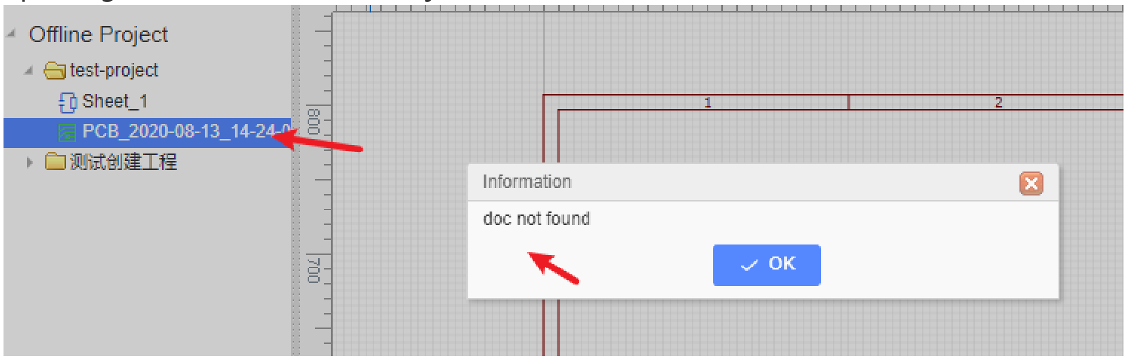


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18363.959]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\ASUS>netsh winsock reset_
```

- 3.Enter
- 4.Open client again. Maybe need to restart the computer.

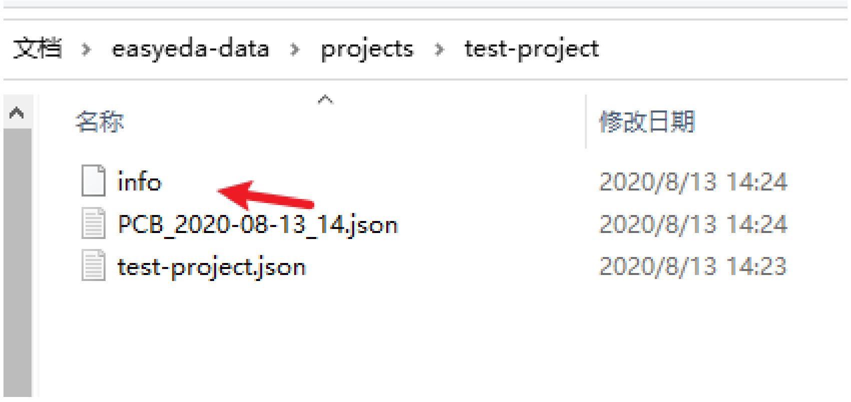
* Linux OS: Show segement fault while runing the client. That is system capability issue, any chance to upgrade the OS version.
* Mac OS: Can't install isse: How to open apps from unidentified developers on macOS Sierra
* If you open the local offline project document show "document not found" after updating the client version on Project Offline mode,



Information

doc not found

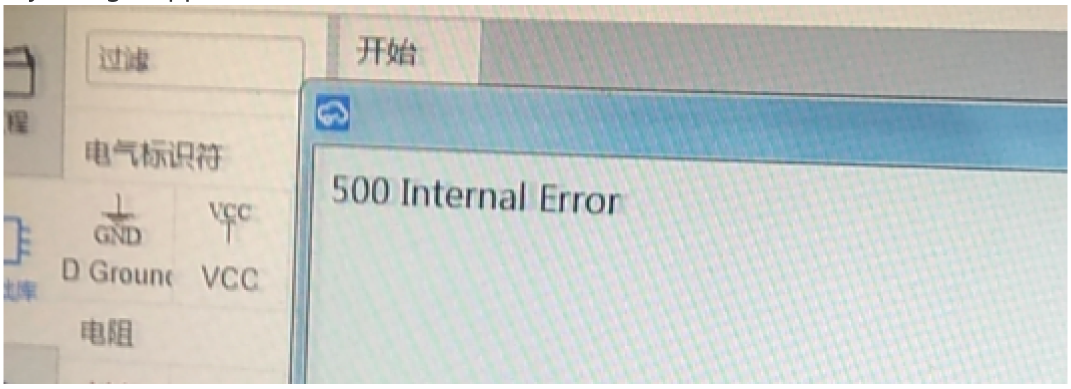
please right-click the project folder, open the project directory, and remove the file "info", then restart the client.



文档 › easyeda-data › projects › test-project

名称	修改日期
info	2020/8/13 14:24
PCB_2020-08-13_14.json	2020/8/13 14:24
test-project.json	2020/8/13 14:23

* If you login appear 500 error, as below:



500 Internal Error

Please use browser instead of desktop client, or try OS global network proxy, desktop client doesn't support network proxy.

How to Update

Version Rule

EasyEDA version number is `ReleaseCountsOfThisYear.MajorVersion.ReleaseCountsOfThisMajorVersion`. For example, v4.9.3 is the fourth year released of EasyEDA, and nine major versions are released in this year, EasyEDA had released 3 times in this major version.

Version Upgrade

If you use EasyEDA online, it can seamlessly upgrade by itself. However,EasyEDA uses an App Cache technique to allow you to use EasyEDA offline (W3C HTML5 Offline Web Applications) which may delay the automatic upgrading process. Therefore, if you want to upgrade to the latest version immediately, you can follow the two simple steps below.

1. Check the About... dialog;
2. If the Built Date is older than 2017/06/01:
   Close your browser open EasyEDA again.
   If the Built Date is still showing older than 2017/06/01:
   Close your browser and open EasyEDA again.
   If the Built Date is at or newer than 2017/06/01, you don't need to do anything.

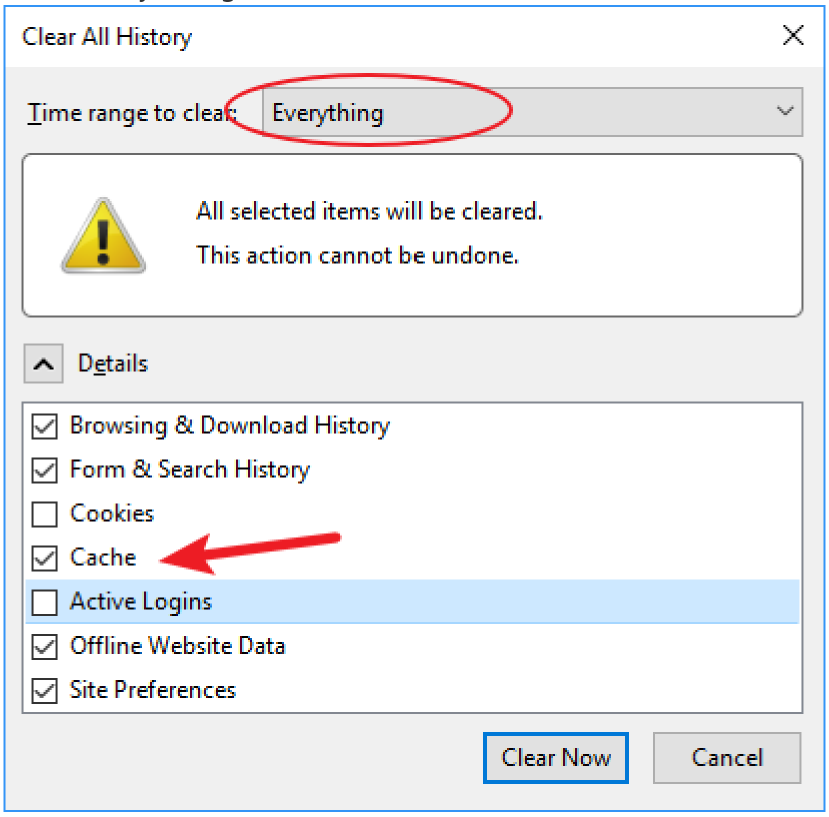
Note: **2017/06/01** *is just an example.*

If those two steps don't work, you may need to clear your browser's cache:

Mozilla Firefox

- Close the editor, Go to “Preferences... > Privacy & Security > History > clear your recent history” or use **Ctrl+shift+Delete**,
- Click on “Clear now” ,

- Reload easyeda again.



Clear All History

Time range to clear: Everything

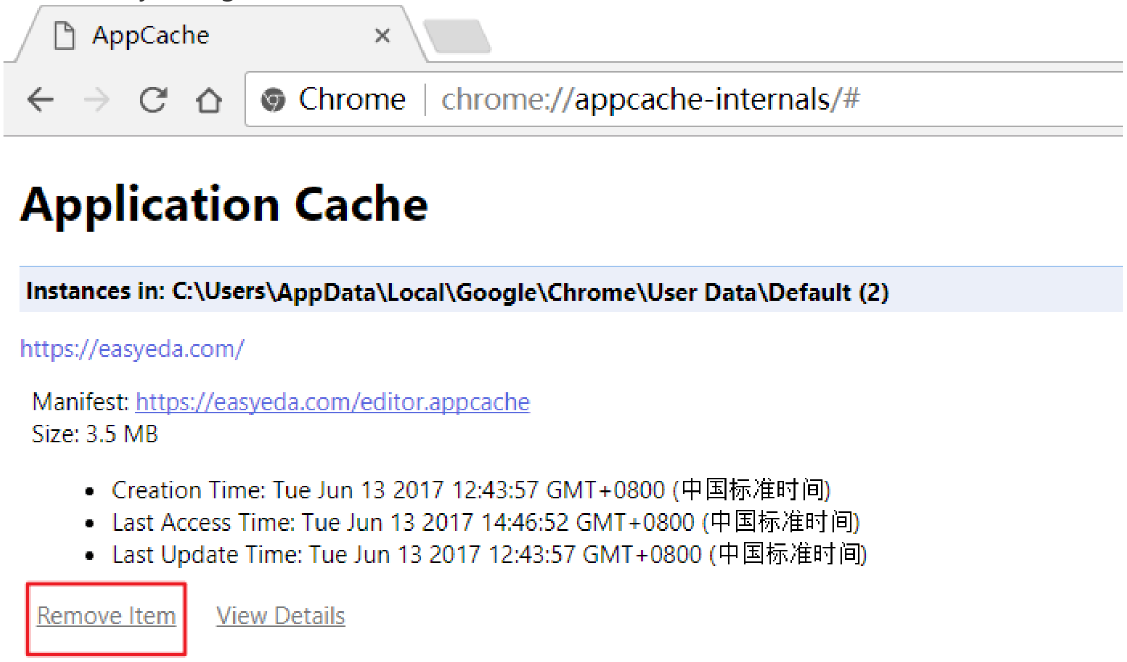
All selected items will be cleared.
This action cannot be undone.

Details

- [✓] Browsing & Download History
- [✓] Form & Search History
- [ ] Cookies
- [✓] Cache
- [ ] Active Logins
- [✓] Offline Website Data
- [✓] Site Preferences

Chrome

- Close the editor, Open the following URL: `chrome://appcache-internals/`.
- Look for easyeda.com and click "Remove" .
- Reload easyeda again.



Application Cache

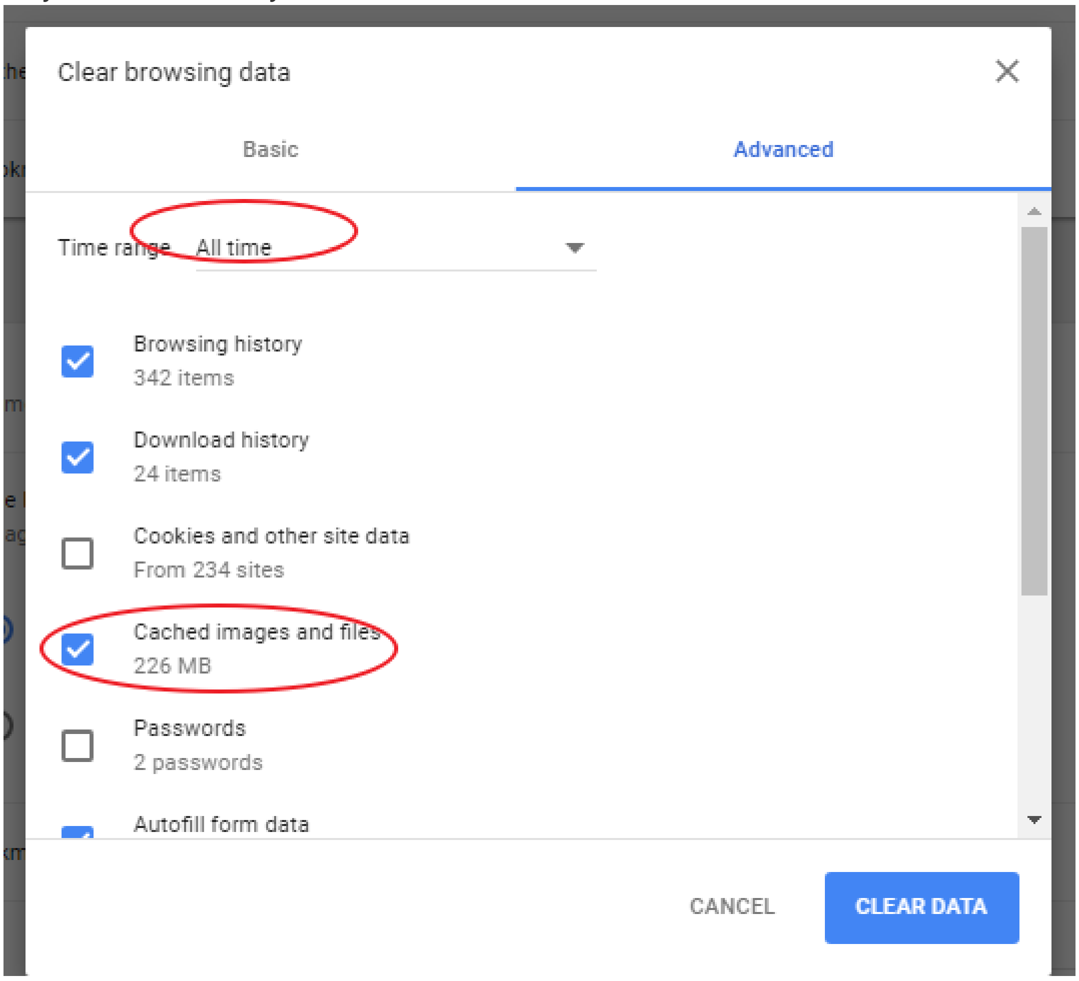
Instances in: C:\Users\AppData\Local\Google\Chrome\User Data\Default (2)

https://easyeda.com/

Manifest: https://easyeda.com/editor.appcache
Size: 3.5 MB

- Creation Time: Tue Jun 13 2017 12:43:57 GMT+0800 (中国标准时间)
- Last Access Time: Tue Jun 13 2017 14:46:52 GMT+0800 (中国标准时间)
- Last Update Time: Tue Jun 13 2017 12:43:57 GMT+0800 (中国标准时间)

- Or you can use hotkey **Ctrl+shift+Delete** to delete Chrome caches.


Clear browsing data

Basic | Advanced

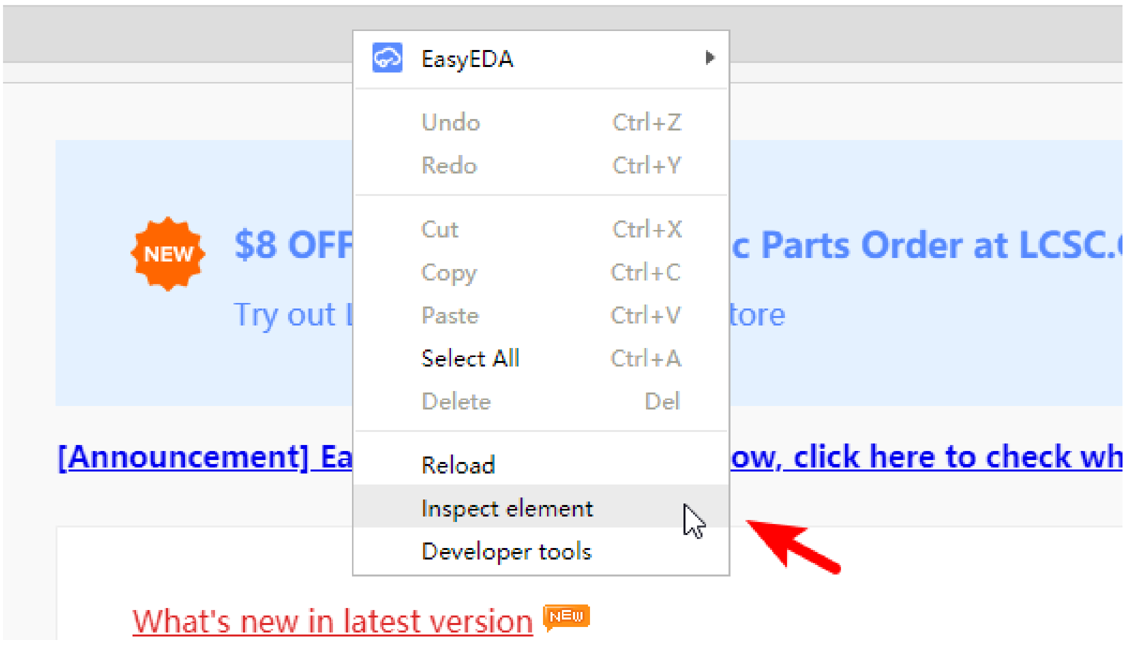
Time range: All time

- [✓] Browsing history — 342 items
- [✓] Download history — 24 items
- [ ] Cookies and other site data — From 234 sites
- [✓] Cached images and files — 226 MB
- [ ] Passwords — 2 passwords
- [✓] Autofill form data

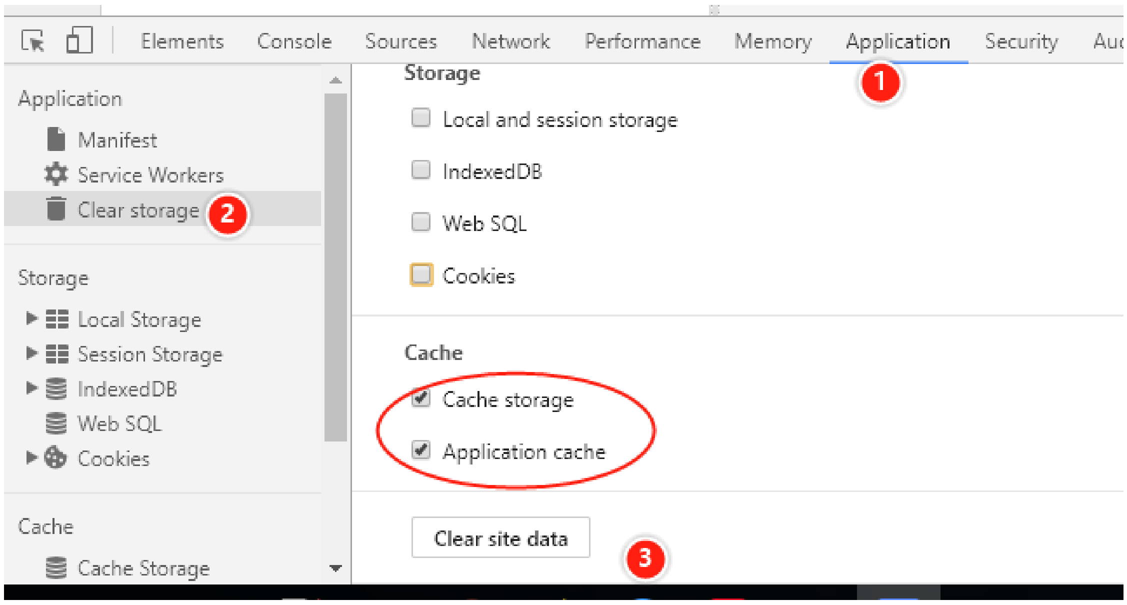
CANCEL | CLEAR DATA

Desktop Client

- Close client and re-open it.
- If doesn't work, rght-clik start page, use "Inspect Electment".



- Switch to "Application" - "Clear storage", enable "Cache storage" and "Application cache", then click "Clear site data".



Storage

- [ ] Local and session storage
- [ ] IndexedDB
- [ ] Web SQL
- [ ] Cookies

Cache

- [✓] Cache storage
- [✓] Application cache

- Restart Client.

User Center FAQ

How to change password

Via: User Center > Account > Password Setting

How to recover the deleted file

Via: User Center > Recycle Bin

Find the file you want and then recover it.

How to transfer project or library to the team

Transfer project: Enter the project, via: Setting > Advance Setting > Transfer Project

Transfer library: Move the mouse to the library, and then click the transfer icon.

How to delete project

Via Project > Manage > Setting > Advanced > Delete Project

Contact Us

Contact

PCB Order Problems:

- support@jlcpcb.com

- At present, the EasyEDA PCB fabrication and PCB Assembly service is transferred to JLCPCB. Although EasyEDA is part of the same company group, for any PCB order problems please contact with JLCPCB using the email address above.

Parts Order Problems:

- support@lcsc.com
- EasyEDA provides direct links to thousands of components which can be ordered directly from LCSC. For any problems with LCSC parts orders, please contact LCSC using the email address above.

All Other Inquiries About EasyEDA:

- Tutorials: EasyEDA tutorial
- User forum: EasyEDA forum
- If you find a problem whilst using EasyEDA please first post to the forum to ask for help. If the issue cannot be resolved there, please download and attach the EasyEDA source file of your Schematic, PCB or Project together with a clear, step-by-step description of how to repeat the issue to:

  support@easyeda.com

Notice

EasyEDA team may not have the time or resources to help you fix all your problems; we may just be able to help you to fix problems commonly encountered by newbies, such as using a drawing polyline in place of a wire, finding a spice model for a simulation or selecting the right PCB footprint.

- Please note that although some browsers or plug-ins allow you to use gestures, EasyEDA does not work with gestures, so you should disable this function.
- Simultaneous editing is not yet fully supported: care must be taken because the last save by any collaborator overwrites all previous saves.
- When signing up for an account with EasyEDA, please take a few moments to think about your username because this is the name that other users will see on your designs and posts if you choose to share them or make them public. Once you have created an account, you cannot change your username.
- You can use upper and lower case letters, numbers and symbols to make a strong password but please note that the password entry is case sensitive.
- PCBs ordered directly from an EasyEDA project are passed on to and fullfilled by, JLCPCB.

Business Development/Cooperation About EasyEDA:

please contact

dillon@easyeda.com

Address:

- F5, Tianjian Building, No.7 Shangbao Road, Futian District, Shenzhen, Guangdong, 518000, China

Introduction

Introduction to EasyEDA

Welcome to EasyEDA, a great web based EDA(Electronics Design Automation) tool for electronics engineers, educators, students, makers and enthusiasts.

There is no need to install any software. Just open EasyEDA in any HTML5 capable, standards compliant web browser.

Whether you are using Linux, Mac or Windows, it is highly recommended to use Chrome or Firefox as your browser. You can also download EasyEDA client.
EasyEDA has all the features you expect and need to take your design rapidly and easily from conception through to production.

EasyEDA Editor:

https://easyeda.com/editor

Instruction:

- This tutorial document will be updated as changes are made to the EasyEDA editor.

Tutorial for PDF

EasyEDA-Tutorials.pdf

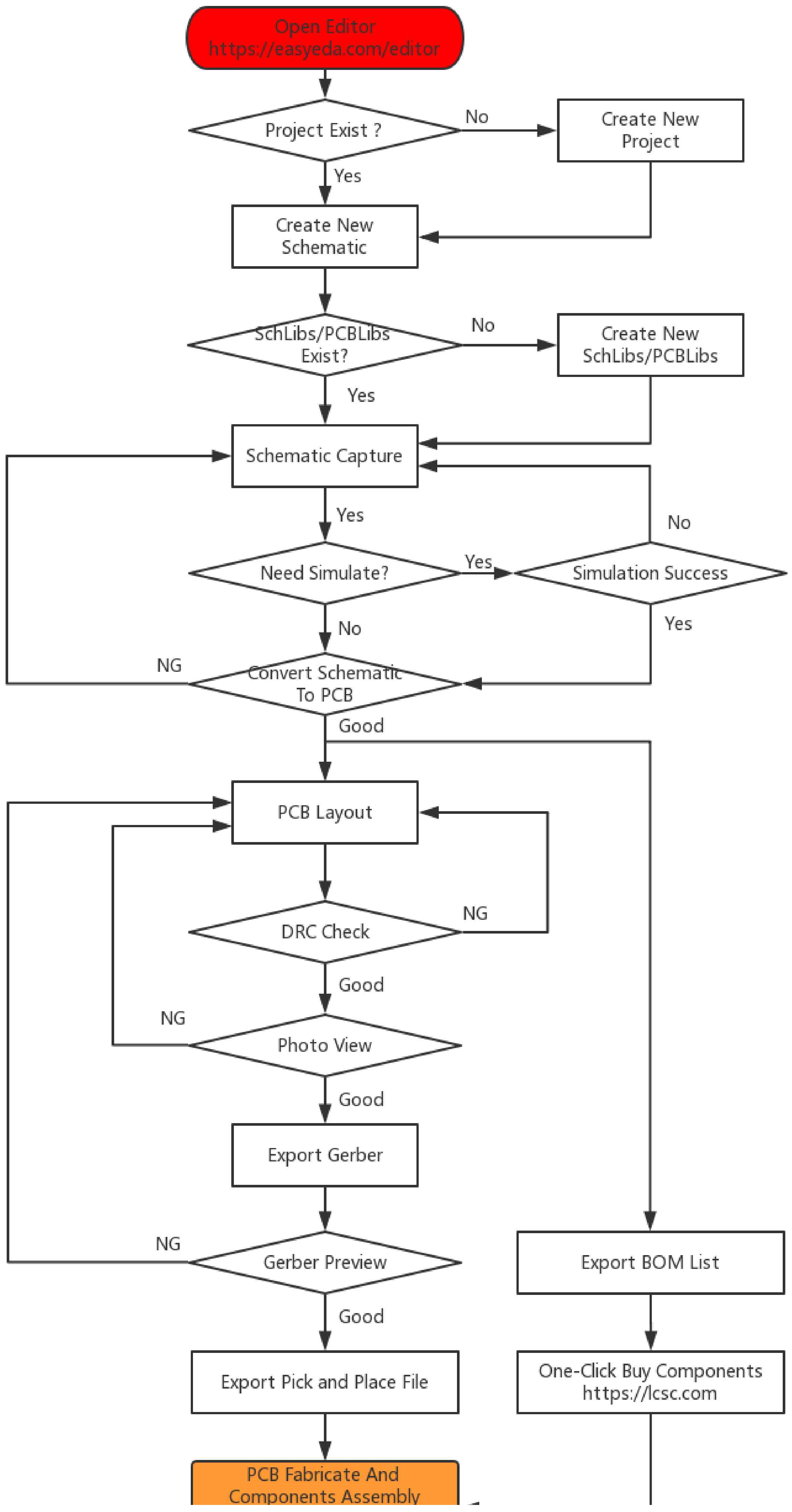
EasyEDA Provides:

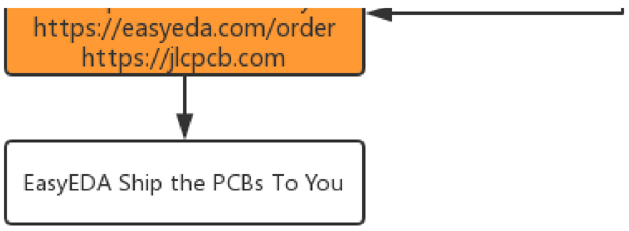
- Simple, Easy, Friendly, and Powerful drawing capabilities
- Works Anywhere, Anytime, on Any Device
- Real-time Team Cooperation
- Sharing Online
- Thousands of open-source projects
- Integrated PCB fabrication and Components purchase workflow
- API provided
- Script support
- Schematic Capture
    - LTSpice-based Simulation
    - Spice models and subcircuits create
    - Waveform viewer and data export(CSV)
    - Netlist export(Spice, Protel/Altium Designer, Pads, FreePCB)
    - Document export(PDF, PNG, SVG)
    - EasyEDA source file export(json)
    - Altium Designer format export
    - BOM export
    - Multi-sheet schematics

    - Schematic module
    - Theme setting
    - Document recovery
- PCB Layout
    - Design Rules Checking(DRC)
    - Multi-Layer, 6 copper layer supported
    - Document export(PDF, PNG, SVG)
    - EasyEDA source file export(json)
    - Altium Designer format export
    - BOM export
    - DXF export
    - Photo view
    - 3D View
    - Generate fabrication file(Gerber)
    - Export Pick and Place file
    - Auto Router
    - PCB module
    - Document recovery
- Import
    - Altium/ProtelDXP ASCII Schematic/PCB
    - Eagle Schematic/PCB/Libraries
    - KiCAD Schematic/PCB/Libraries
    - DXF
- Libraries
    - More than 1,000,000 public Libraries(Symbol and Footprint)
    - Library management
    - Symbol/Subpart create and edit
    - Spice symbol/model create and edit
    - Libraries management
    - Footprint create and edit

Design Flow by Using EasyEDA

You can design circuits easily using EasyEDA. The design flow is as shown here:


Open Editor
https://easyeda.com/editor
Project Exist ?
No
Create New Project
Yes
Create New Schematic
SchLibs/PCBLibs Exist?
No
Create New SchLibs/PCBLibs
Yes
Schematic Capture
Yes
No
Need Simulate?
Yes
Simulation Success
No
Yes
NG
Convert Schematic To PCB
Good
PCB Layout
DRC Check
NG
Good
NG
Photo View
Good
Export Gerber
NG
Gerber Preview
Good
Export BOM List
Export Pick and Place File
One-Click Buy Components
https://lcsc.com
PCB Fabricate And Components Assembly

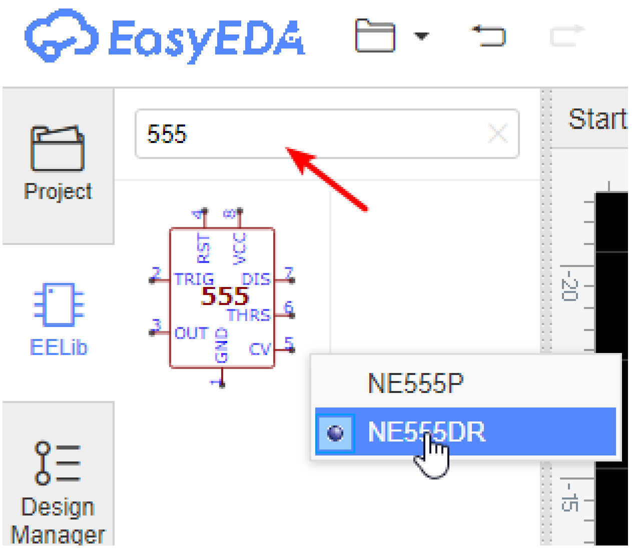


UI Introduction

EasyEDA Editor has a clear and friendly user interface. It has a short learning curve, and you will be productive in a short time.

Filter

To use the filter first select what module you need in the left navigation panel, and then you can find projects, files, parts, and footprints quickly and easily just by typing a few letters. For example, if you want to find all files containing "NE555" in the title, just type "555". The filter is non-case sensitive.



The Filter can only find projects, files, part titles, and names. It does not search the Descriptions and Content fields.

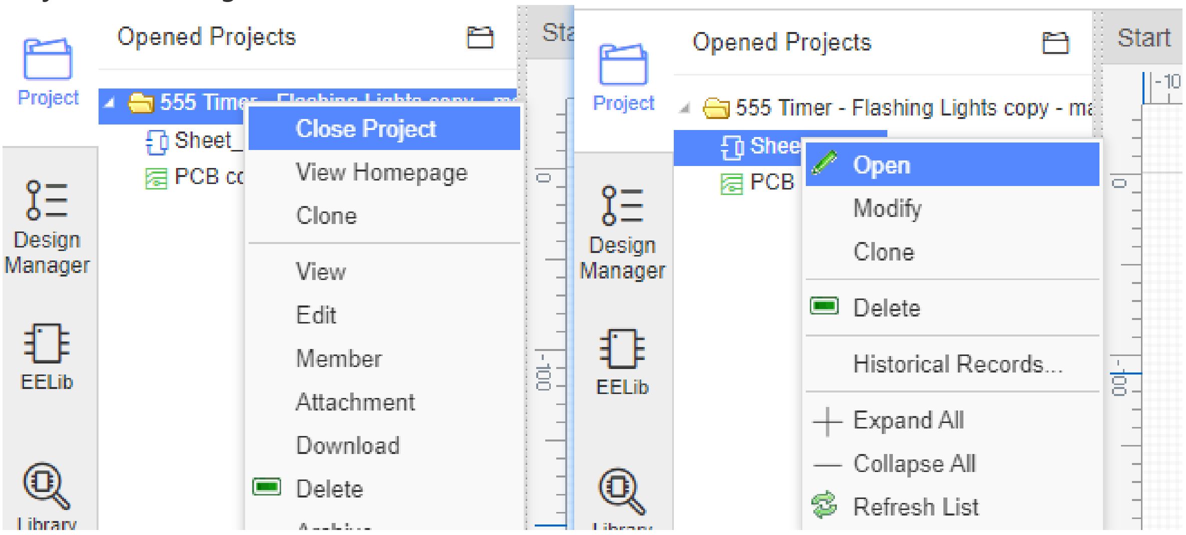
Click the X to clear the filter.

Navigation Panel

The Navigation panel is especially important for EasyEDA: This is where you can find all of your projects, files, parts and footprints.

Project

Here you can find all your projects, both private or shared with the public, or fork them from someone else's. These options have a context menu found by right clicking and selecting Projects. You will get a menu tree like:

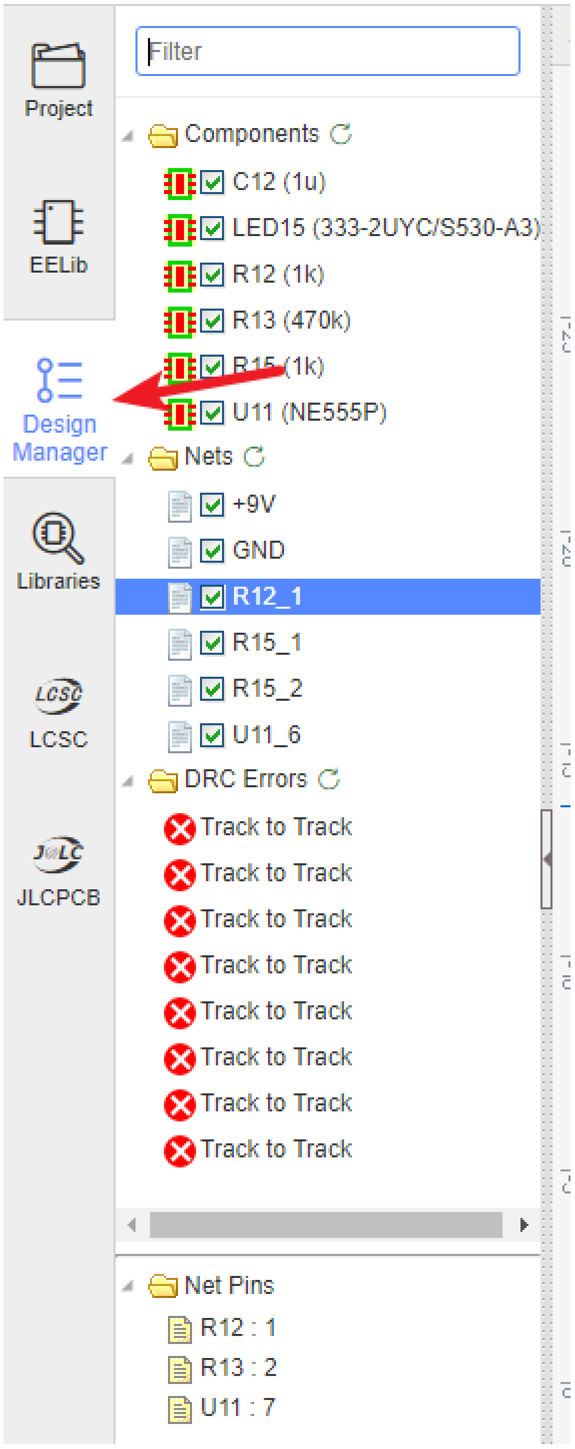


EELib

EElib means EasyEDA Libraries, it provides lots of components complete with simulation models, many of which have been developed for EasyEDA to make your simulation experience easier.

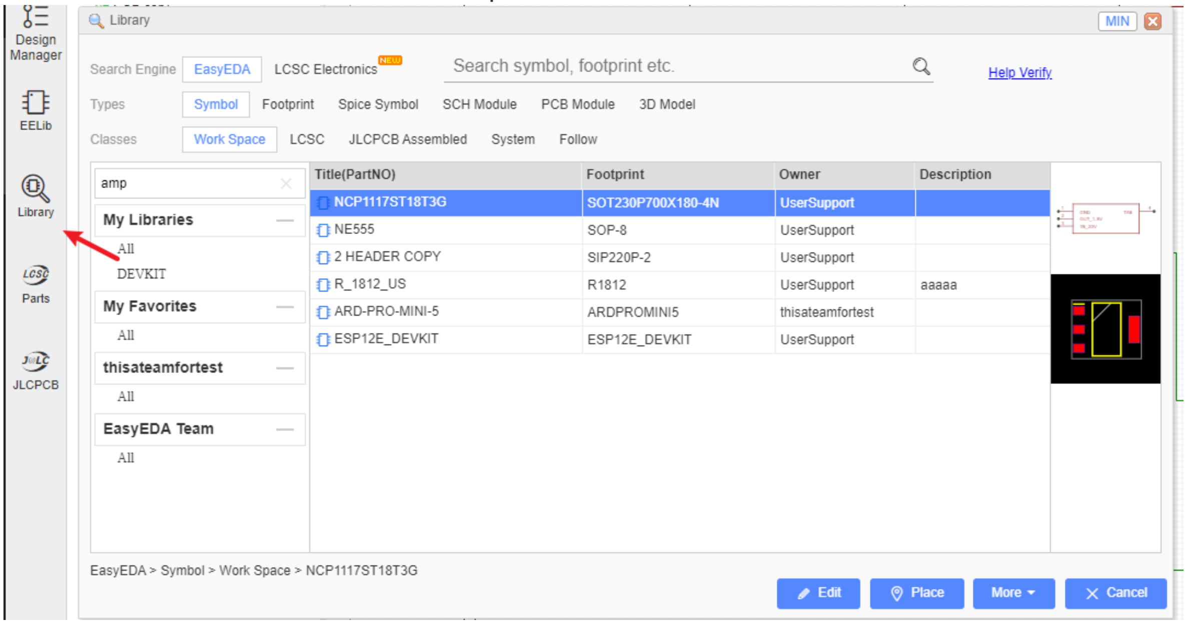
Design Manager

Design Manager, you can check each component and net wiring, and it provides DRC(Design rule check) to help improve your design.



Library

Contains schematic symbols and PCB footprints for many available components and projects. Your own libs and modules will show up here.



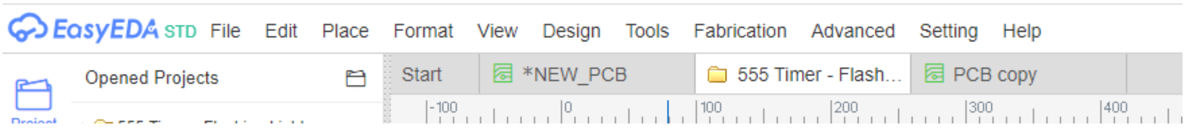
Title(PartNO)	Footprint	Owner	Description
NCP1117ST18T3G	SOT230P700X180-4N	UserSupport	
NE555	SOP-8	UserSupport	
2 HEADER COPY	SIP220P-2	UserSupport	
R_1812_US	R1812	UserSupport	aaaaa
ARD-PRO-MINI-5	ARDPROMINI5	thisateamfortest	
ESP12E_DEVKIT	ESP12E_DEVKIT	UserSupport	

EasyEDA > Symbol > Work Space > NCP1117ST18T3G

- **LCSC**
  If you want to buy components to finish your PCB, you should try the **LCSC** module. LCSC.com and EasyEDA are the same company.
  EasyEDA partners with China's largest electronic components online store by number of customers and product quantity shipped. https://lcsc.com.
  LCSC means **L**ove **C**omponents? **S**ave **C**ost! We suggest to our users to use LCSC parts to design. Why?
  - Low minimum required quantities & Global Shipping.
  - More Than 25,000 Kinds of Components.
  - All components are genuine high quality.
  - Ordering components is easy.
  - Savings can exceed 40%.
  - You can use LCSC component symbols and footprints directly in EasyEDA editor.
- **JLCPCB**
  JLCPCB.com, LCSC.com and EasyEDA are all part of the same company group.
  https://jlcpcb.com
  More than 200,000 customers worldwide trust JLC, 8000 + online orders per day, JLCPCB (Shenzhen JIALICHUANG Electronic Technology Development Co.,Ltd.), is the largest PCB prototype enterprise in China and a high-tech manufacturer specializing in quick PCB prototype and small-batch production. Affordable, high quality boards ae fully manufactured in China. Boards are fully e-tested. Pricing is clear and easy to understand.

Top Menu

Most EasyEDA features can be found on the top menu:



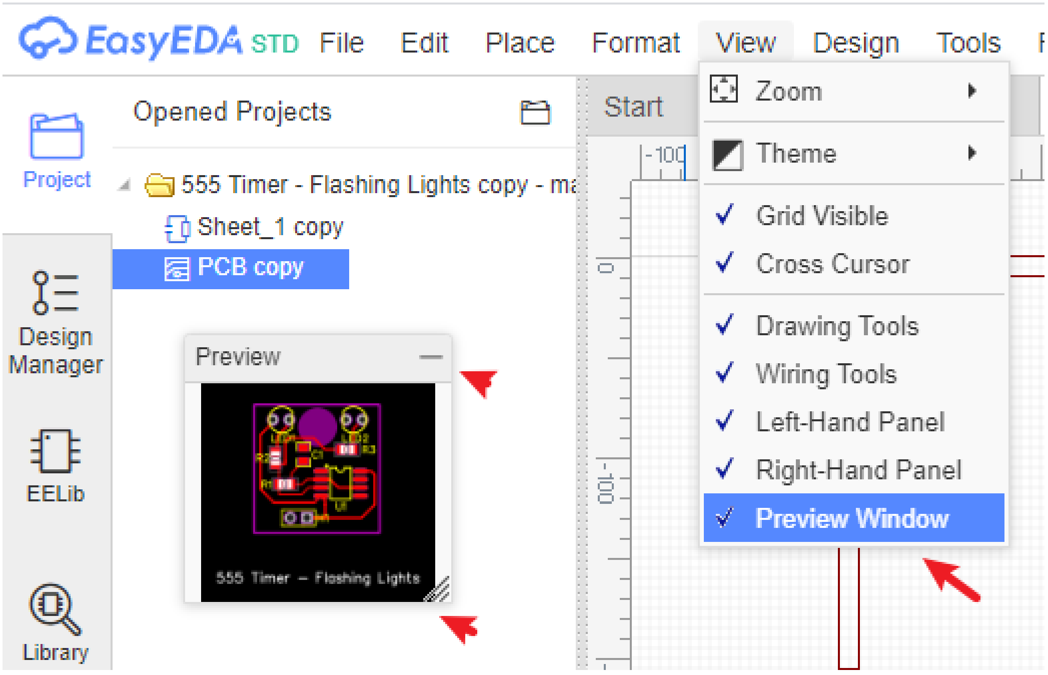
You can find what you need easily and clearly.

Preview Dialog

The Preview dialog will help you choose components and footprints and can help you to identify schematics and PCB layouts.

You can close or open this dialog with:

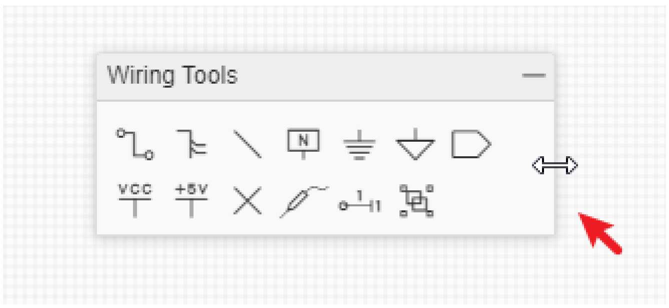
Top Menu > View > Preview Window.



- The Preview Dialog has a resizing handle in the bottom right corner.
- The Preview Dialog can't be closed but double clicking on the top banner will roll up the panel or you can click the top right corner – . Double clicking the top banner again toggles it back to the selected size.

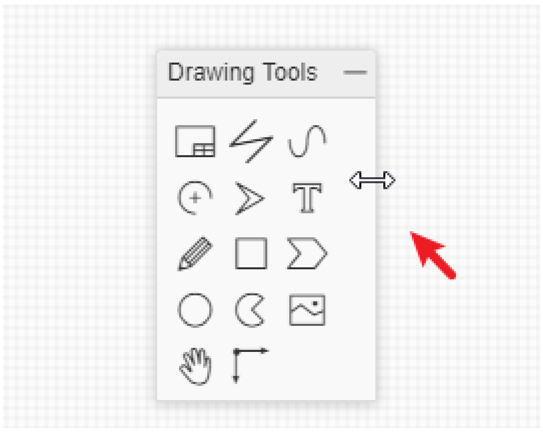
Wiring Tools

The Wiring Tools are document type sensitive: different document types have different tools.



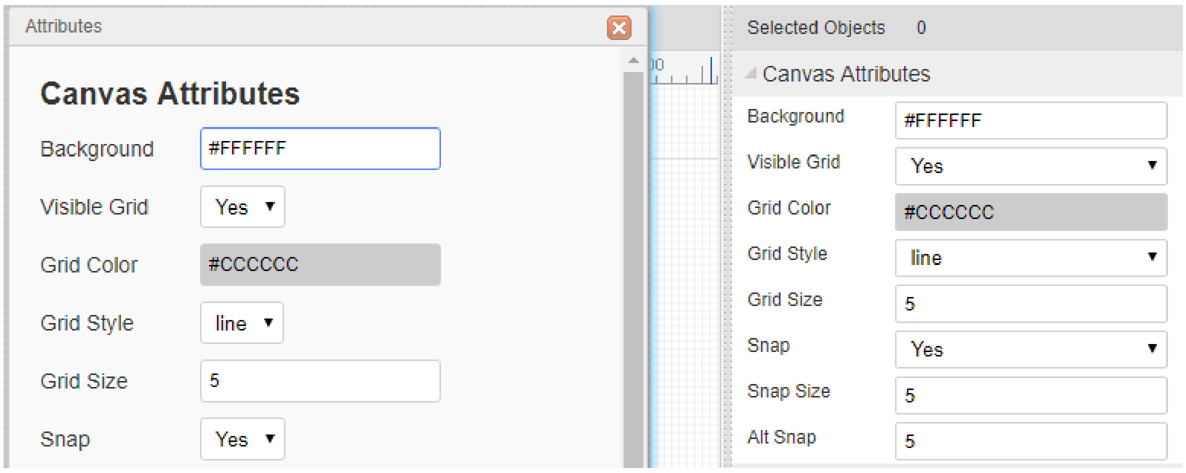
Drawing Tools

To keep EasyEDA's UI clean and sharp, the Wiring and Drawing tools palettes can be resized horizontally, rolled up or hidden so if you want to focus on drawing or have a smaller monitor, you can roll up or hide them to free up more monitor space and reduce the clutter.



Canvas Attributes

You can find the canvas Properties setting by clicking in any blank space of the canvas.



Attributes

Canvas Attributes

Background	#FFFFFF
Visible Grid	Yes
Grid Color	#CCCCCC
Grid Style	line
Grid Size	5
Snap	Yes

Selected Objects 0

Canvas Attributes

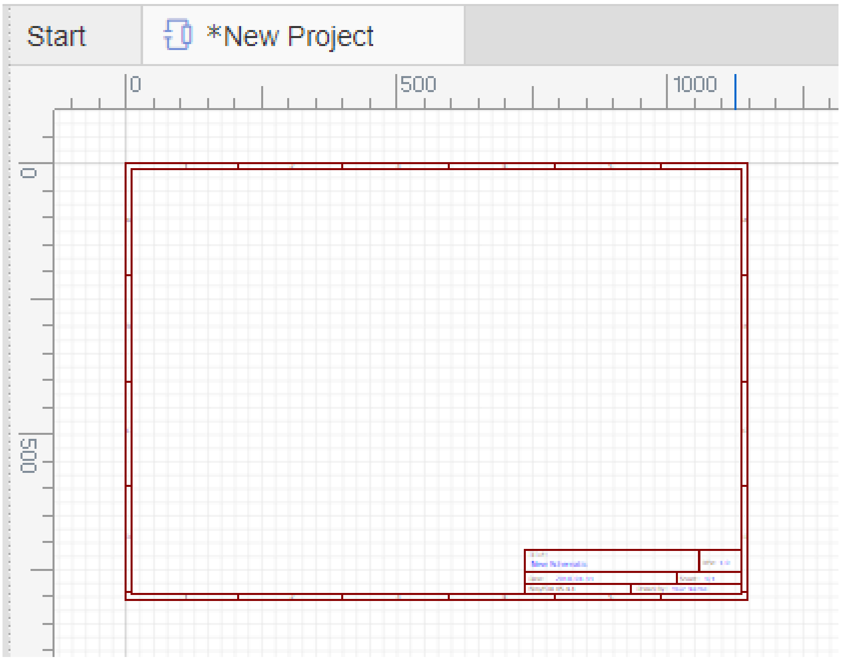
Background	#FFFFFF
Visible Grid	Yes
Grid Color	#CCCCCC
Grid Style	line
Grid Size	5
Snap	Yes
Snap Size	5
Alt Snap	5

The background and grid colors and the style, size, visibility and snap attributes of the grid can all be configured.

The canvas area can be set directly by the Width and Height or from the available preset frame sizes.

Canvas

This is where it all happens! This the area where you create and edit your schematics, PCB layouts, symbols, footprints, and other drawings. You also run simulations and display Waveform traces from here.

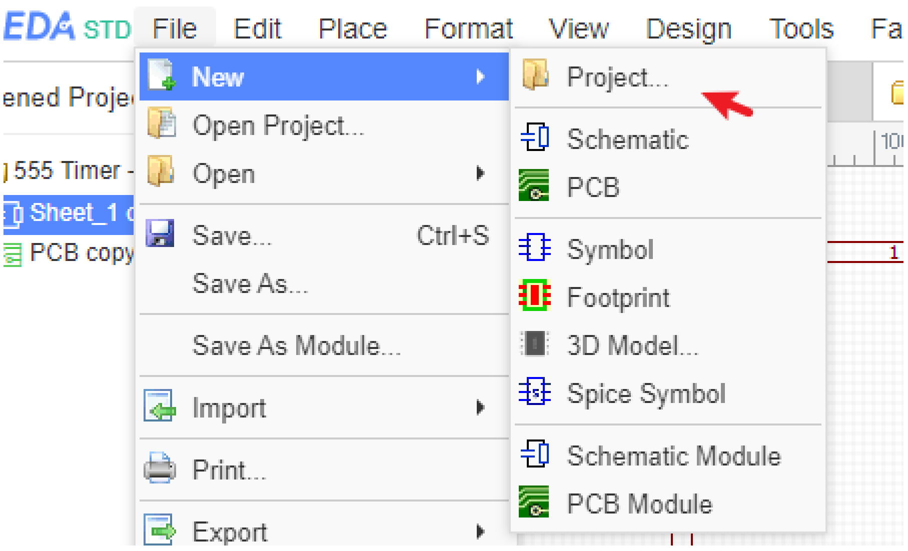


How to Create a New Project or File

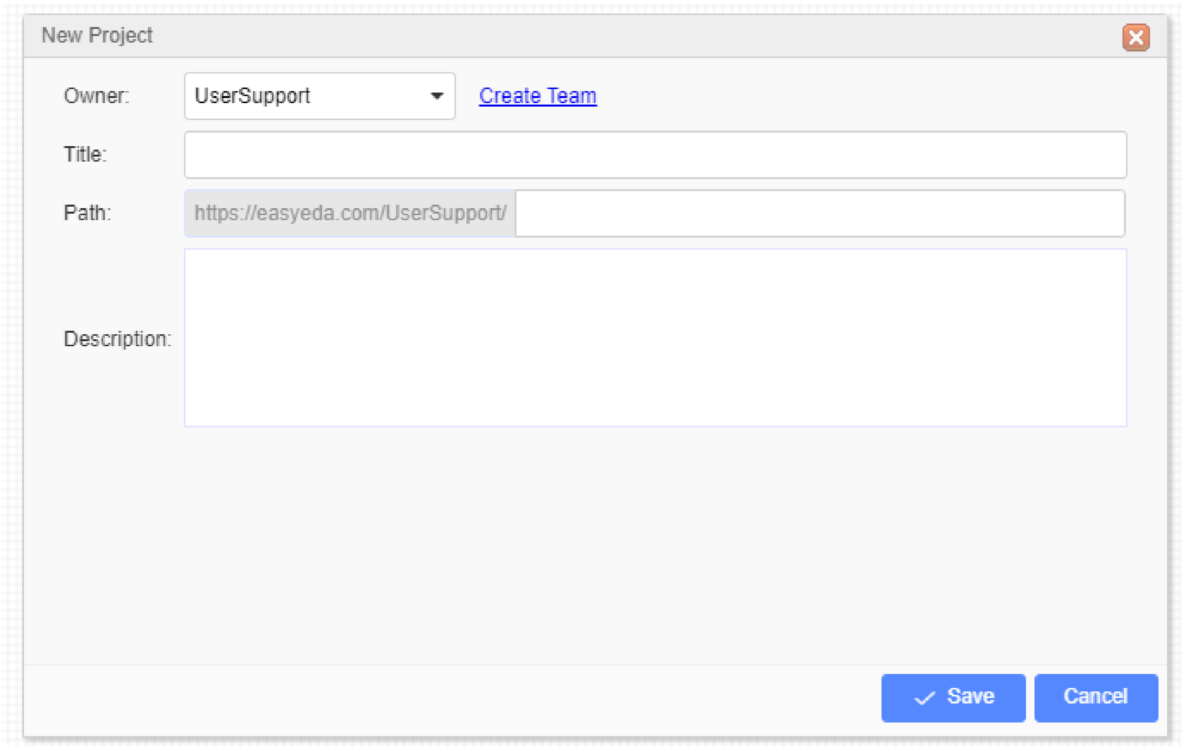
Create Project

After logging in, you can create a new project:

File > New > Create a new project/Schematic..etc



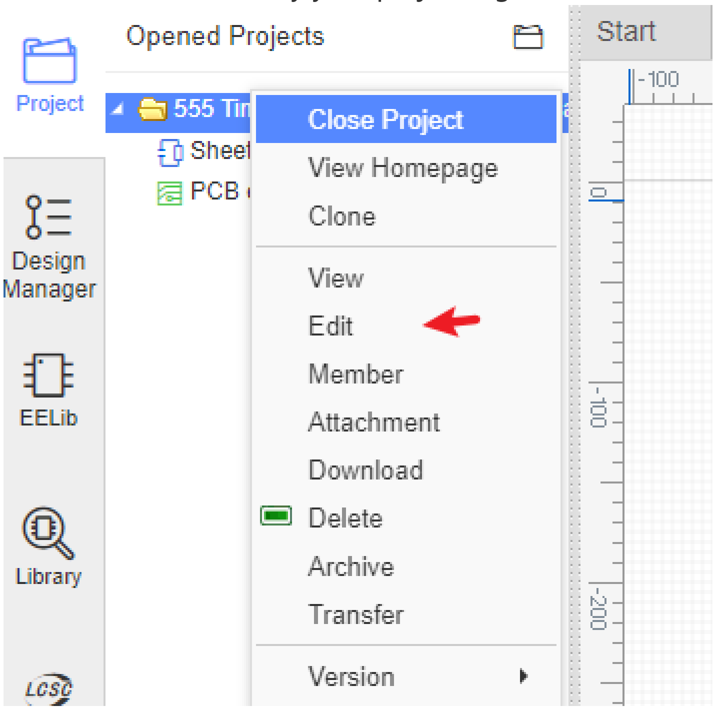
The Project concept is important in EasyEDA because it is the foundation of organizing your designs.


New Project

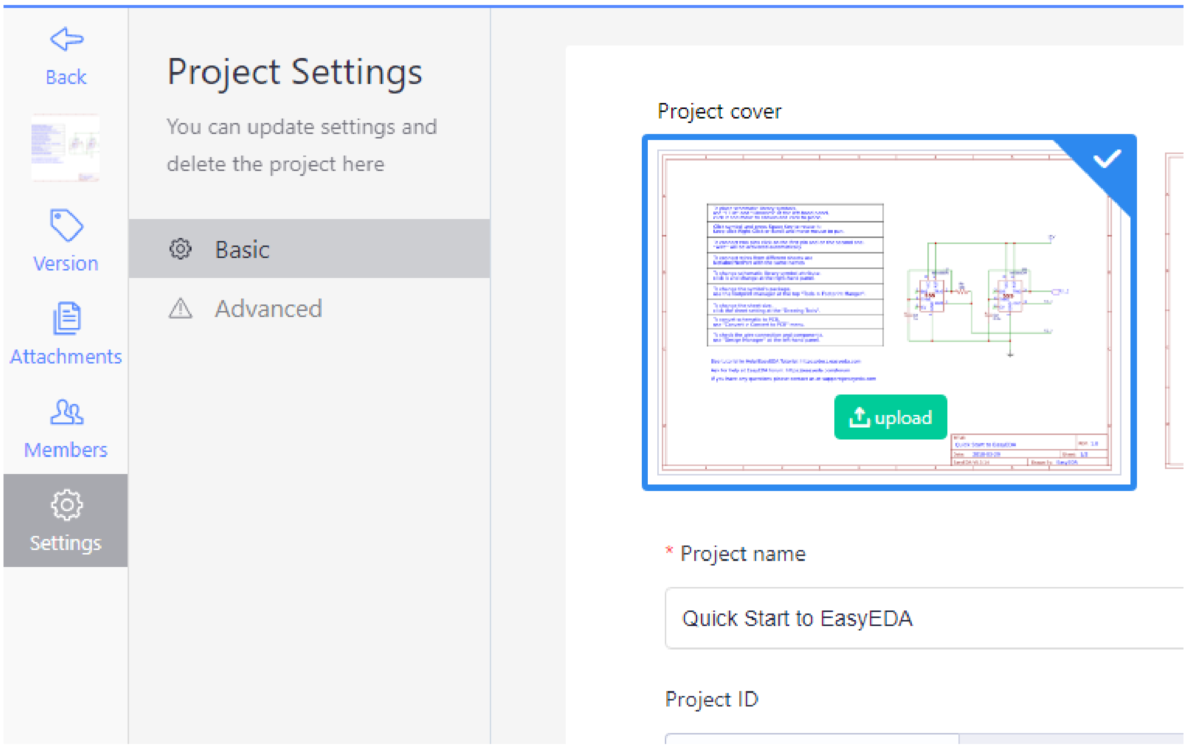
Owner:	UserSupport
Title:	
Path:	https://easyeda.com/UserSupport/
Description:	

- **Owner**: You can change the owner of this project; you can change the owner to the team that you have joined.
- **Title**: Give it a title: this will show in the project tree in the left-hand panel.
- **Path**: EasyEDA allows you to set the path for the project, this is useful for sharing with your friends. It cannot be edited once it has been created.
- **Description**: Adding a short description helps you and anyone you are sharing this project with understand what the project is about.

Once created, to modify your project, right click on it in the project tree in the left-hand panel:



this will open a web page where you can edit your project:

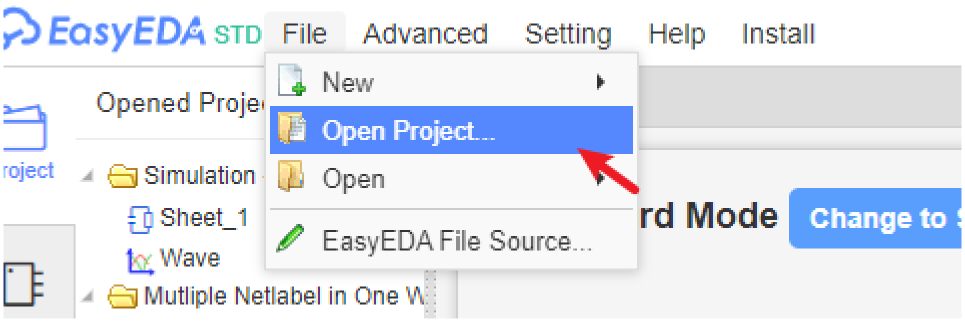


From here, you can change to publish or not, allow other people to comment on your project, or type a more detailed description of the project content. To help you make your project stand out or to maybe simply make a detailed description of your project easier to read, you can use Markdown syntax.

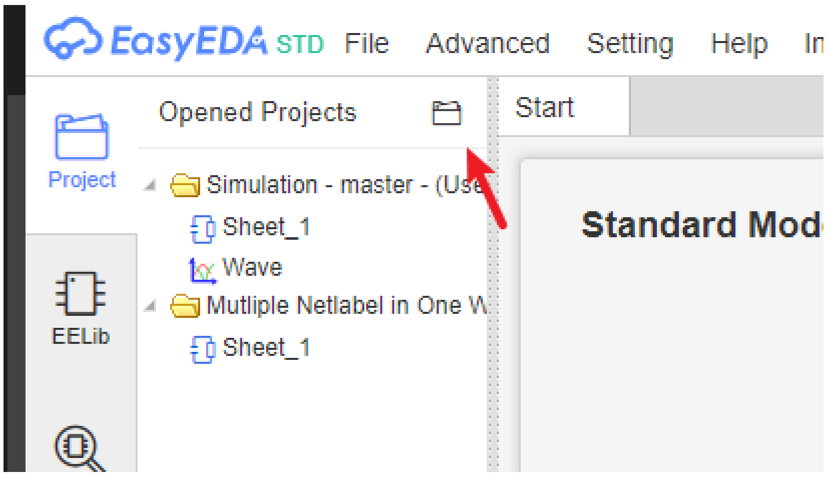
Open Project

You can open your created project using:

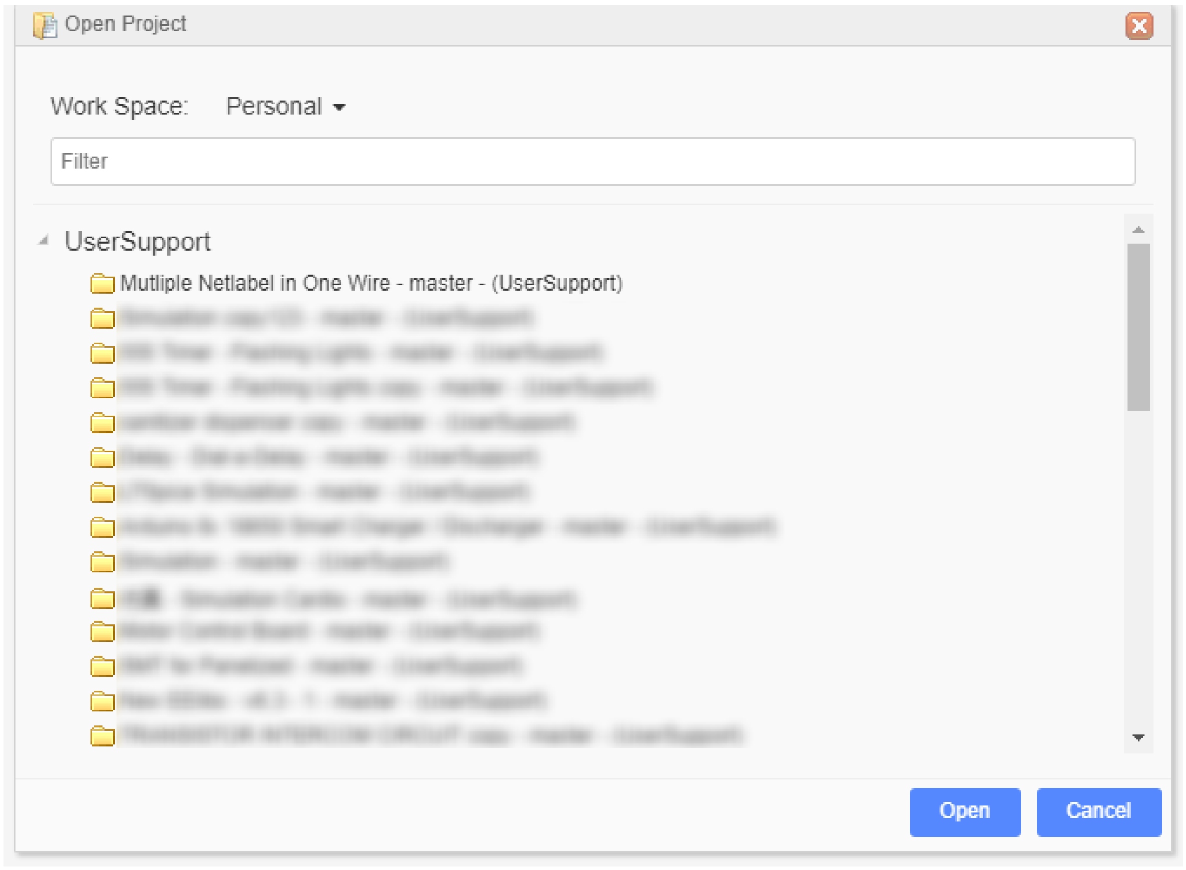
Top Menu - File - Open Project



Or click the Opened Project "open project" icon.

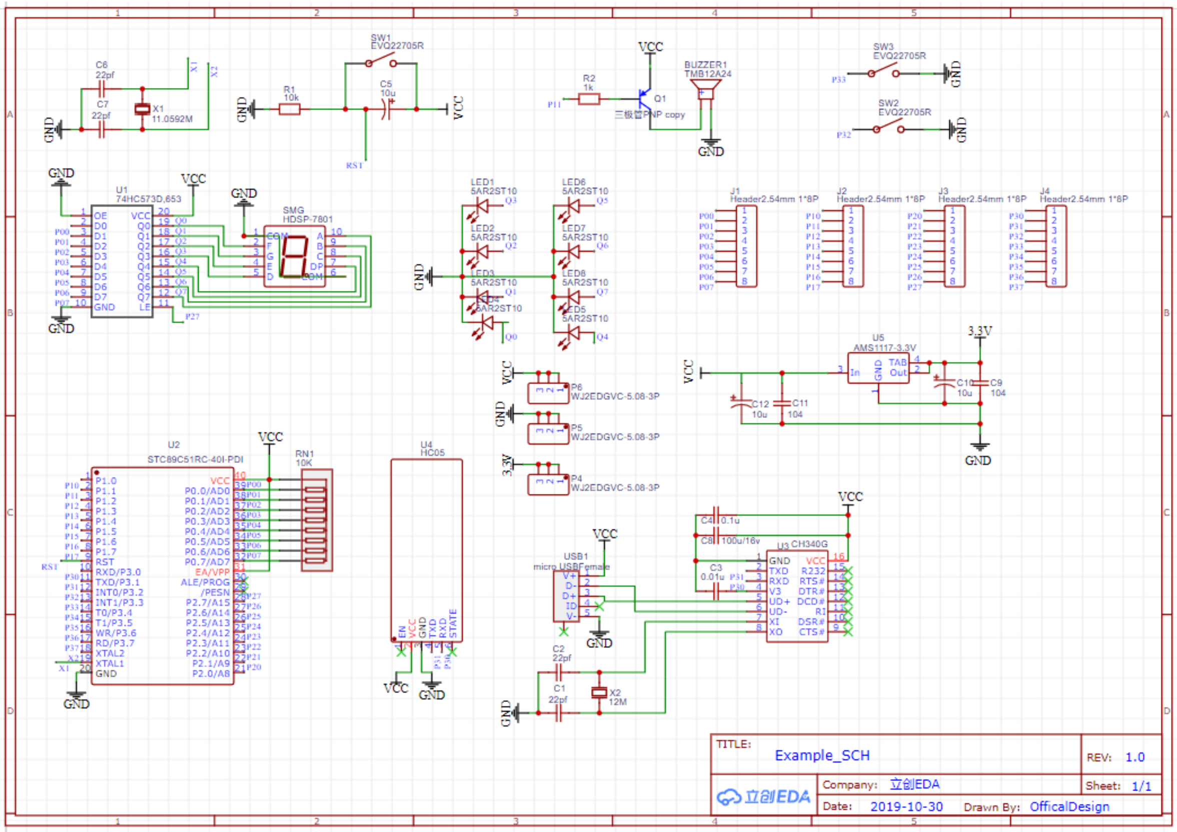


Select the project and open it.



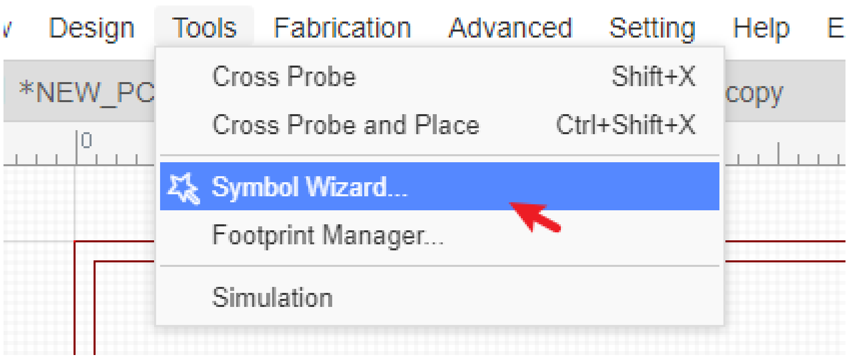
Schematic Capture

EasyEDA can create professional quality schematics.



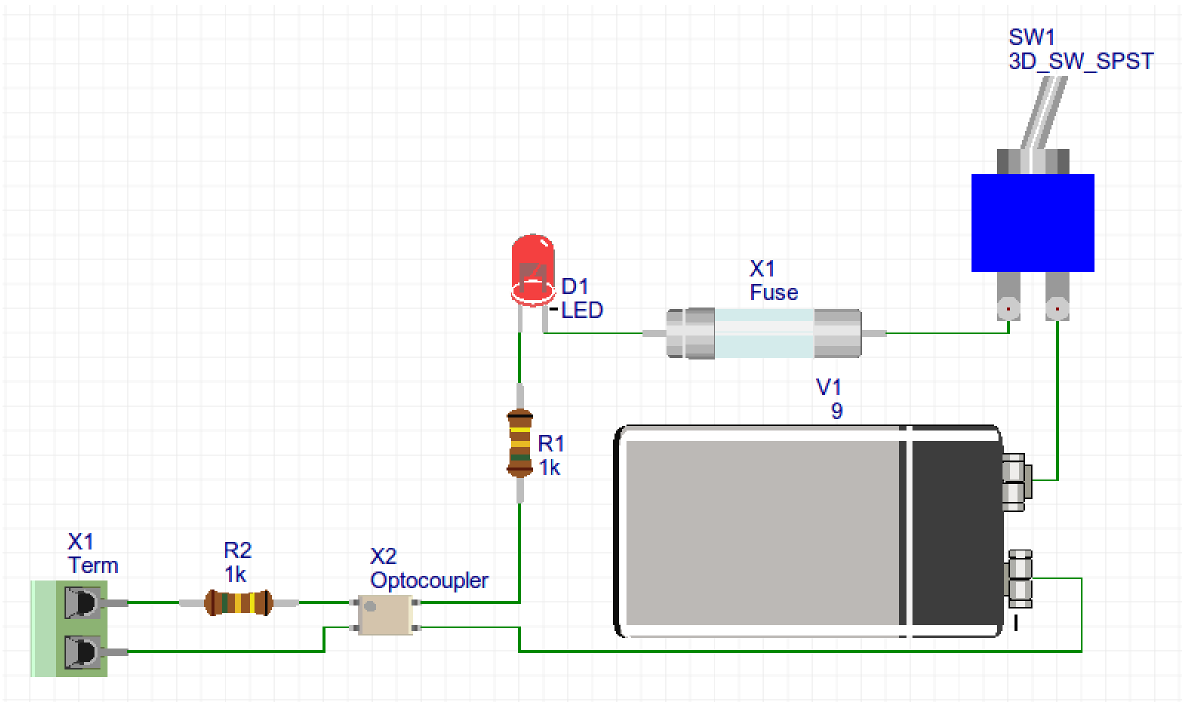
Because EasyEDA has some simple but powerful drawing capabilities, you can create your own symbols either by copying existing symbols into your own library and then editing and saving them, or by drawing them from scratch.

There is also a **Symbol Wizard** to quickly draw new symbols for `DIP`, `QFP` and `SIP`.



A feature of EasyEDA is that as well as extensive libraries of the usual simple "2D" graphical schematic symbols, it has a library of drawn "3D" component symbols, i.e. symbols that look like the physical components that they represent.

Using the drawing features in symbol creation, your schematic can be built like this:



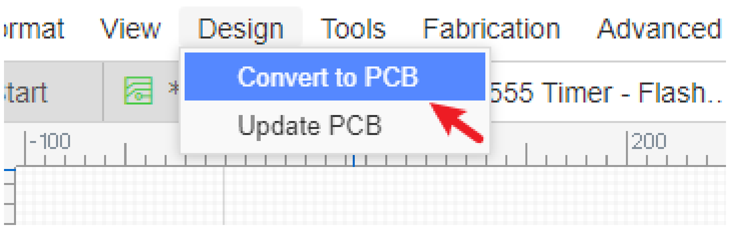
Another powerful feature is that it is possible to import symbols from `Kicad`, `Eagle` or `Altium` libraries.

PCB Layout

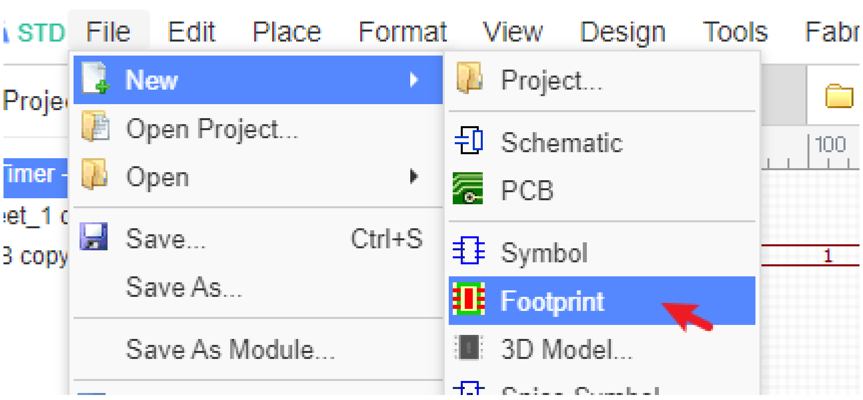
When you are satisfied with your schematic design and simulation results, you can then quickly proceed to produce your finished and populated PCB without leaving EasyEDA.

EasyEDA's PCB Design canvas helps you to quickly and easily lay out even complex multilayer designs from schematics you have already created in the Schematic canvas or directly as a layout with no schematic.

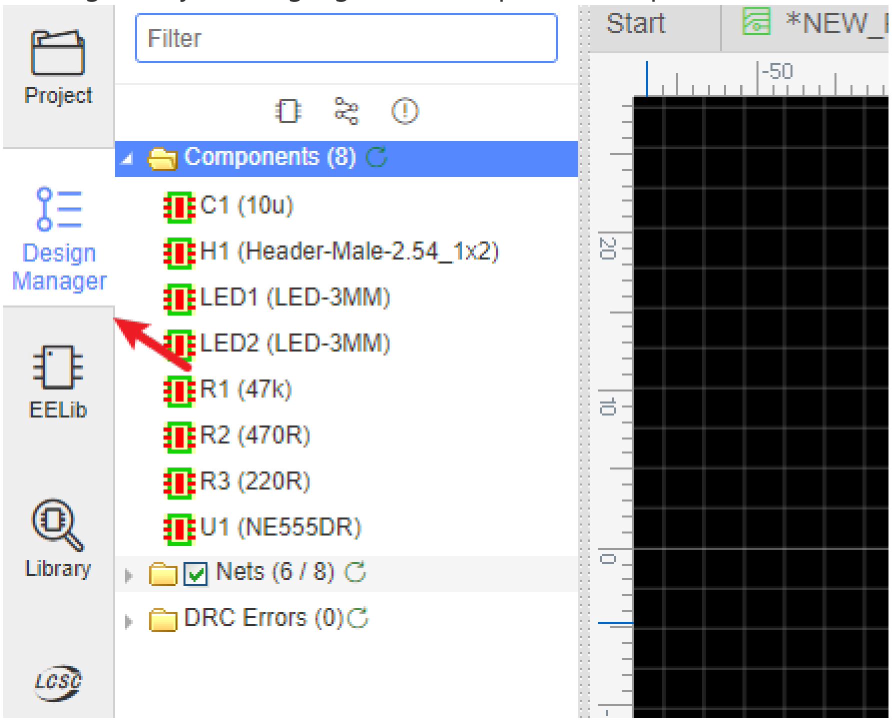
- Passing an EasyEDA Schematic into the PCB Design editor is as easy as clicking a button: Just click the **Convert to PCB** using: "Menu - Design - Convert to PCB".



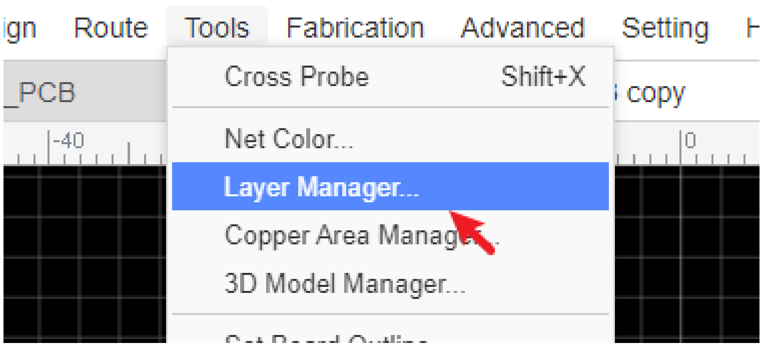
- EasyEDA has extensive component footprints. You can also build up your own library of unusual and specialized parts by copying and modifying existing parts or from scratch using EasyEDA' s powerful footprint creation and editing tools.



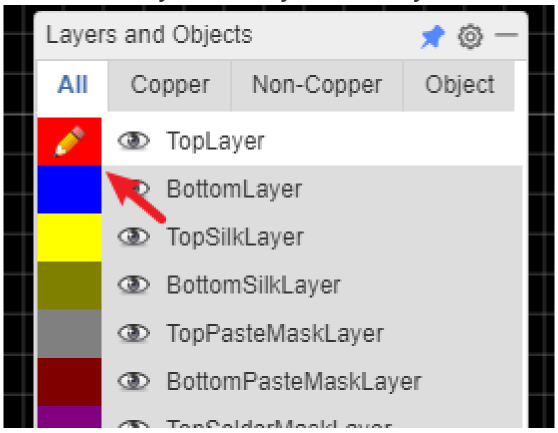
- When working in the PCB Design canvas there is a PCB Design Manager which works in a similar way to the Schematic design canvas, this will help you locate items and navigate your way around.
  Left Navigation Panel > Design Manager
  The PCB Design Manager is a powerful tool for finding components, tracks (nets) and pads (Net Pads).
  Clicking on any item highlights the component and pans it to the center of the window.



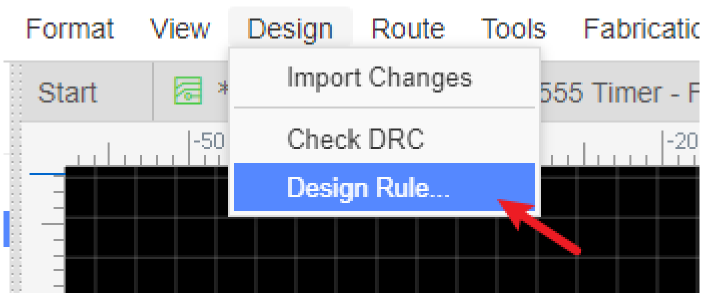
- You can set up layers used in the PCB and their display colours and visibility using **Top Menu - Tools - Layer Manager...**



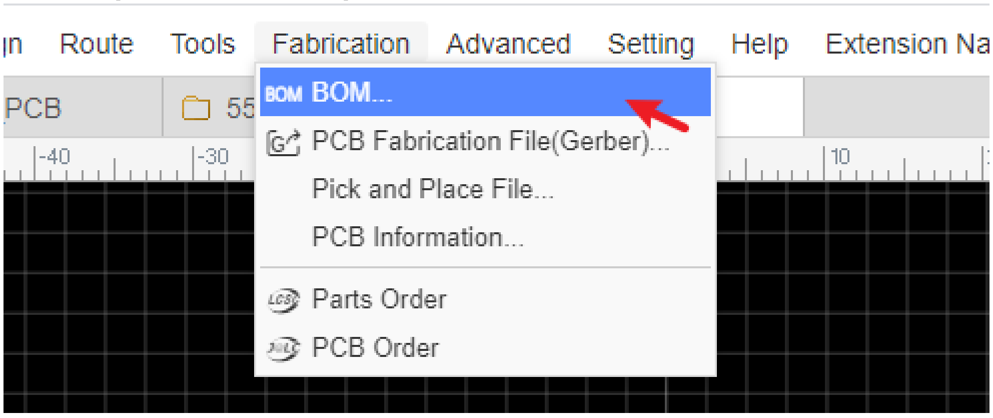
The active layer and layer visibility can be selected using the Layers Toolbar.



- Default track widths, clearances and via hole dimensions can all be configured in the Design Rule Check dialog which is opened by:
  Top Menu > Design > Design Rule...



  The Design Rule Check (**DRC**) is created when beginning your board layout. It can also be modified at any time. Running a DRC is one of the last steps in checking your PCB design before you generate **Gerber** and **Drill** files for board manufacture and are ready to place your order for a finished PCB.
- The final step is to check the Gerber and Drill files using a software Gerber viewer. This is an easy to install and use Open Source Software Gerber Viewer: [Gerbv: http://gerbv.geda-project.org/](http://gerbv.geda-project.org/)
- While you are waiting for your PCB to be delivered or at any time it is needed, you can create a Bill of Materials (BOM) with:
  File > Export BOM... or **Top Menu - Fabrication - BOM...**



- You can produce professional quality `SVG`, `.png` or `.pdf` output files for your documentation.

PCB Designs can be shared with colleagues and made public in the same way as Schematics. The size of PCB that you can produce using EasyEDA is almost unlimited: designs of over 100cm * 100cm are possible ... but you might need a powerful computer for that.

EasyEDA supports up to 6 layer PCBs by default but it is capable of handling more, so if you need more layers then please contact us.

Search footprints

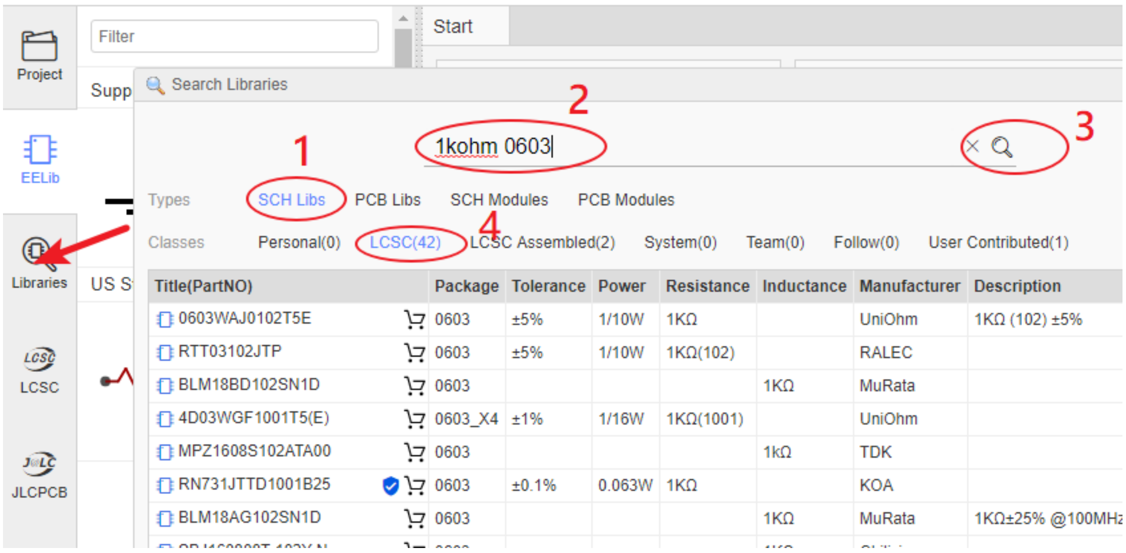
Searching footprints is the same as searching symbols by using **Library** in the Schematic editor.
You can place the selected footprints in the canvas after a successful search.

Libraries Management

Thanks to the Free and Open Source Kicad Libs and some Open Source Eagle libs, EasyEDA now has 700,000+ components, which should be enough for most projects!

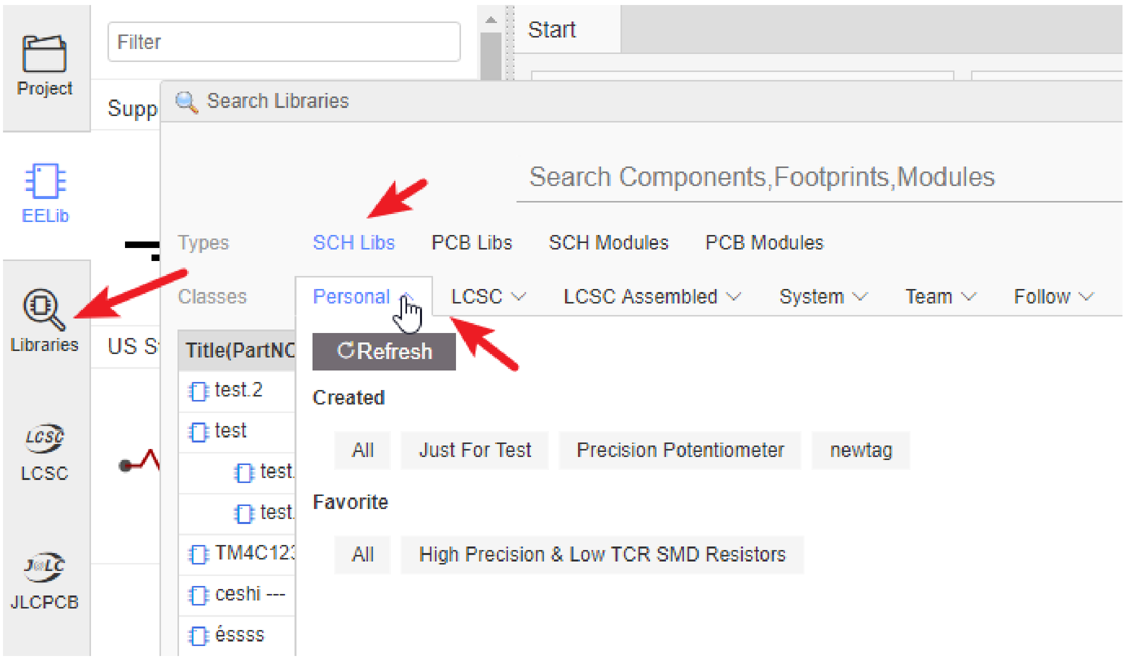
With these libraries you can enjoy using EasyEDA without having to spend so much time hunting for or building schematic symbols and PCB footprints.

- **Library**
  On the left-side Navigation panel you will find "**EElib**" and "**Library**", just type what components you want and search.
  At Libraries:

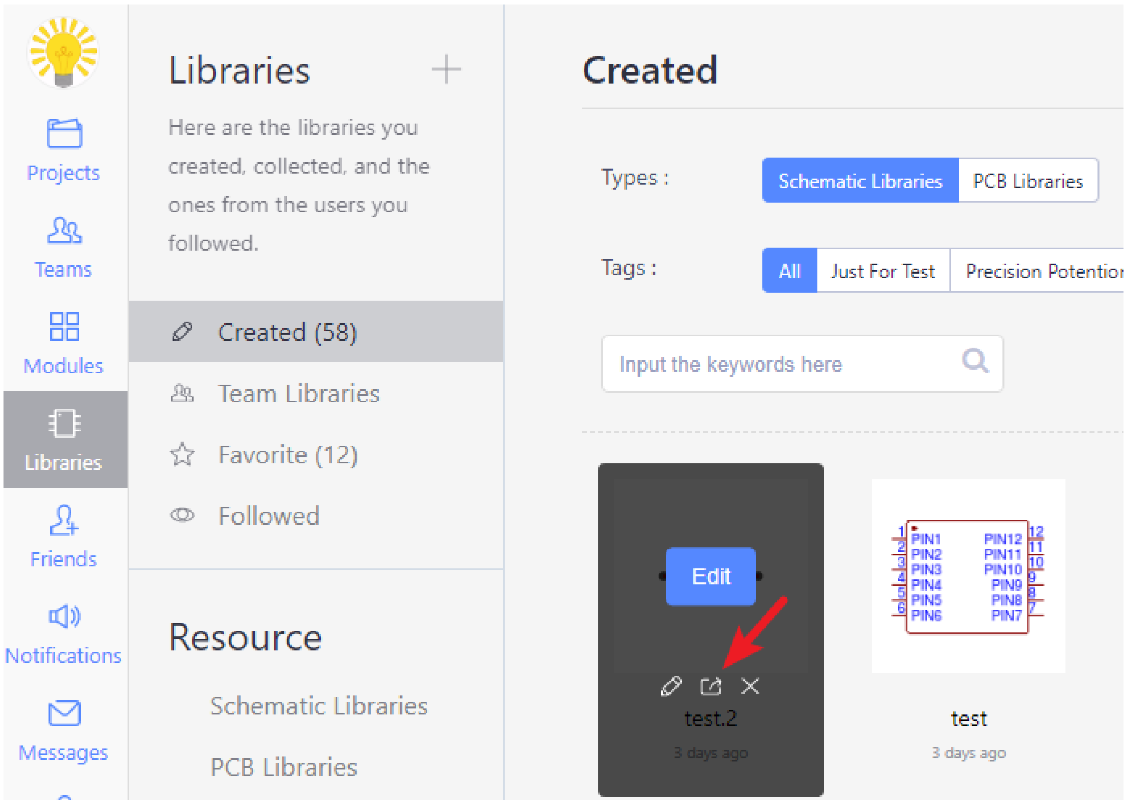


Title(PartNO)	Package	Tolerance	Power	Resistance	Inductance	Manufacturer	Description
0603WAJ0102T5E	0603	±5%	1/10W	1KΩ		UniOhm	1KΩ (102) ±5%
RTT03102JTP	0603	±5%	1/10W	1KΩ(102)		RALEC	
BLM18BD102SN1D	0603				1KΩ	MuRata	
4D03WGF1001T5(E)	0603_X4	±1%	1/16W	1KΩ(1001)		UniOhm	
MPZ1608S102ATA00	0603				1kΩ	TDK	
RN731JTTD1001B25	0603	±0.1%	0.063W	1KΩ		KOA	
BLM18AG102SN1D	0603				1KΩ	MuRata	1KΩ±25% @100MHz

  Steps:
  - 1. Choose the library type
  - 2. Type keywords such as "1k 0603"
  - 3. Click the search button
  - 4. Make your choice from the search results
  - 5. When you are done searching remove all the keywords
- **Create Library**
  EasyEDA supports creating your own symbols. After creation you can find your own components at **Library** > **Symbols/Footprints** > **Workspace**, and it is easy to manage your libraries.



- **Transfer Libraries**
  If you want to transfer your libraries to the team, you can do that in "User Center > Libraries > Personal".



To prepare for the final assembly stage you can create a Bill of Materials (**BOM**) using: **File > Export BOM...**

You can also produce professional quality `SVG`, `.PNG` or `.PDF` output files for your documentation.

All EasyEDA Schematic Symbol and PCB Footprint libs are public, so after you have created and saved a new symbol or footprint, others will be able to find your part. You will be credited as a contributor. https://easyeda.com/page/contribute

Version-Control

EasyEDA provides a simple but powerful version control feature. Each version is independent, you can edit and save each version.
When creating a new project, the default name will be set to "master", you can edit the name using the "Project Manage - Version" page.

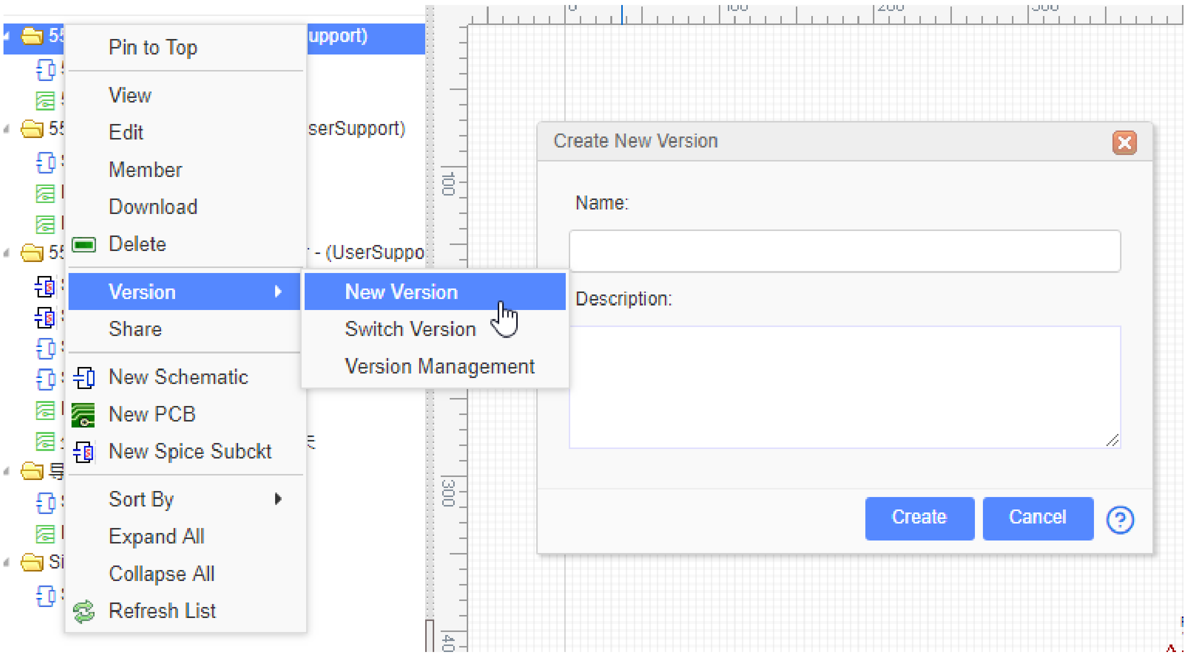
You can create up to 10 versions for every project. To create a new version, you must first delete an older version.

Create New Version

Use: Project folder - right-click menu - Version - New Version

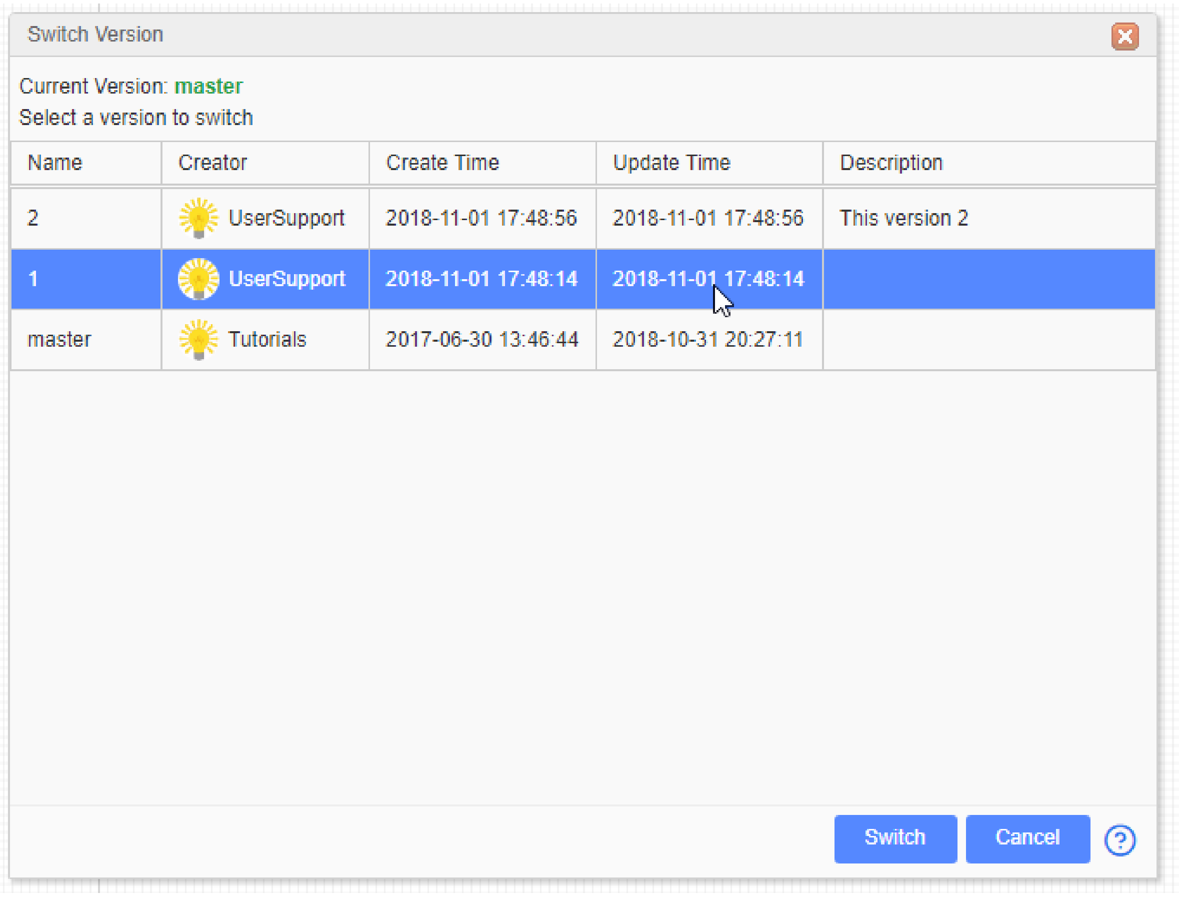
In the new version dialog, you type the version's name and description, and then create it.

To switch to another version use "Version - Switch Version".



Switch Version

Click "Switch", the dialog will list the current version and all the other versions for this project, you can select one.



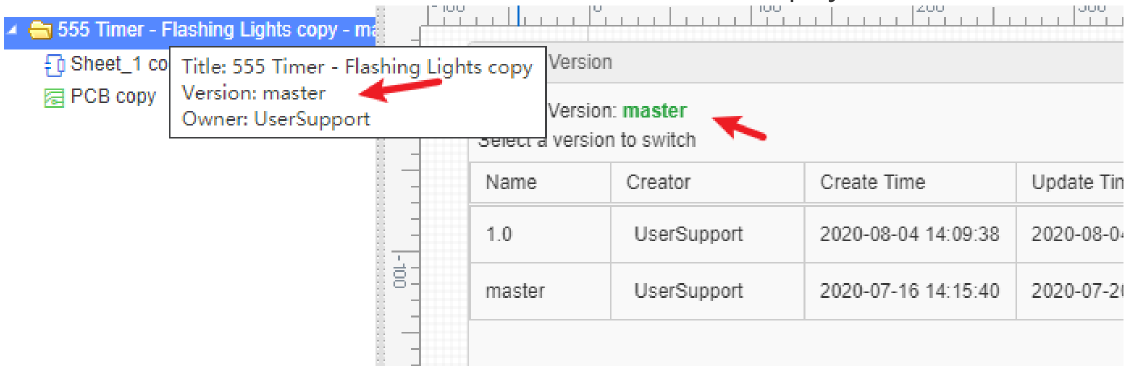
Switch Version

Current Version: master
Select a version to switch

Name	Creator	Create Time	Update Time	Description
2	UserSupport	2018-11-01 17:48:56	2018-11-01 17:48:56	This version 2
1	UserSupport	2018-11-01 17:48:14	2018-11-01 17:48:14	
master	Tutorials	2017-06-30 13:46:44	2018-10-31 20:27:11	

Note:

- Before switching to the other version, you must close the current version.
- You only can open the current version document, if you want to open other's version's document, you must switch the version first.
- If you are not sure which version it is, you can check it using "Switch Version" dialog to check the "Current Version", or hover the mouse cursor on the project folder.



Title: 555 Timer - Flashing Lights copy
Version: master
Owner: UserSupport

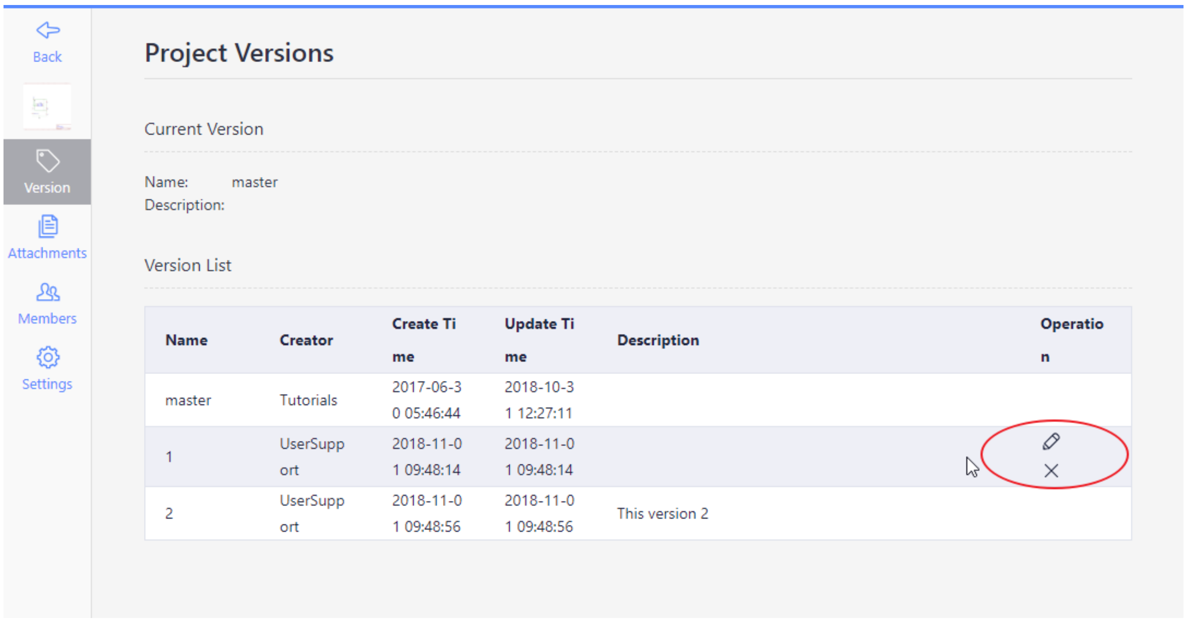
Version

Version: master
Select a version to switch

Name	Creator	Create Time	Update Tin
1.0	UserSupport	2020-08-04 14:09:38	2020-08-0
master	UserSupport	2020-07-16 14:15:40	2020-07-2

Version Management

Using "Version Management" will open the "Project Page - Version".
which will list all versions. You can edit each versions name and description or delete them. The current version cannot be deleted.



Project Versions

Current Version

Name: master
Description:

Version List

Name	Creator	Create Time	Update Time	Description	Operation
master	Tutorials	2017-06-30 05:46:44	2018-10-31 12:27:11		
1	UserSupport	2018-11-01 09:48:14	2018-11-01 09:48:14		
2	UserSupport	2018-11-01 09:48:56	2018-11-01 09:48:56	This version 2	

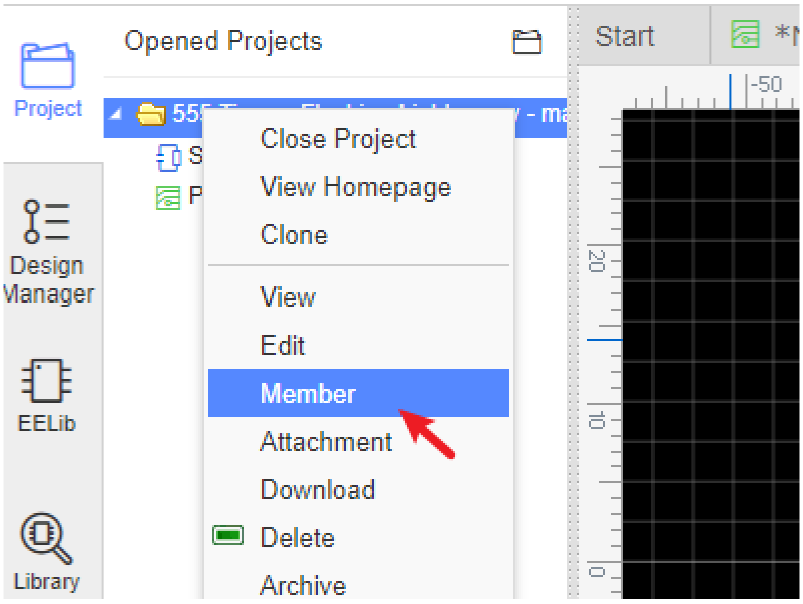
Project Member

How to share project with selected people.

Can you share a private project with your partner? Can your partner modify your designs?

If the answer is yes, you can use **Member** to do this.

Right click on the project and you will see the **Member** on the context menu; clicking on it will open the Member webpage.



To share a project with someone,

1. You need to know the E-mail address they used to create an account with EasyEDA.
2. The project member is set as "Developer", "Manager", or "Observer".

After setting up **Member** and Permissions, your partner will find your project in the **Open Project** when they log in.

If you partner does not wish to accept the shared project, they can reject it by leaving the project when they enter this project "Member" function.

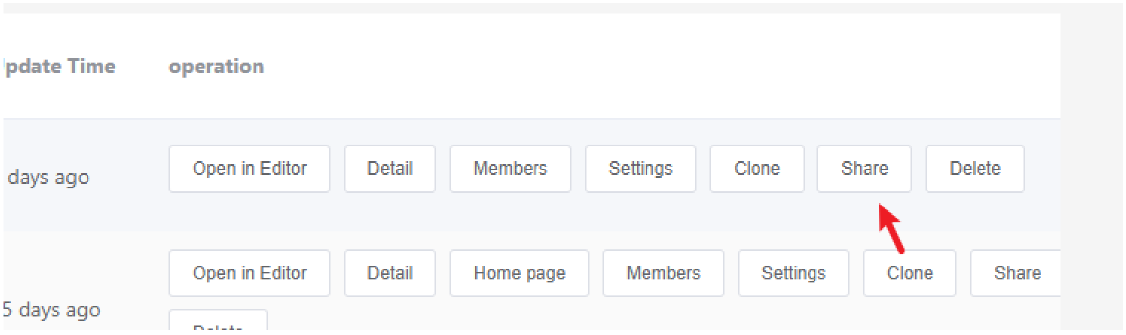
Share with Public

Sharing your work with others is a big feature of web based EDA tools and EasyEDA is no exception in offering great features.

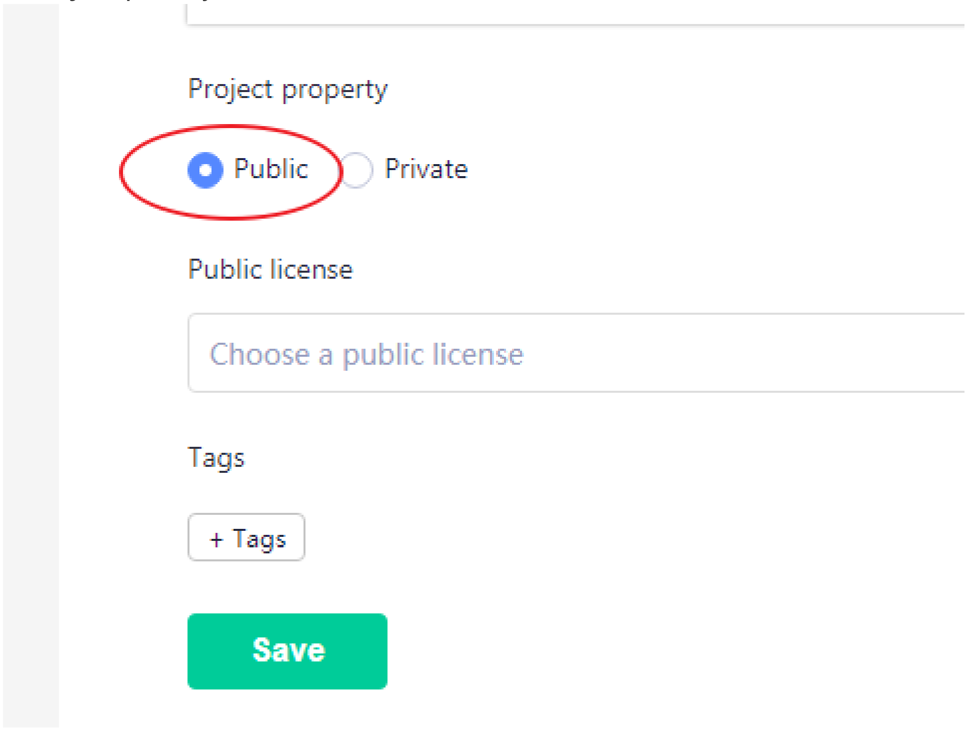
Did you create a cool project with EasyEDA? Show it off and be super helpful to other EasyEDA users, you just need to set your projects to public, so others can explore your circuits.

All projects in EasyEDA are set to private by default, your private project is not shared with anyone.
To make it public, you should right click and edit your existing project to make it a Public project:

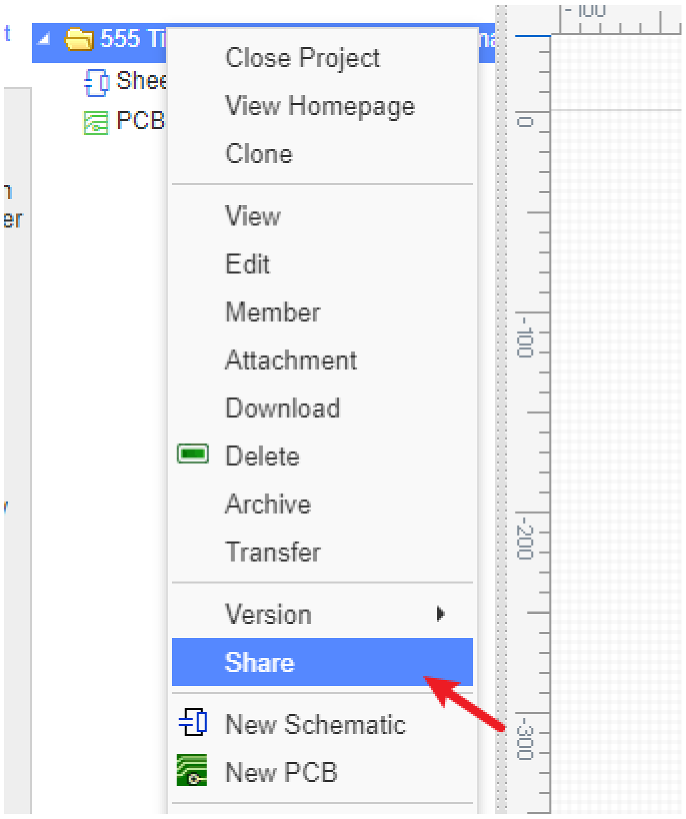
- In the Workspace, click the **Share** icon when the mouse hovers over the project cover, it will ask you to confirm.



Or enter project manage page, using "Workspace > Project > Manage > Settings > Basci > Project proerty:Public"


Project property
[✓ Public] Private
Public license
Choose a public license
Tags

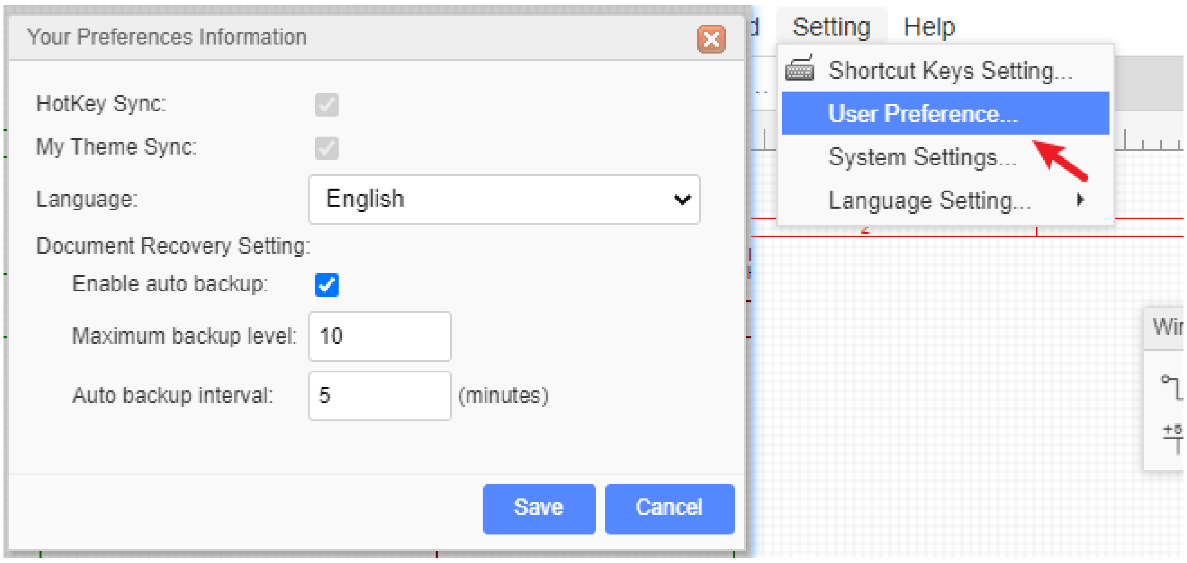
- In the editor, you right-click the project, then click the `Share` menu after setting the project as public.



User Preference

When EasyEDA shows the login success popup in the bottom right of the window, the user management menu will look like this:

Click on **Top Menu - Setting - User Preferences**,


Your Preferences Information

HotKey Sync: ✓

My Theme Sync: ✓

Language: English

Document Recovery Setting:

Enable auto backup: ✓

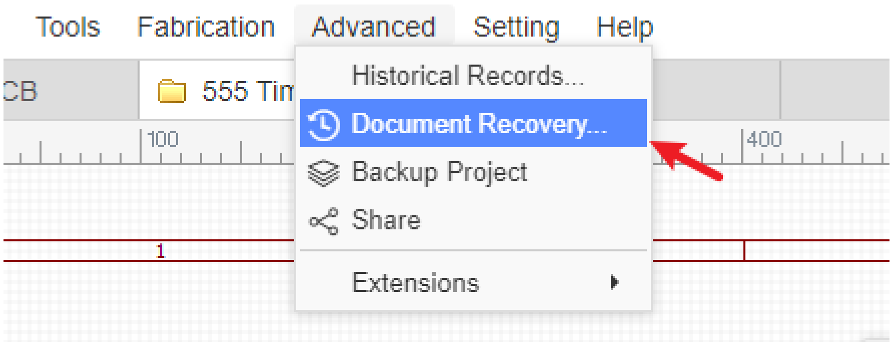
Maximum backup level: 10

Auto backup interval: 5 (minutes)

Document Recovery Setting:

- **Maximum backup level**: Every opened document can be saved as a backup, up to this number of different revisions.
- **Auto backup interval**: This is the time interval between auto saves of all your opened documents.

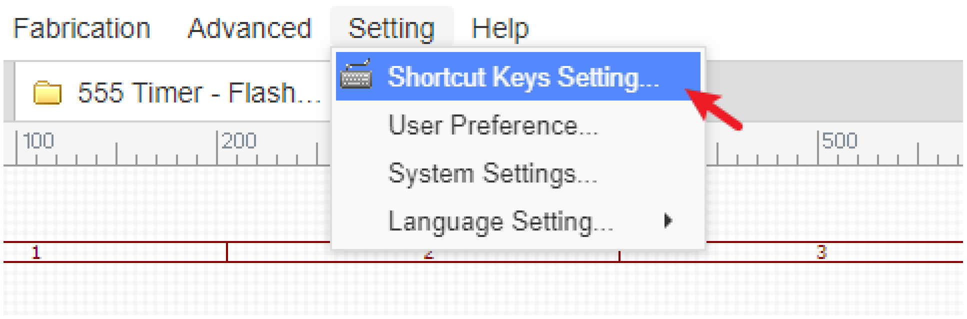
The Document Recovery function you can find at:



Shortcut Keys

While using an EDA tool suite, clicking all over the place with a mouse can get very tedious and seriously reduces your productivity. Keyboard shortcuts or Hotkeys avoid much of that. EasyEDA not only provides many hotkeys, but also every hotkey can be reconfigured to your personal preference.

Under the Setting menu, click the Hotkeys Setting... Menu which will open the Hotkey Setting dialog.



To change a Hotkey, click anywhere in the row for the hotkey you want to change and then press your new key.

For example, if you want to use `R` instead of `space` to rotate selected objects, click on the first row, then press `R`.

After changing any hotkey, you must click the Save Changes button.

The **docType** column describes which type of EasyEDA document each hotkey applies to. **docType** has three types:

- **ALL**: any document type in EasyEDA.
- **SCH**: schematic and schematic libs
- **PCB**: PCB and Footprints.

The functions of some hotkeys may change between docTypes. For example, the hotkey `C` draws an Arc in SCH but draws a circle in PCB.

A list of all the available default hotkeys is given below.

All document

DocType	Shortcut	Function
All	Space	Rotate selected objects
All	Right-Click	Keep right-click to pan canvas; Open offset dialog when select one object
All	Left	Scroll Or Move selected left
All	Right	Scroll or Move selected right
All	Up	Scroll or Move selected up
All	Down	Scroll or Move selected down
All	TAB	Change object's attributes when placing; Open offset dialog when selecting an object
All	Esc	Cancel current drawing
All	Home	setting new canvas origin
All	Delete	Delete Selected
All	F1	Open tutorials
All	F11	Full screen at browser
All	A	Zoom In
All	Z	Zoom Out
All	D	Drag
All	K	Fit Window
All	R	Rotate selected objects
All	X	Flip Horizontal(doesn't support footprint)
All	Y	Flip Vertical(doesn't support footprint)
All	ALT+F5	Full screen at browser
All	ALT+W	Close current tab
All	SHIT+ALT+W	Close all tabs
All	CTRL+X	Cut
All	CTRL+C	Copy
All	CTRL+V	Paste
All	CTRL+A	Select All
All	CTRL+Z	Undo
All	CTRL+Y	Redo

DocType	Shortcut	Function
All	CTRL+S	Save
All	CTRL+F	Find Component
All	CTRL+D	Design Manager
All	CTRL+Home	Open canvas origin setting dialog
All	SHIFT+1	Cycle forward to next open tabbed document
All	SHIFT+2	Cycle backward to next open tabbed document
All	SHIFT+X	Cross Probe
All	SHIFT+F	Search Library
All	SHIFT+Drag	Cursor snap to part's origin
All	SHIFT+ALT+H	Align horizontal centers
All	SHIFT+ALT+E	Align verticas centers
All	CRTL+SHIFT+L	Align left
All	CRTL+SHIFT+R	Align right
All	CRTL+SHIFT+O	Align top
All	CRTL+SHIFT+B	Align bottom
All	CRTL+SHIFT+G	Align grid
All	CRTL+SHIFT+H	Distribute Horizontally
All	CRTL+SHIFT+E	Distribute Vertically
All	CTRL+SHIFT+F	Find similar objects

Schematic

DocType	Shortcut	Function
Schematic	W	Draw Wire
Schematic	B	Draw Bus
Schematic	U	Bus Entry
Schematic	N	NetLabel
Schematic	P	Place Pin
Schematic	L	Draw Polyline
Schematic	O	Draw Polygon
Schematic	Q	Draw Bezier
Schematic	C	Draw Arc
Schematic	S	Draw Rect
Schematic	E	Draw Ellipse
Schematic	F	Freehand Draw
Schematic	T	Place Text
Schematic	I	Edit Selected Symbol
Schematic	CTRL+Q	NetFlag VCC
Schematic	CTRL+G	NetFlag GND
Schematic	F8	Run the Document Simulation
Schematic	CTRL+J	Open the Simulation Setting
Schematic	CTRL+SHIFT+X	Cross Probe and Place
Schematic	SHIFT+T	Open symbol wizard
Schematic	ALT+F	Open footprint manager

PCB

DocType	Shortcut	Function
PCB	W	Draw Track
PCB	U	Draw Arc
PCB	C	Draw Circle
PCB	N	Draw Dimension
PCB	S	Draw Text
PCB	O	Draw Connect
PCB	E	Draw copperArea
PCB	T	Change To TopLayer; Change selected part to toplayer
PCB	B	Change To BottomLayer; Change selected part to bottomlayer
PCB	1	Change To Inner1
PCB	2	Change To Inner2
PCB	3	Change To Inner3
PCB	4	Change To Inner4
PCB	P	Place Pad
PCB	Q	Change canvas unit
PCB	V	Place Via
PCB	M	Measure
PCB	H	Highlight Net all the time, press it again cancel highlight
PCB	L	Change Route Angle
PCB	-	Decrease Routing Width; Switch to the forward signal layer

DocType	Shortcut	Function
PCB	+	Increase Routing Width; Switch to the next signal layer
PCB	*	Cycle switch to the next signal layer
PCB	Delete	Delete selected object; Undo the track when routing
PCB	ALT--	Decrease Snap Size
PCB	ALT++	Increase Snap Size
PCB	CTRL+R	Depend on reference point for copy object repeatly
PCB	CTRL+L	Open layer manager
PCB	CTRL+Q	Hide/show network text
PCB	SHIFT+M	Remove All Copper Area fill data
PCB	SHIFT+B	Rebuild All Copper Area
PCB	SHIFT+D	Move Object(s) by reference point
PCB	SHIFT+G	Display track length while routing
PCB	SHIFT+W	Show favorite track width while routing
PCB	SHIFT+R	Change routing conflict
PCB	SHIFT+S	Toggle layers which is not active
PCB	SHIFT+Double Click	Delete selected track segment
PCB	CTRL+SHIFT+V	Paste object(s) and keep the prefix, and hide the ratline layer
PCB	CTRL+SHIFT+SPACE	Change routing angle, same as hotkey L
PCB	CTRL+ALT+L	Enable all layers
Footprint	CTRL+SHIFT+ALT+D	Open custom data dialog

Basic Skills

To use EasyEDA you need to be familiar with a few basic terms and concepts. The best way to learn them is to open up EasyEDA, and create a new schematic:

File > **New** > **Schematic** , and play!

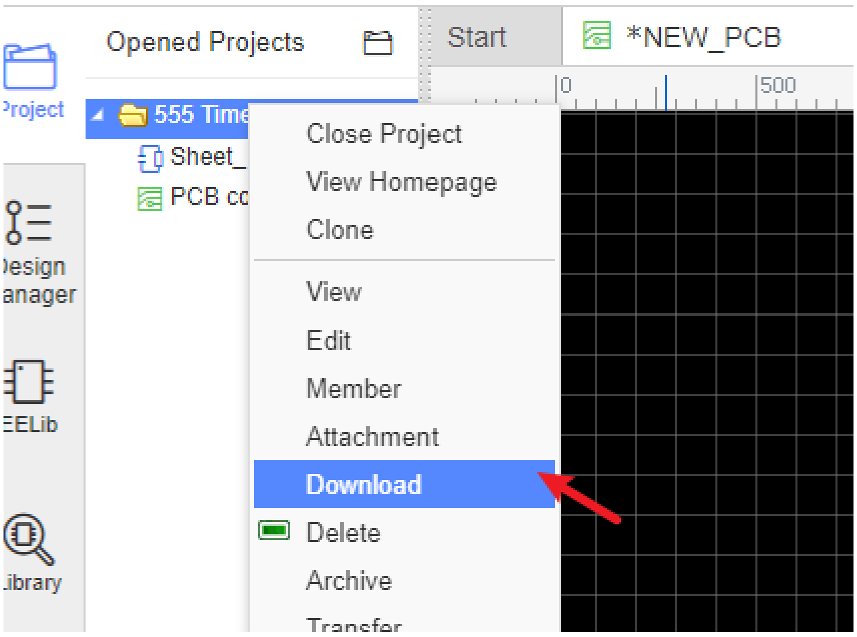
Saving Your Work Locally

Although EasyEDA saves all your files on our Server, sometimes you may want to save your work locally and EasyEDA does provides a way to do this.

You can right-click your project folder, and click "Download Project", or export your design as EasyEDA source file via "File > EasyEDA Source".

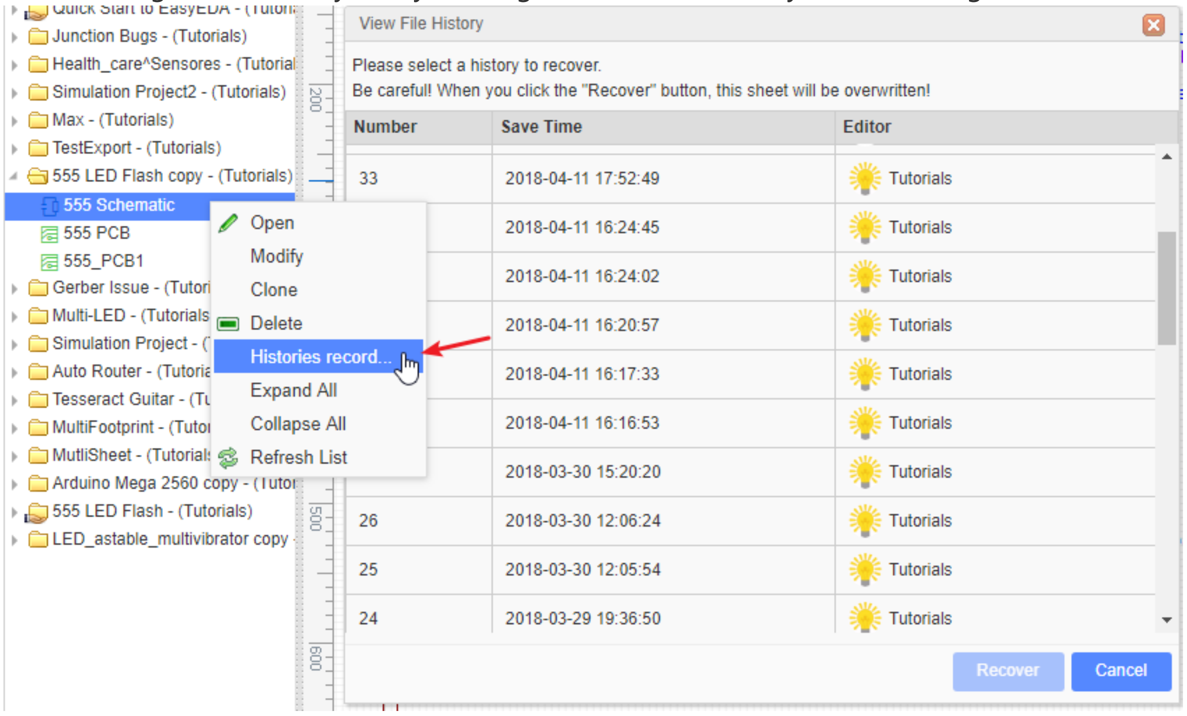
For more detail you look here: Export EasyEDA Source section.

To download your project:



History Record

It is easy to use this function, right click on the document for which you need the history as in the image below:
After clicking on the history link, you will get a list of the History as in the image below.



View File History

Please select a history to recover.
Be careful! When you click the "Recover" button, this sheet will be overwritten!

Number	Save Time	Editor
33	2018-04-11 17:52:49	Tutorials
	2018-04-11 16:24:45	Tutorials
	2018-04-11 16:24:02	Tutorials
	2018-04-11 16:20:57	Tutorials
	2018-04-11 16:17:33	Tutorials
	2018-04-11 16:16:53	Tutorials
	2018-03-30 15:20:20	Tutorials
26	2018-03-30 12:06:24	Tutorials
25	2018-03-30 12:05:54	Tutorials
24	2018-03-29 19:36:50	Tutorials

Click the History number to open the saved file in the editor, if this is what you need, you can save it to your project and delete your bad file.

Note:
1. For now, all the History entries are tagged with a number. An upcoming feature will allow you to add a custom tag.
2. Saving your files too frequently will create many History entries and it will become more difficult to find the exact one you want.

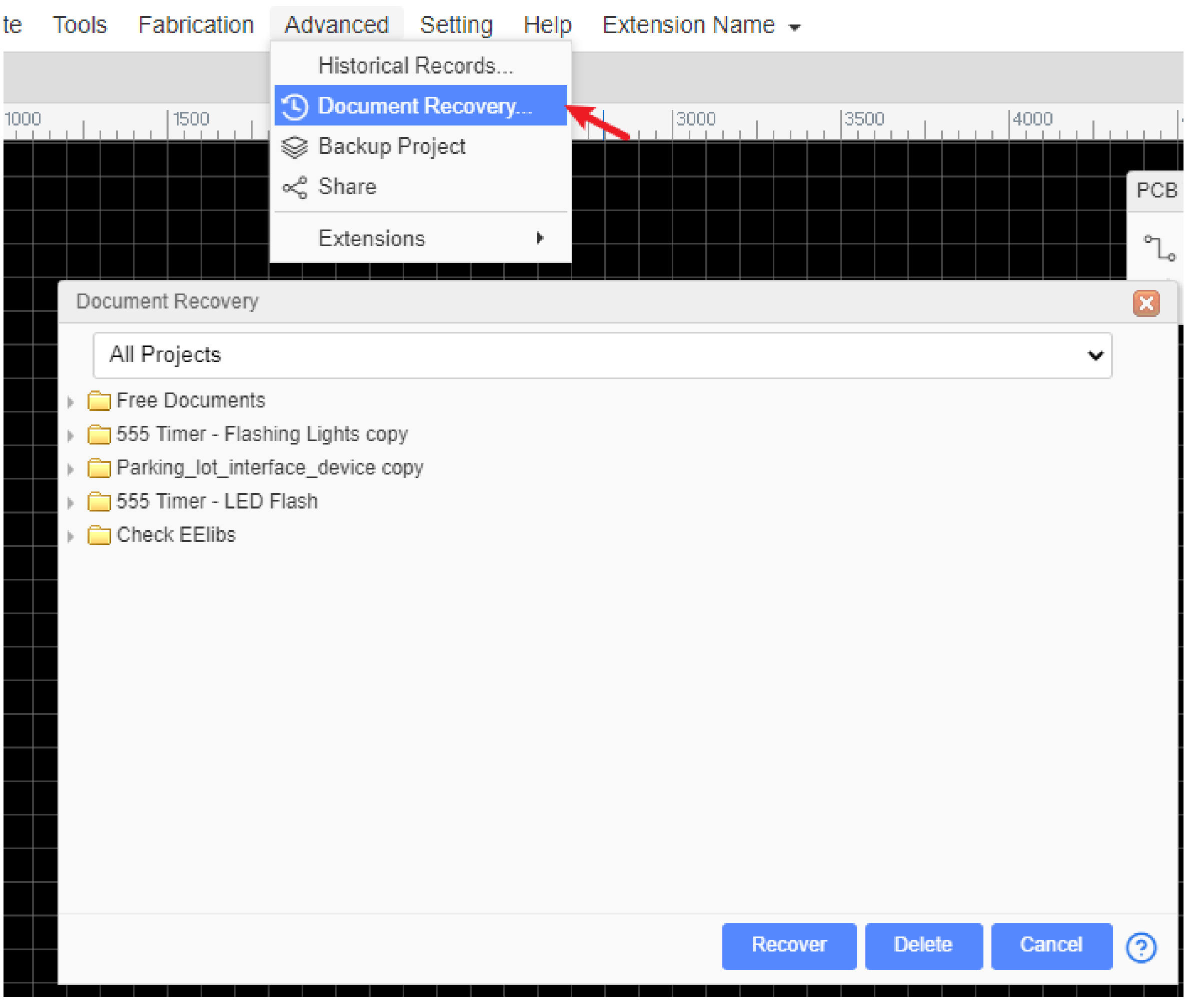
Document Recovery

No operating system, software or network is perfect, so sometimes things go wrong. Having your Desktop or web browser freeze or your broadband connection drop, two hours into laying out a PCB, could spoil your day.

However, with EasyEDA, your day will be just fine.

This is because EasyEDA auto saves and makes backups of all your open files to your computer, crash recovery is built into EasyEDA.

On the top menu, click **Menu - Advanced - Document Recovery** as below:



Expand the folder to the latest version, Select the file which you would like to **recover**, then click the Recover button; your file will be opened in a new tab, then save this opened file.

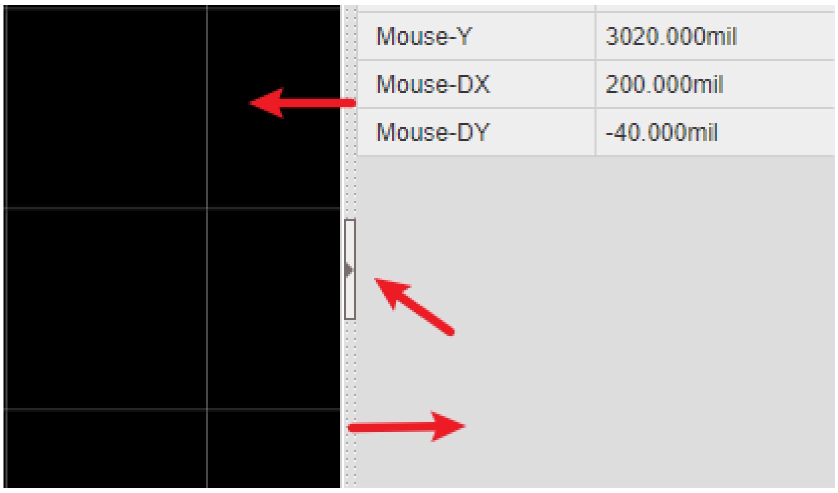
Please note:

- *EasyEDA saves these crash recovery files on your computer and not on the EasyEDA server. Therefore you cannot recover files from a crash on one computer or browser by changing to a different computer or browser.*
- *If you cleaned your browser's cache, the recovery files will disappear.*

- *If you made a mistake deleting a file and removing the cache, you might be able to find your document in the recycle bin: https://easyeda.com/account/user/recycles/personal.*

Resizing the canvas area

Hovering the mouse cursor over the areas indicated by the three green ellipses will bring up the blue sidebar toggle lines. Clicking on them will toggle the visibility of their associated right and left areas to expand the canvas area. The vertical lines can be dragged horizontally to resize the panels.

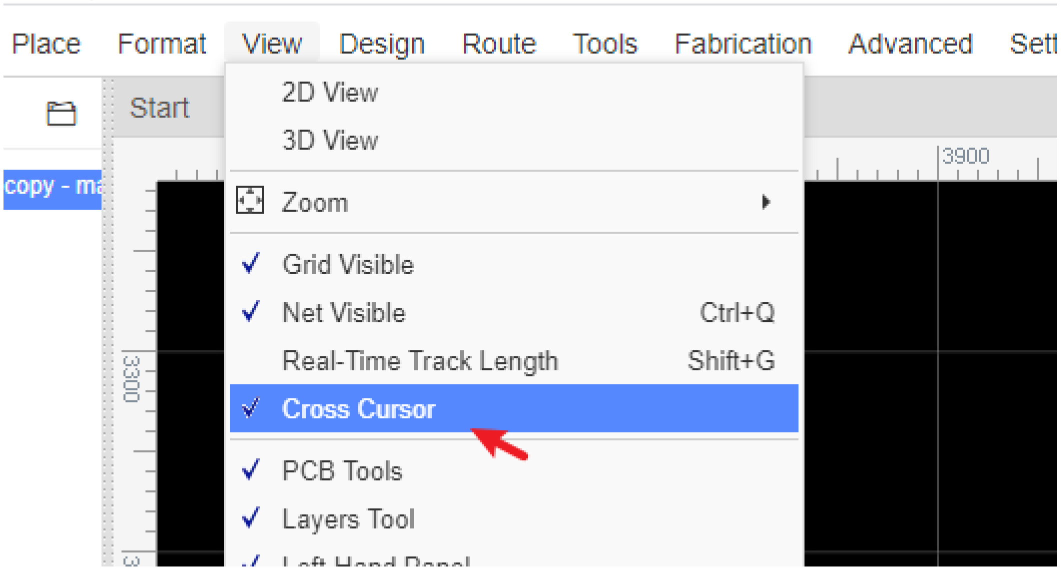


Mouse-Y	3020.000mil
Mouse-DX	200.000mil
Mouse-DY	-40.000mil

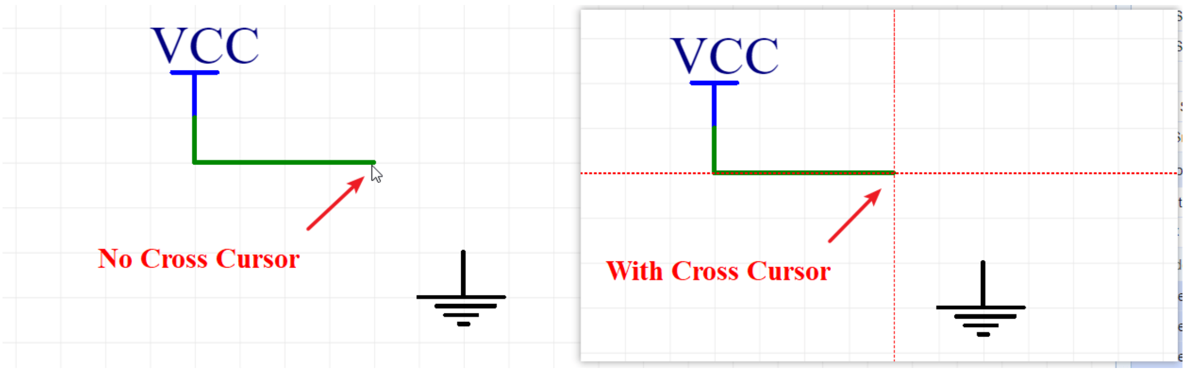
Cursor Style

Some users do not like the cross cursor, so you can change it to an arrow cursor like in the image below.

Via: Top Menu - View - Cross Cursor



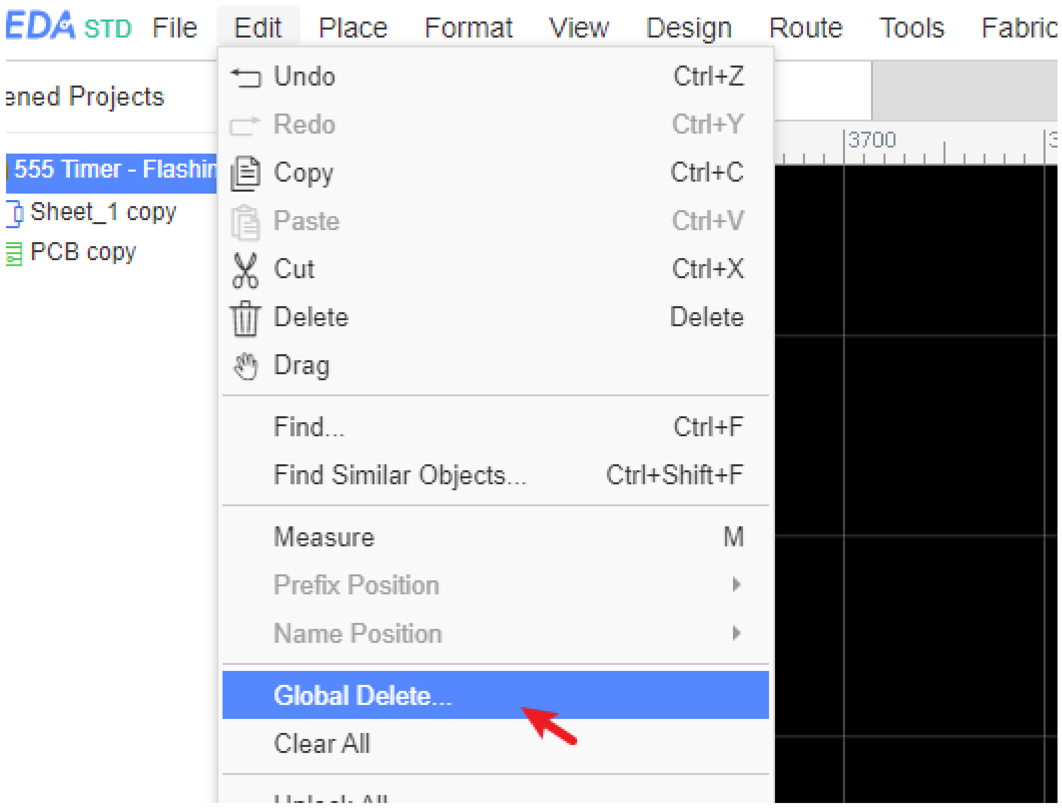
The difference between these options is shown below:



Clear and delete

If you think your schematic or PCB looks terrible, and you want to redraw everything, you can do this:

- **Top Menu > Edit > Global Delete**.



- Delete this schematic and create a new one.
- Click one object or CTRL+A, press delete key to remove all objects.

Left clicking

Similar to other EDA software:

- Click on an item to select it;
- If over a selected item, click and hold to drag a selected item;
- If not over a selected item, clicking and holding while dragging creates a selection box;
- The selection box, using click and drag to the right, selects everything inside the box;
- The selection box, using click and drag to the left, selects everything inside and intersected by the box;
- Double click on a text area to edit the text;
- The exact left click functionality depends on what item is being selected and in what Canvas the item exists (Schematic or PCB).

Right clicking

In EasyEDA, right-clicking opens a context sensitive menu:

- When you are placing a symbol, right-clicking will stop placing and return to select mode. This is the same as the ESC key.
- When you are drawing a shape such as a polyline, after a right click, the polyline will be stopped at the place where you right click but the mouse will remain as a **cross**, so you can draw another shape.
- To get out of the current active context sensitive command such as placement or drawing mode and go back to **select mode**, just double right click or press ESC (sometimes twice).

Right-Click and drag Right-clicking and holding the button anywhere in the Schematic, Waveform or PCB Canvas while dragging the mouse will move the canvas around within the EasyEDA window. Holding the middle button and dragging performs the same operation.

ESC key

Pressing the `ESC` key ends the current drawing action but does not exit the current active context sensitive command mode (i.e. it does not return the cursor to select mode). Pressing ESC again returns to select mode.

Select more shapes

- CTRL+Left Clicking on items adds those items to your selection;
- Clicking and holding creates a selection box;
- Creating a selection box, using click and drag to the right, selects everything inside the box;
- Creating a selection box, using click and drag to the left, selects everything inside and intersected by the box;

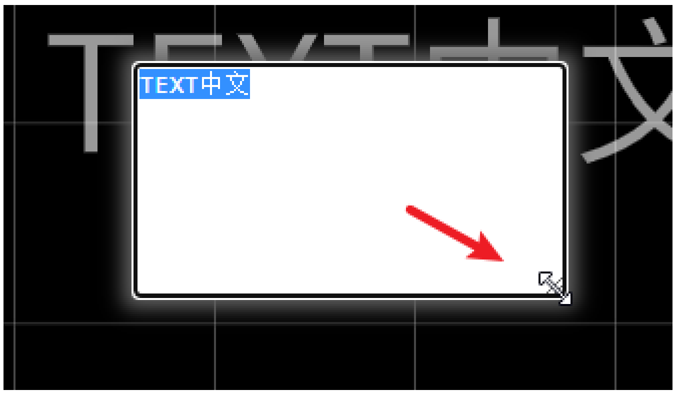
Zoom in and Zoom out

- Using the middle mouse button:
- Roll forward to zoom in;
- Roll back to zoom out;
- Using hotkeys, the default hotkey `A` for zoom in, `Z` for zoom out.

Please note:
Do not scroll your mouse at the same time as pressing the CTRL key when your cursor on the top menu, the browsers will zoom the whole website, if you just want to zoom the canvas in the EasyEDA window, you need to put your cursor onto the canvas. If the "zoom the whole website" happens, just press `Ctrl+0` to reset the browser view zoom.

Double clicks

Double clicking any text area opens a resizable text box that allows you to edit the text.



Press the enter key to save your changes. Click outside the box or press ESC to discard your changes.

Pan/Move Canvas

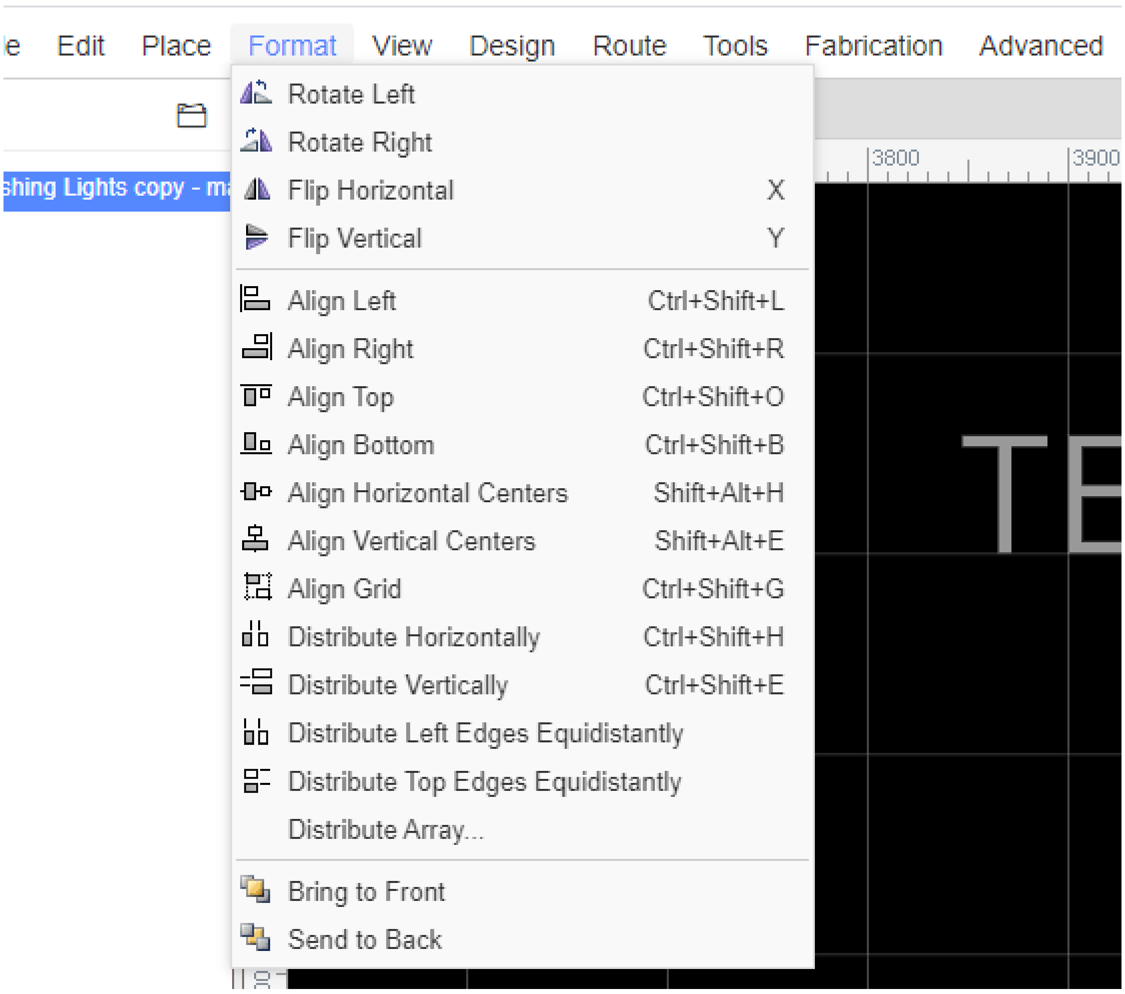
- Right click anywhere in the Schematic, WaveForm or PCB Canvas and Hold down right button to drags the canvas around within the EasyEDA window.
- If your canvas is bigger than the EasyEDA window and is showing scroll bars, you can use either the scroll bars or the Arrow keys to scroll the canvas to pan.
- When drawing a wire, a graphic line or shape that you wish to extend beyond the edge of the EasyEDA window holding down the left mouse button after starting the line will pan the canvas to keep the drawn item inside the window.

Tip:
If you use Chrome, and cursor is in the canvas while pressing CTRL or ALT key and rolling your mouse, the canvas will move vertically, and when pressing SHIFT and rolling your mouse, the canvas will move horizontally.

Rotate

After selecting one or more items, you can rotate the selected items using:

Top Menu > **Format** > **Rotate** or by pressing the default rotate hotkey: 'Space' .

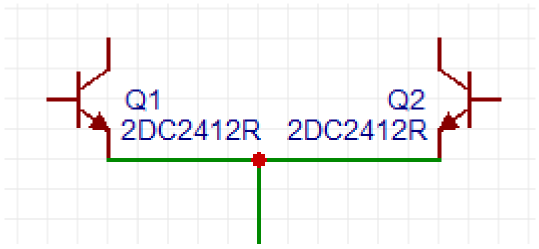


When in PCB view mode you can click the footprint and change its rotation in the right property panel.

Please note:
Rotating a multiple selection rotates each item about its own symbol origin. It does not rotate the items about the centroid of the group of items.

Flip

To place a Q2 as shown in the schematic below you need to Flip the item. Via: Top menu - Format - Flip.



You can Flip one or more selected items using:

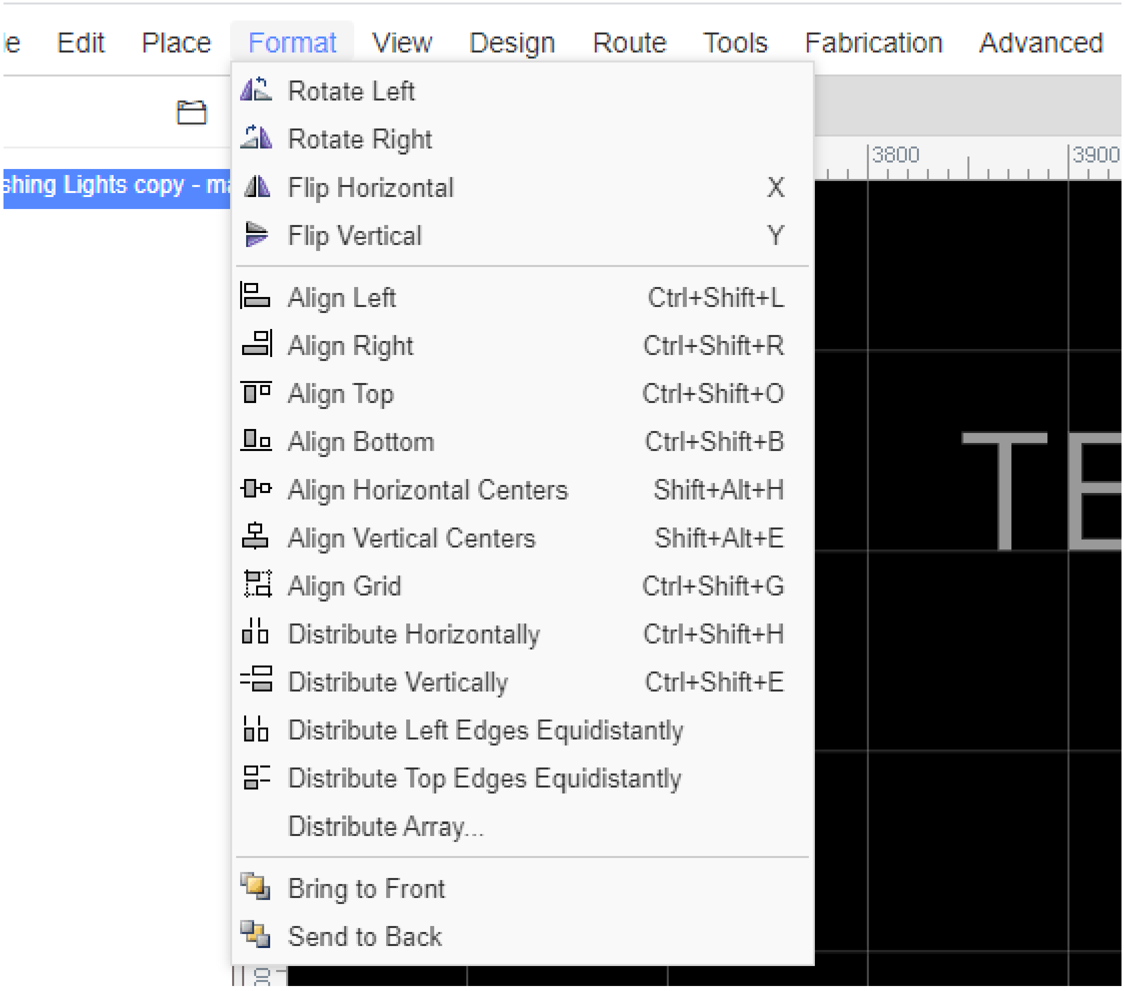
Rotate and Flip > Flip Horizontal or Flip Vertical from the toolbar,

or by pressing the default flip hotkeys: X to Flip Horizontal, Y to Flip Vertical.

Notice: Footprint does not support the flip command.

Align

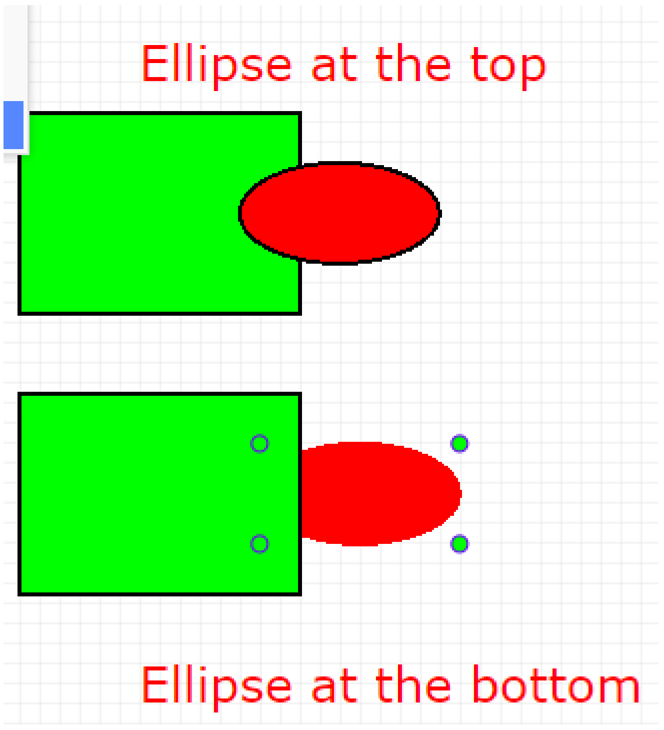
EasyEDA provides many align option features, you can align your symbols or footprints very easily using: Top menu - Format - Align. There are also icons on the toolbar for this.



Bring to Front and Send to Back

In the image below, both the rectangle and the ellipse are filled. Use: Top menu - Format - Bring/Send to Front/Back.

If you draw an ellipse before drawing a rectangle, the rectangle will overlap and therefore hide the ellipse. To reveal the ellipse, select the rectangle and then use the Bring and Send function, as shown:

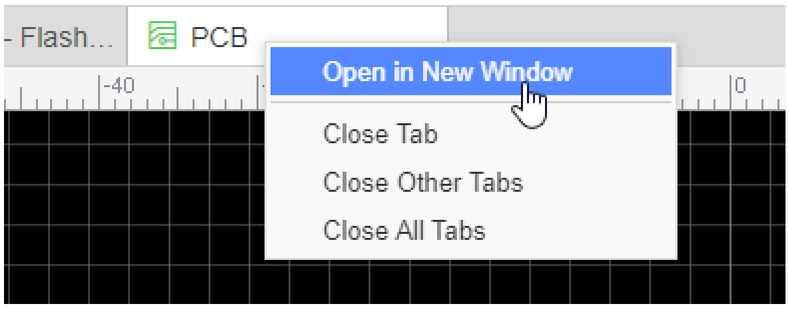


Multiple Windows

Since v6.4.0, EasyEDA supports multiple windows design.

How do it works?

1. Open schematic and PCB
2. Right-click the schematic or PCB tab, click "Open in New Window"

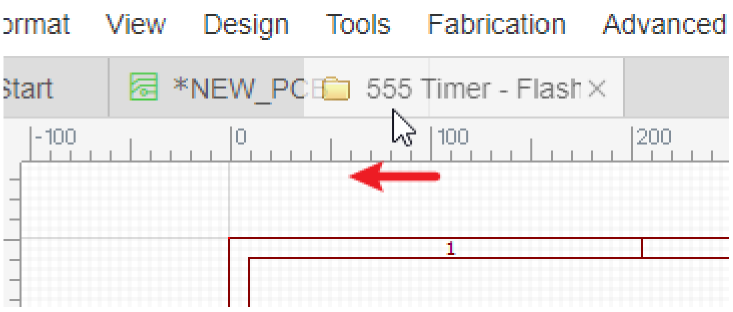


3. This will open the document in a new window, then you can do the cross probe: Click the component, pads, click the Design Manager list, the "Cross Probe and Place" also works.

Documents Tab Switch

It is easy to modify the tab positions of your documents.

Simply drag the tab location, or use the hotkeys SHIFT+1 and SHIFT+2

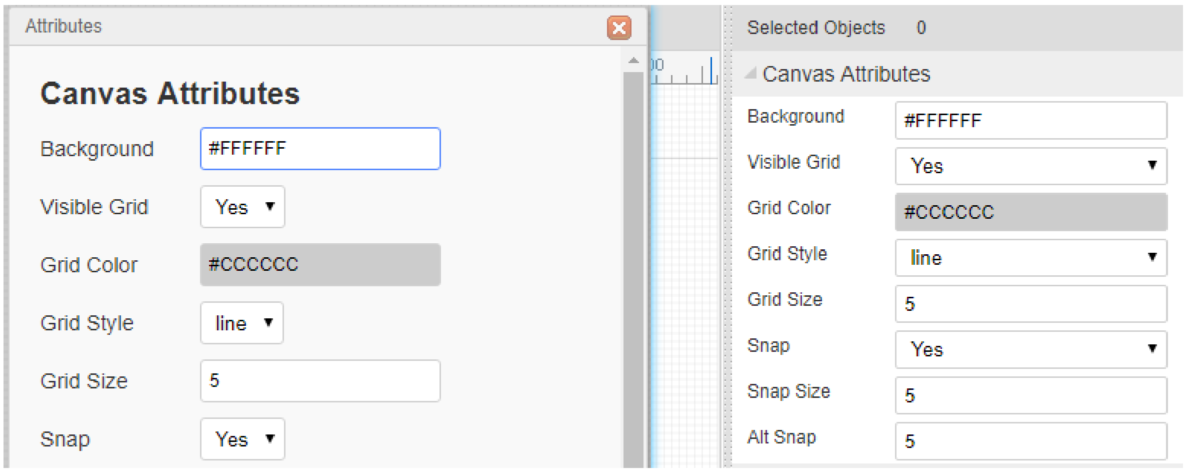


Schematic Capture

Canvas Setting

During this tutorial we will create a simple Schematic design to guide you in using EasyEDA Schematic capture.

You can find the canvas Properties setting by clicking on any the blank space in the canvas.



Canvas Attributes

Background #FFFFFF

Visible Grid Yes

Grid Color #CCCCCC

Grid Style line

Grid Size 5

Snap Yes

Snap Size 5

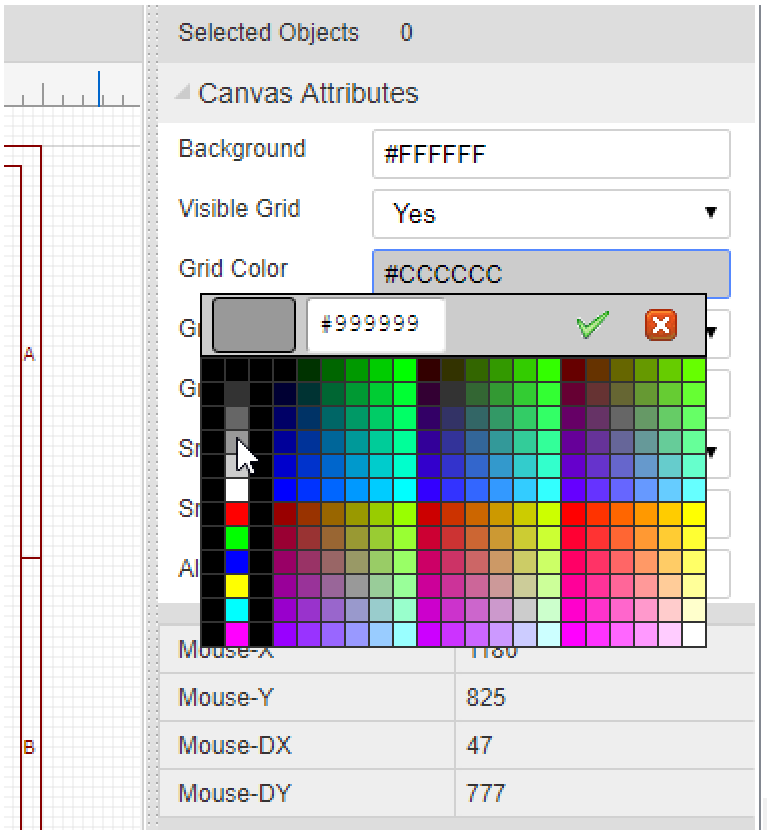
Alt Snap 5

As described earlier, background and grid colours and the style, size, visibility and snap **attributes** of the grid can all be configured.

The canvas area can be set directly by the Width and Height or by using the available preset frame sizes.

Grid:

- **Visible Grid** : Yes or No
- **Grid Color**: Any valid colour
- **Grid Style**: Line or Dot
- **Grid Size**: To ensure proper alignment of all EasyEDA parts, it is advisable to set in 10, 20, 100. the unit is pixel.
- **Grid** (and background) colour can be set directly by entering the hexadecimal value of the colour you want or by clicking on a colour in the palette that opens when you click on the colour value box:


Selected Objects 0

Canvas Attributes

Background #FFFFFF

Visible Grid Yes

Grid Color #CCCCCC

#999999

Mouse-Y 825

Mouse-DX 47

Mouse-DY 777

Snap:

- **Snap**: Yes or No. Pressing this key toggles switching snap to grid on and off.
- **Snap Size**: To ensure proper alignment of all EasyEDA parts, it is advisable to set in 10, 20, 100 but any valid number can work, such as 1, 5, 10.

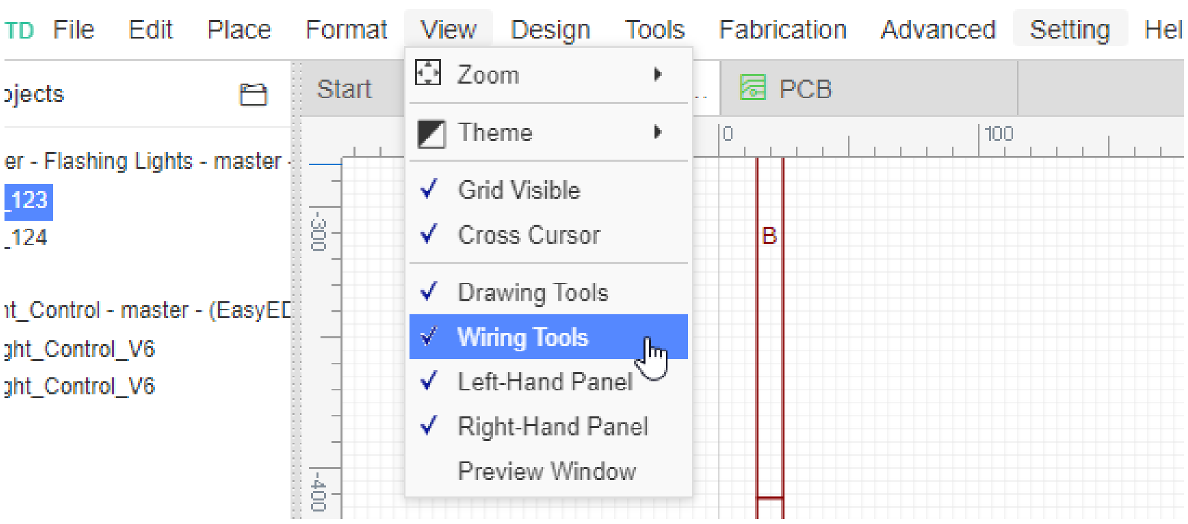
It is strongly recommended that you keep **Snap = Yes** all the time. Once items are placed off-grid it can be very difficult to reset them back onto the grid. Off-grid placement can result in wires looking as though they are joined when in fact they are not and so causing netlisting errors that can be hard to track down.

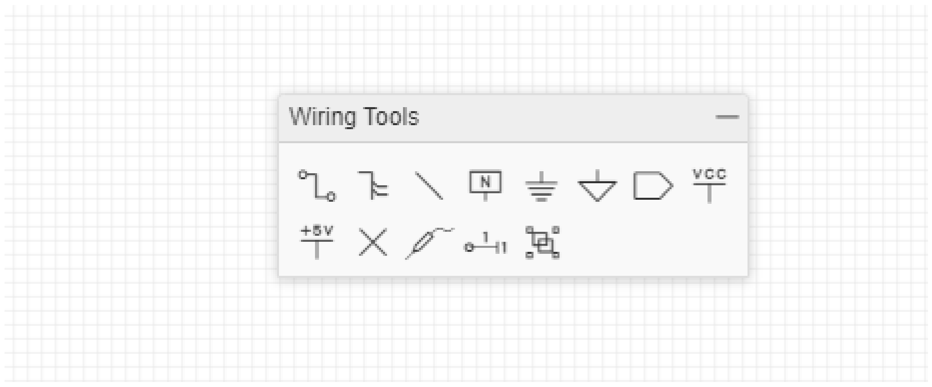
If you need to draw detailed parts of new symbols or footprints that need to go between grid points, try to reduce the grid spacing to draw these elements and then reset the grid back to your chosen default value as soon as you have completed that part of the drawing. Setting Snap=No should only really be used as a last resort.

- **ALT Sanp**: Snap size when pressing the `ALT` key.

Wiring Tools

If you have hidden your tools , you can open them from here:
Top toolbar **Top Meun** > **View** > **Wiring Tools...**

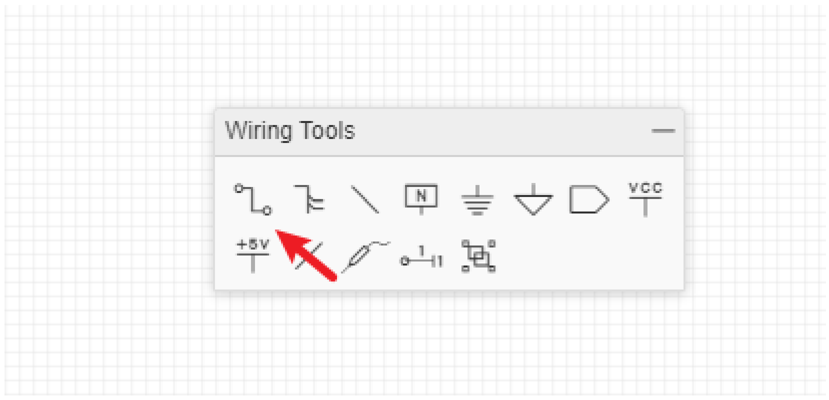




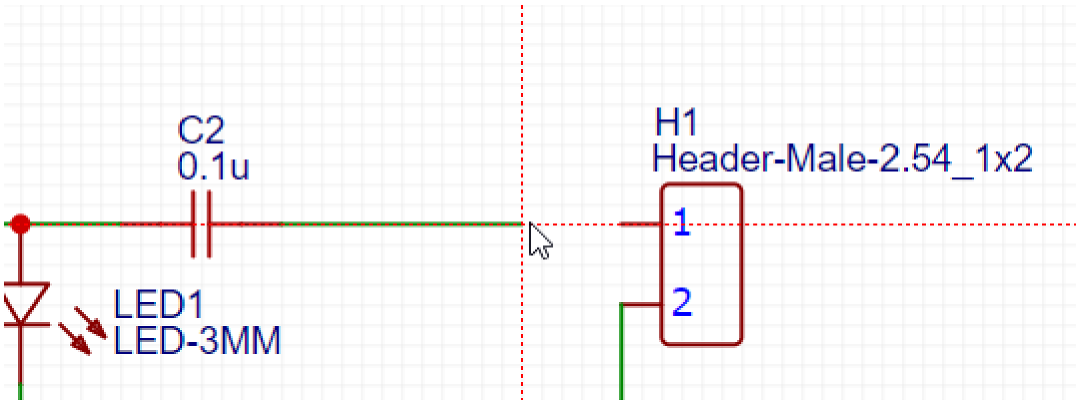
Note: *All of the commands in Wiring Tools are electronics related. Don't use a wire when you just need to draw a line, shape or an arrow: use Drawing Tools instead.*

Wire

There are three ways to enter the wire mode in EasyEDA.

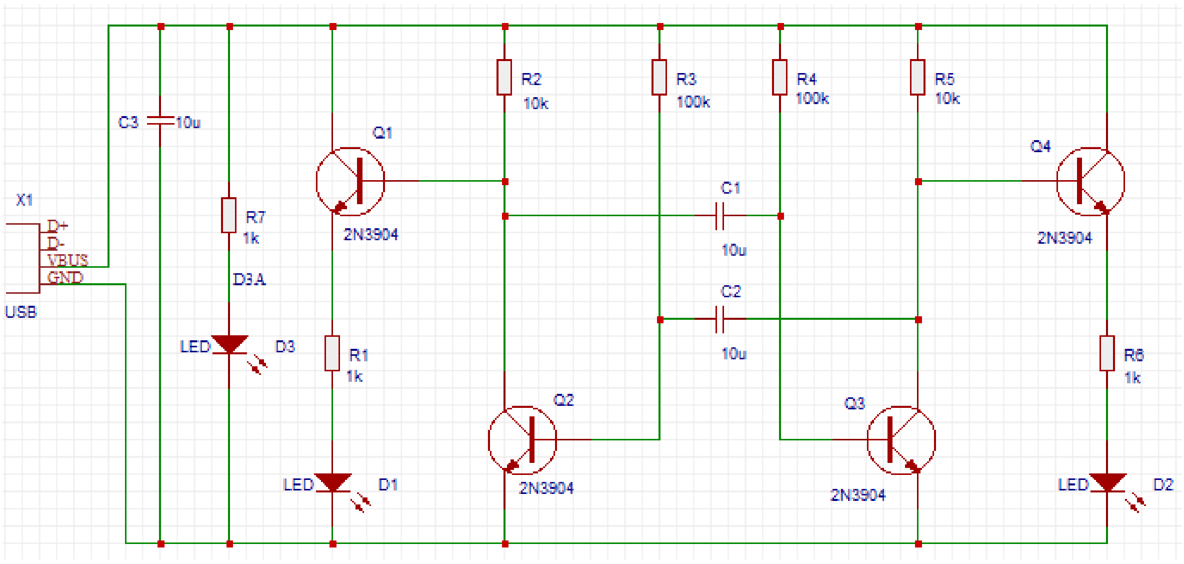


1. Click the **Wire** button from the **Wiring Tools** palette.
2. Press the `W` hotkey.
3. Click on the end of a component pin (where the grey pin dot appears if you select the component):



EasyEDA automatically enters **Wire** mode.

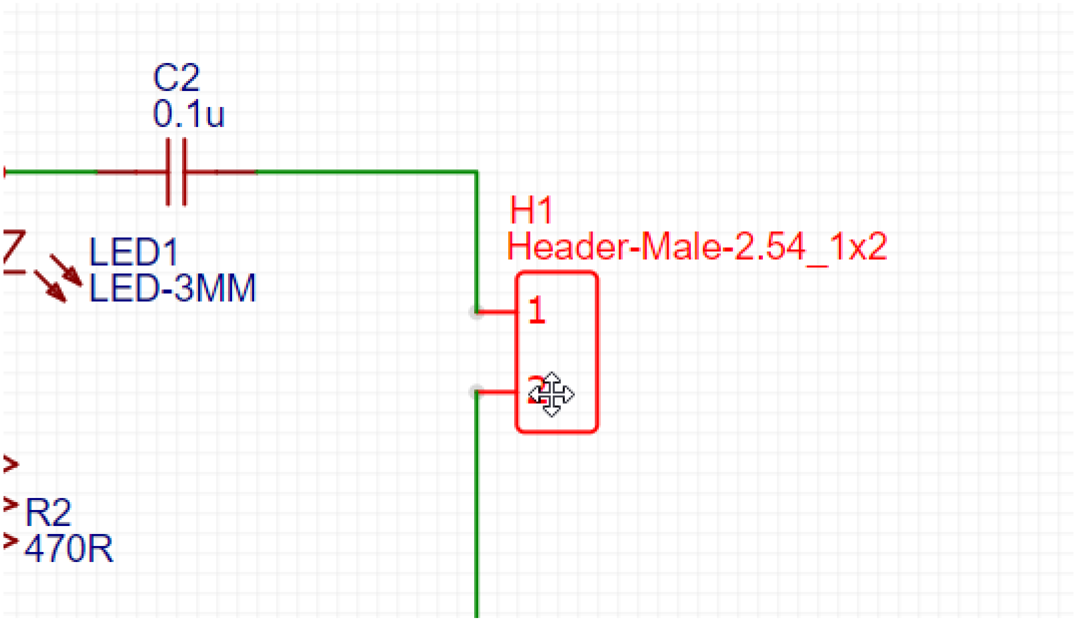
Here is a screenshot of the **Astable Multivibrator LED project schematic** after wiring:



Moving Components and Wires:

If you place a component, such as a resistor, on top of a wire then the wire breaks and reconnects to the ends of the component.

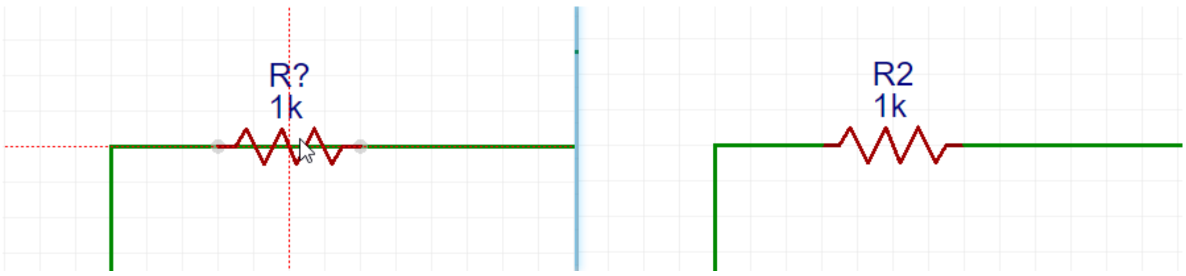
When moving selected components using the mouse, they will drag attached wires with them (“rubber band”) to some extent but please be aware that the rubber banding feature has some limitations. When moving selected components most wire will move vertically and horizontally. Using the arrow keys will not rubber band. Selected wires do not rubber band.



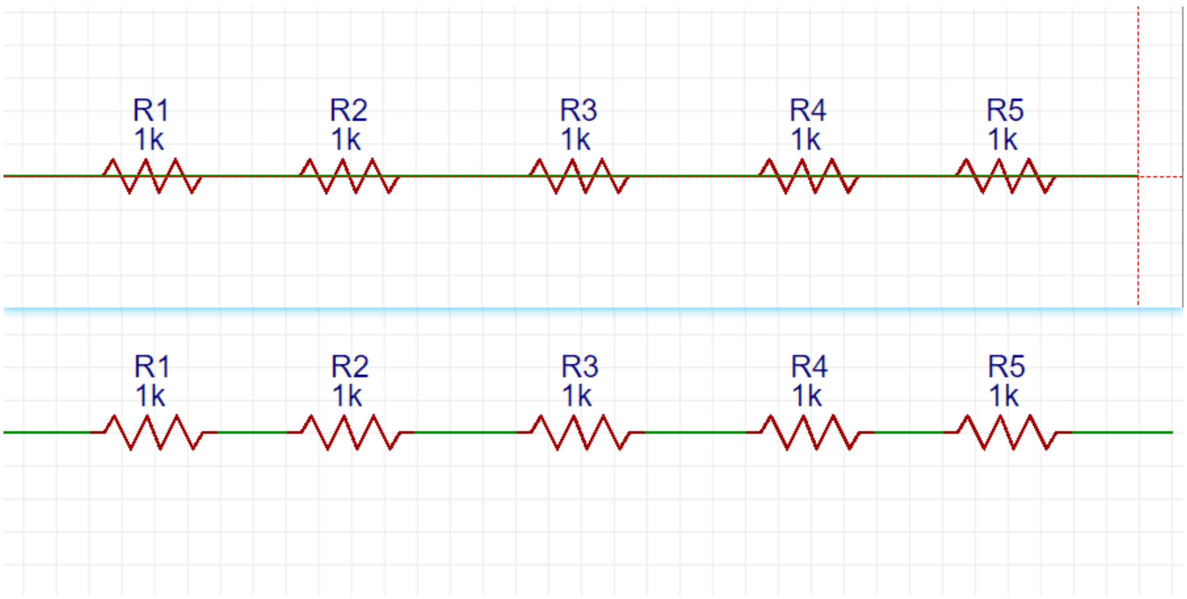
A selected wire can be moved directly by clicking on it using the mouse or by the arrow keys. If a wire is selected by clicking on it using the mouse then green grab handles will appear at the ends and vertices.

Auto adjust connection

If you put a resistor or capacitor on a wire, the wire will auto connect the pins as below:

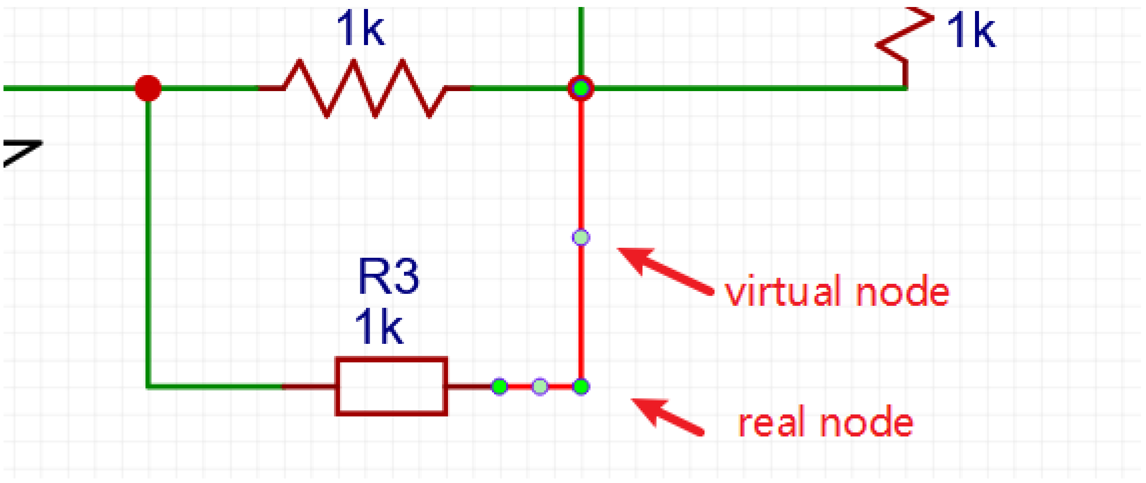


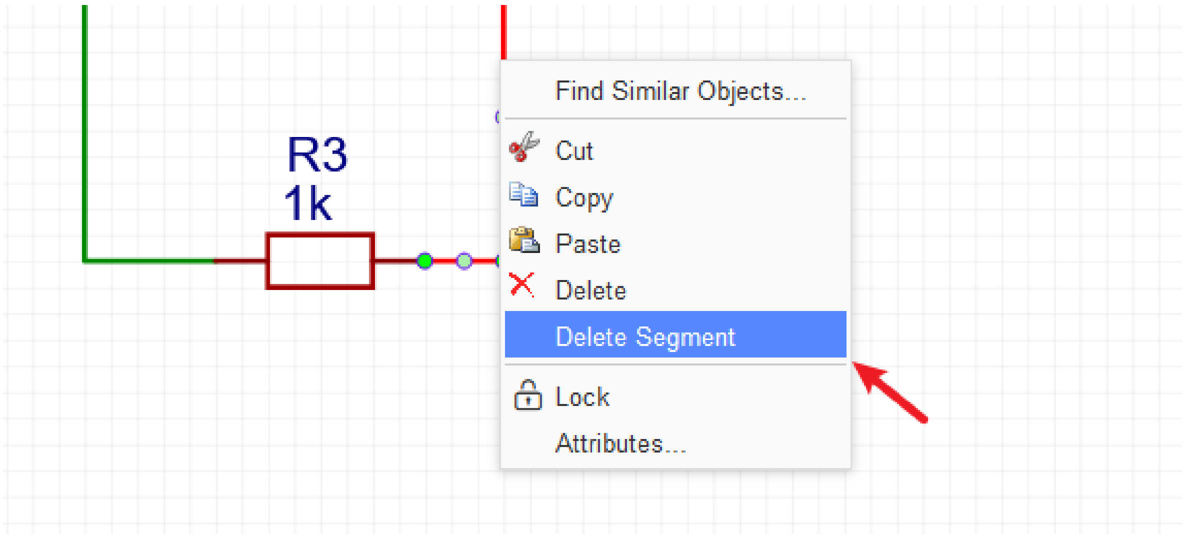
When you want to wiring a series of resistors which are in a row, you can just wire through them, and then you will find they all be connected.



Wire Node

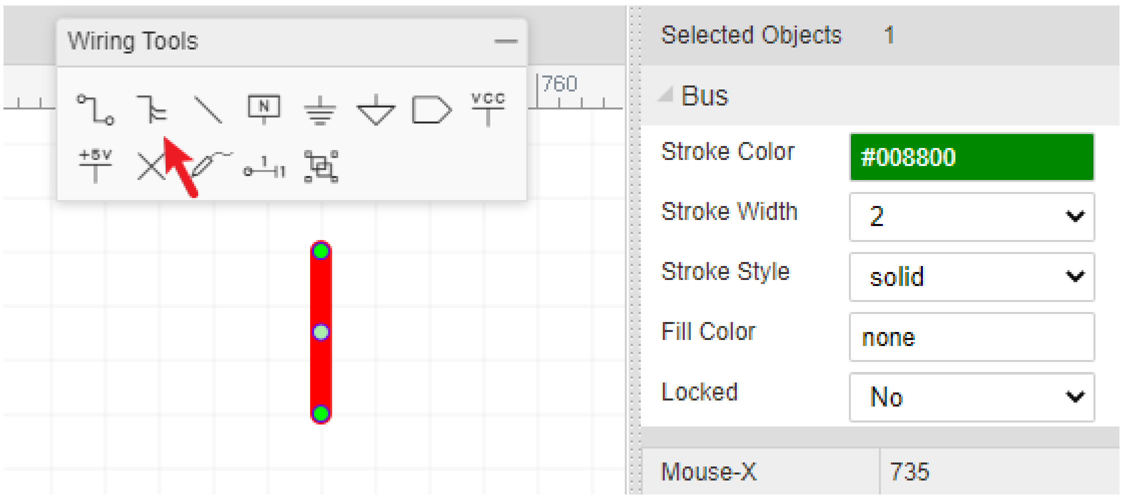
When you click on the wire, you can see the nodes on the wire, where the white is the virtual node, the red is the real node, drag the virtual node to generate the real node, and right-delete the line segment is to delete the line segment between the real nodes.





Bus

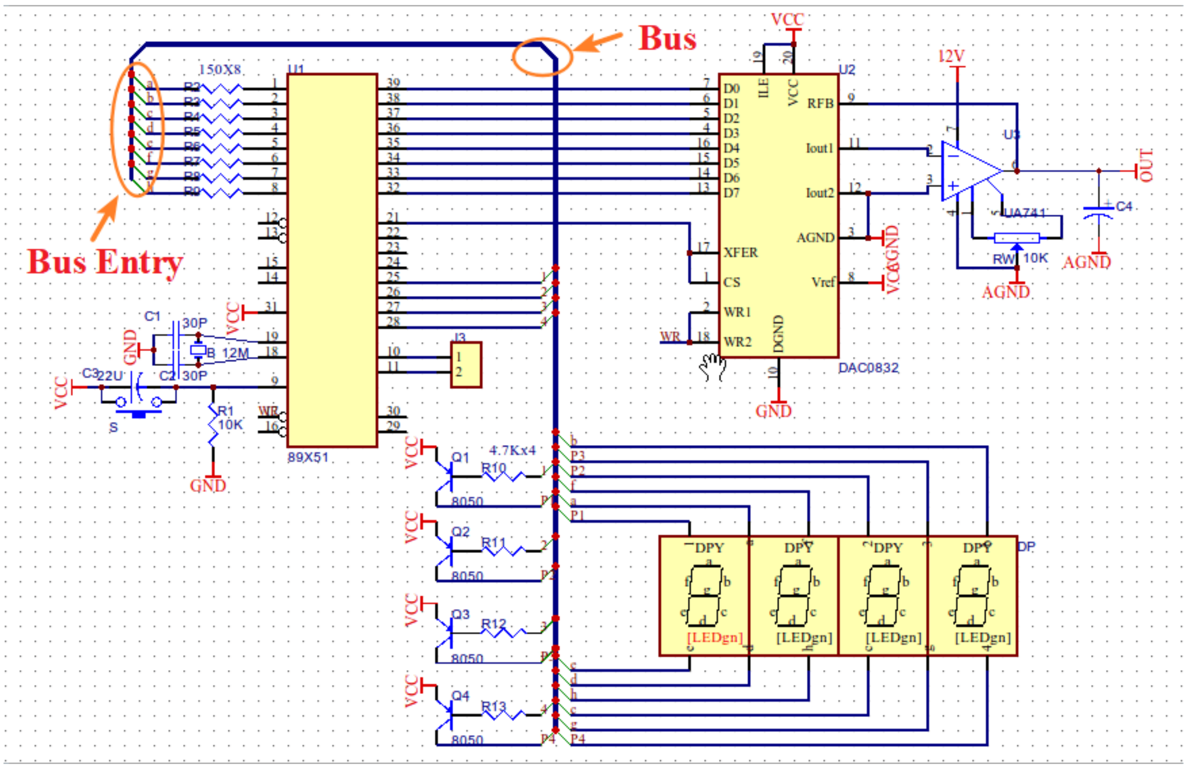
When you design a professional schematic, perhaps it will use a lot of wires. If you wiring one by one, much time would be wasted, and then you need to use `Bus`.



Selected Objects 1

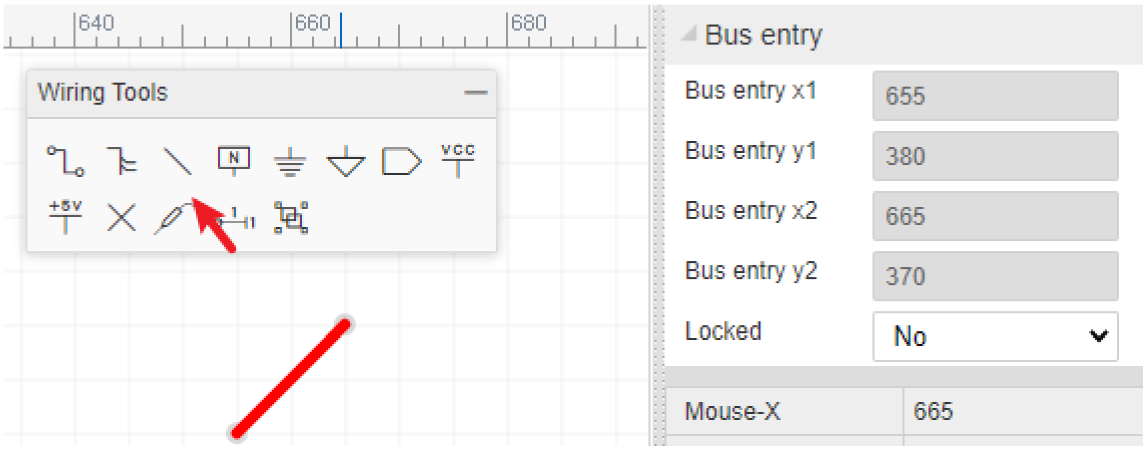
Bus

Stroke Color	#008800
Stroke Width	2
Stroke Style	solid
Fill Color	none
Locked	No
Mouse-X	735



Bus Entry

If you decide to wire with `Bus` , the `Bus Entry` must connect to Bus and other nets with wires. such as in the above image.



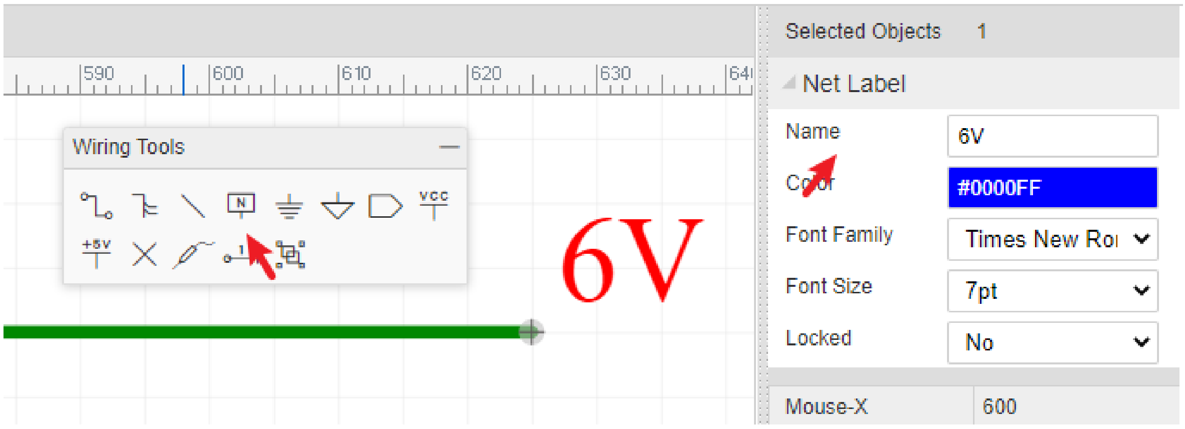
Bus entry

Bus entry x1	655
Bus entry y1	380
Bus entry x2	665
Bus entry y2	370
Locked	No
Mouse-X	665

The "Bus" and "Bus Entry" just for the indication, because when you place Bus and Bus Entry, you have to place the netlabel on the Bus Entry dot point.

Net Label

NetLabel can be used to give your wires names to help you find them and identify any misconnections. You can find the **NetLabel** from the Wiring Tools palette or by using the `N` hotkey. When selecting the netlabel, you will find its attributes in the right hand Properties panel:



Selected Objects 1

Net Label

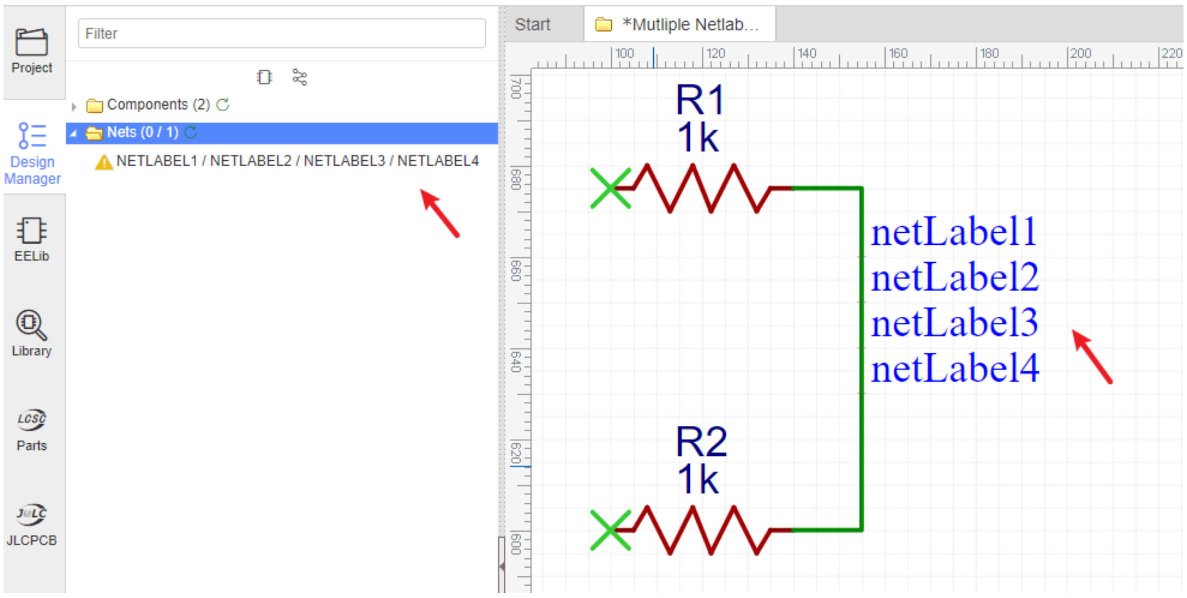
Name	6V
Color	#0000FF
Font Family	Times New Ro
Font Size	7pt
Locked	No
Mouse-X	600

You can change its name and colour. If you only want to change its name, it may be easier to just double click the netlabel.

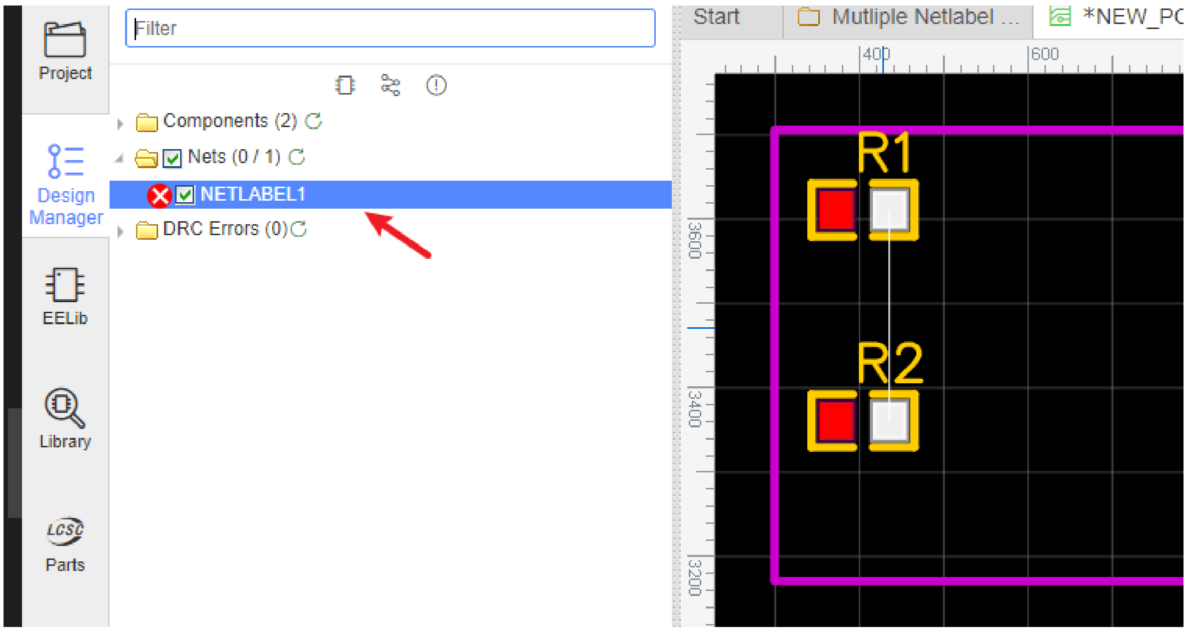
Multi-NetLabels in One Wire

EasyEDA support mutil-netlabel in one wire now.

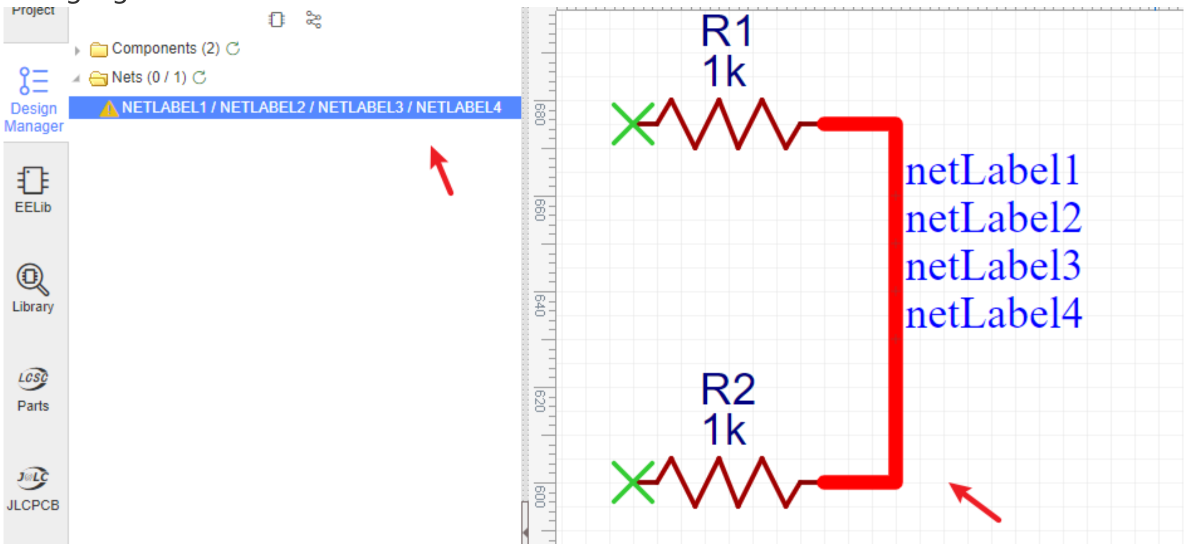
When you convert the schematic to the PCB, the editor will choose the first netlabel you placed as the net name for this wire, as below NETLABEL1.



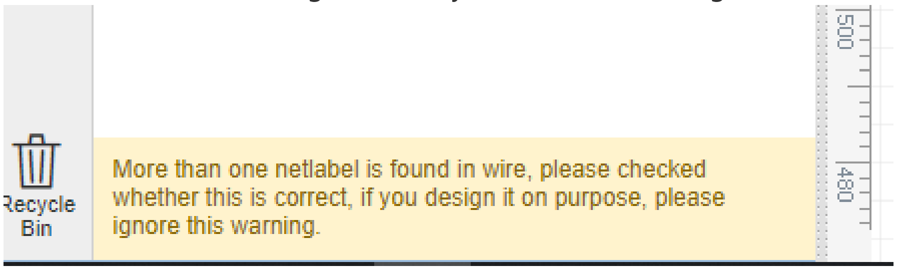
Convert to PCB:



As above image, when you click anyone netlabel's name in the design manager, the wire will be highlighted.



And check the bottom right corner, you will see a warning:



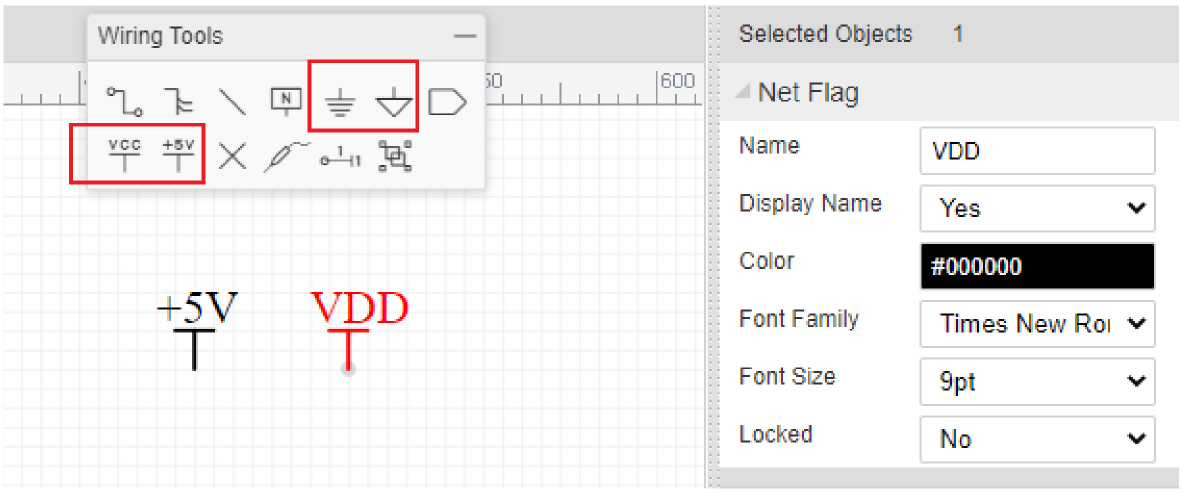
More than one netlabel is found in wire, please checked whether this is correct, if you design it on purpose, please ignore this warning.

Notice:

- *If wire 1 has 3 netlabels A B and C, and wire 2 has netlabel A, then wire 1 and wire 2 are the same net.*
- *Netlabel/Netflag/Netport/volprobe only support English characters and letters, and Arabic numerals.*
- *If a part prefix is P1, which has two pins, it will have two nets "P1_1" and "P1_2" by default, if you place a netlabel named P1_1 at other wire which is not connect with P1 pin1, the default "P1_1" will change to "P1_1(1)" for avoid the wrong connection with netlabel "P1_1".*

Net Flag

NetFlag is the same as NetLabel, you can find the NetFlag from the Wiring Tools palette or using the `Ctrl+G` hotkeys for **GND** or `Ctrl+Q` for **VCC**. You can also change its name, for example from **+5V** to **VDD**:

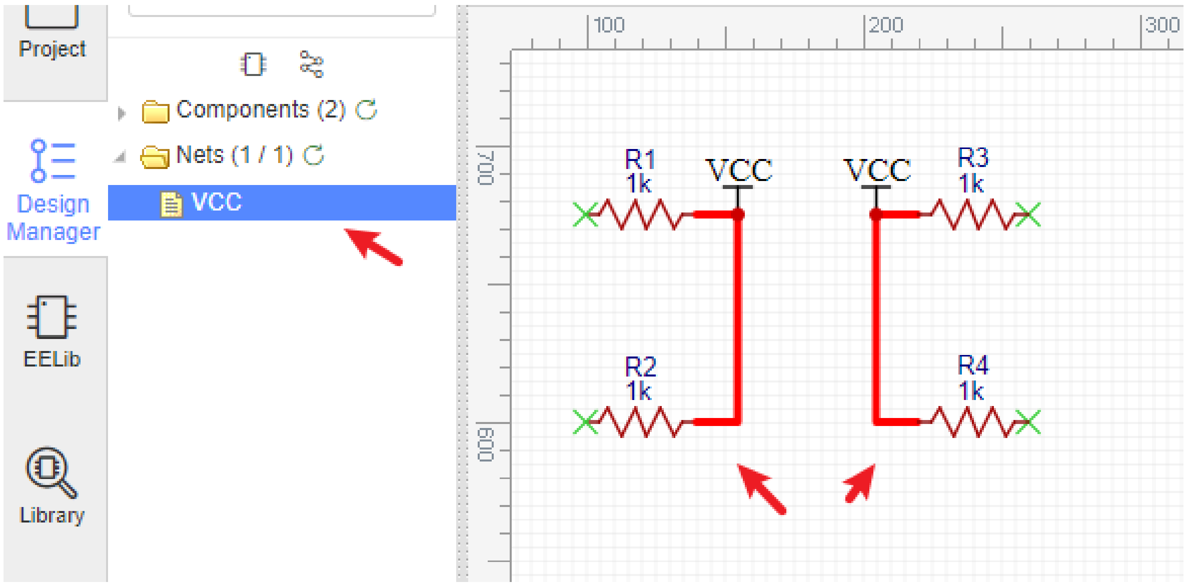


Selected Objects 1

Net Flag

Name	VDD
Display Name	Yes
Color	#000000
Font Family	Times New Roı
Font Size	9pt
Locked	No

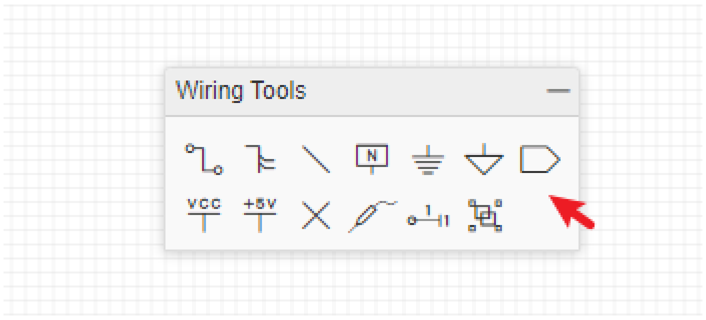
When appear two and more Netflag or Netlabels which are the same name, they will connected with each other.

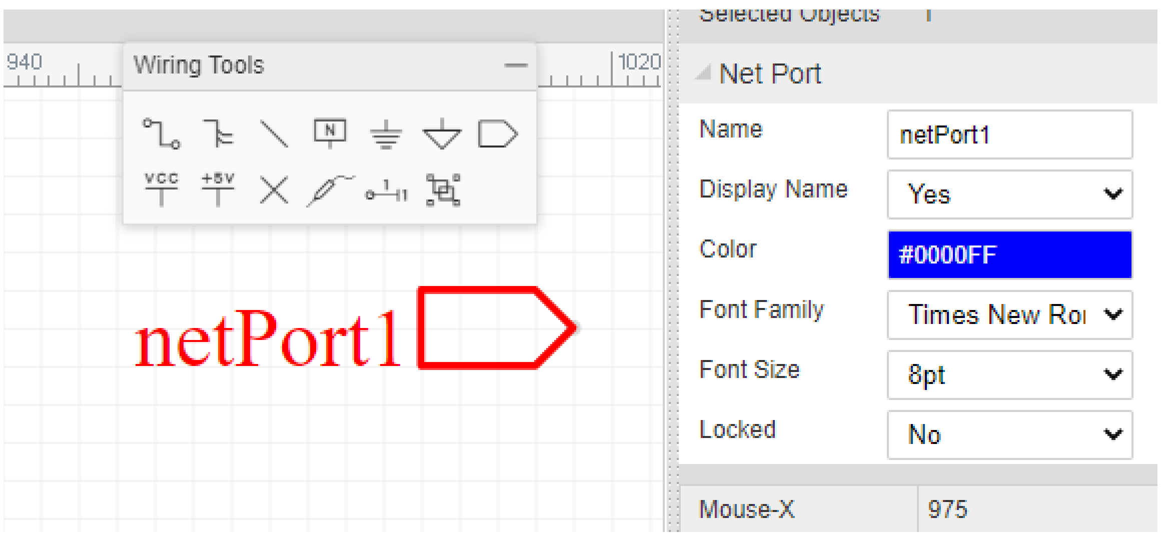


Wiring Tools palette provides NetFlag: Digital GND, Analog GND, VCC and +5V for your convenience.

Net Port

At EasyEDA, Net Port works like Net Label, it doesn't differentiate the input and output net port. When you don't want to route too many wires, how about trying `Net Port` :



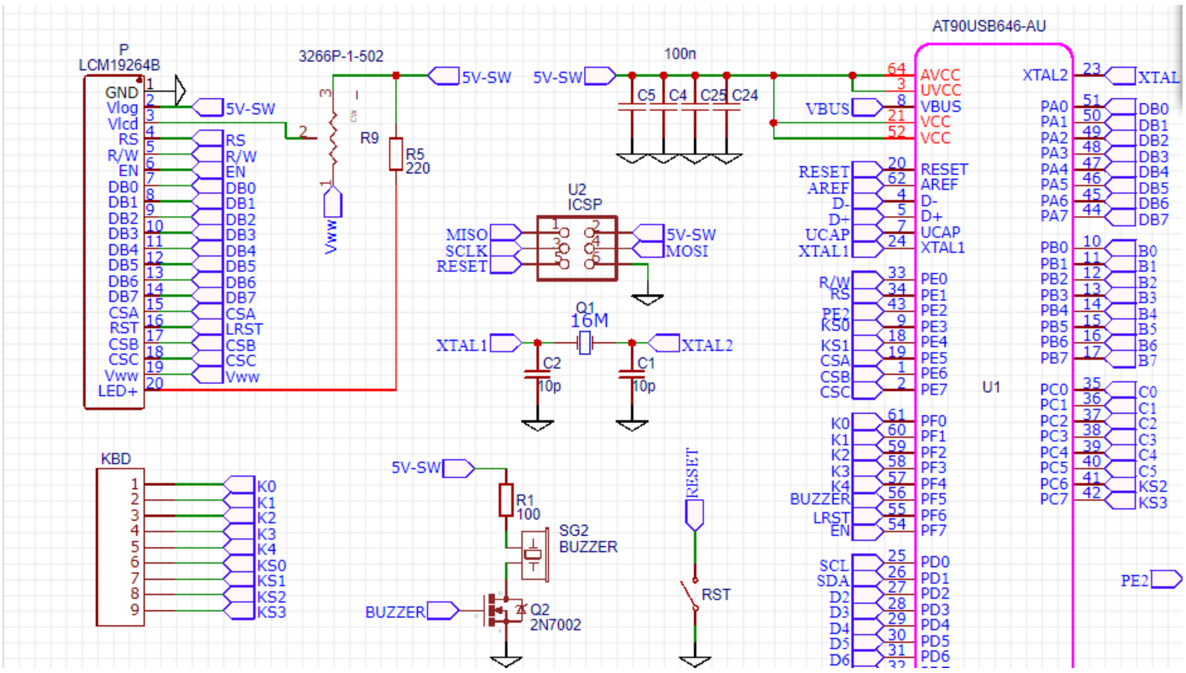


Selected Objects 1

Net Port

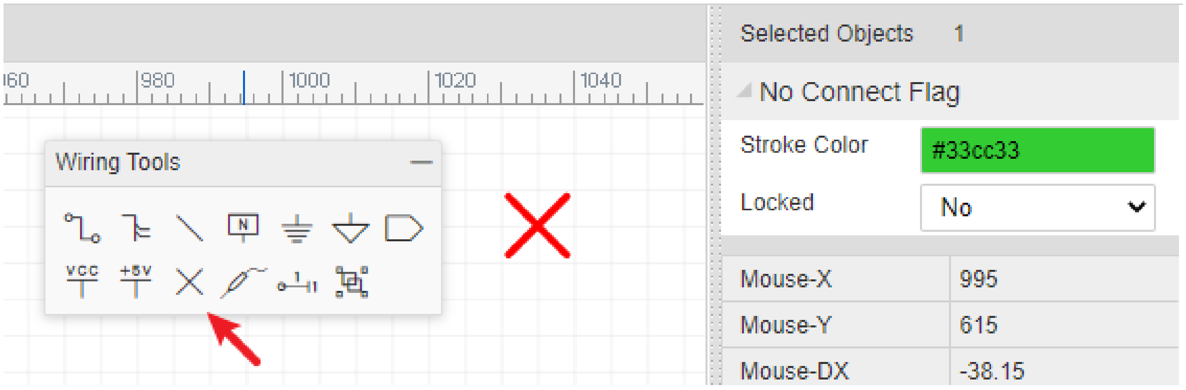
Name	netPort1
Display Name	Yes
Color	#0000FF
Font Family	Times New Roı
Font Size	8pt
Locked	No
Mouse-X	975

It will make your schematic look more clean, and you just need to set each Net Port a net name.



No Connect Flag

You can find the `NO Connect Flag` via wiring tool,

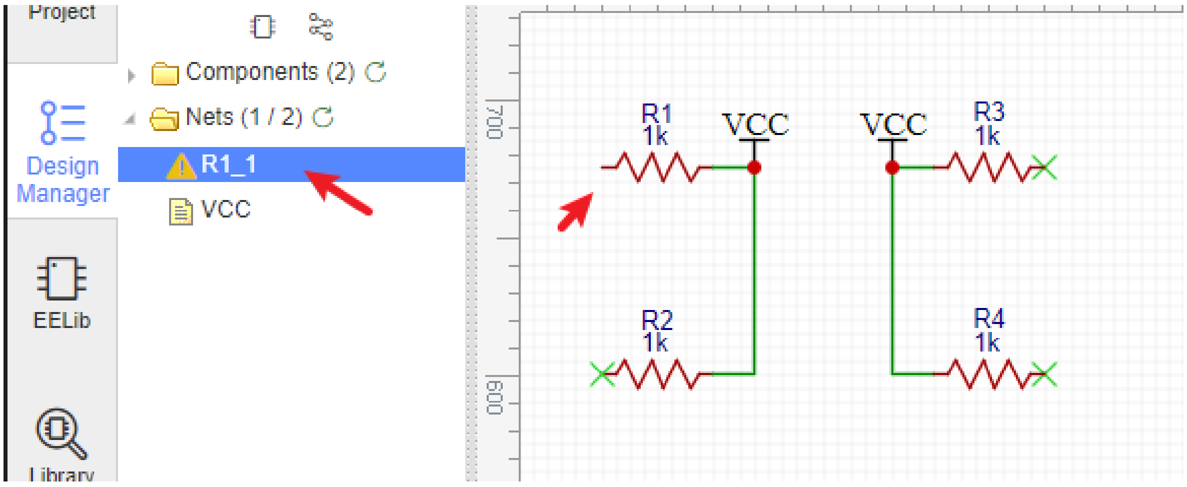


Selected Objects 1

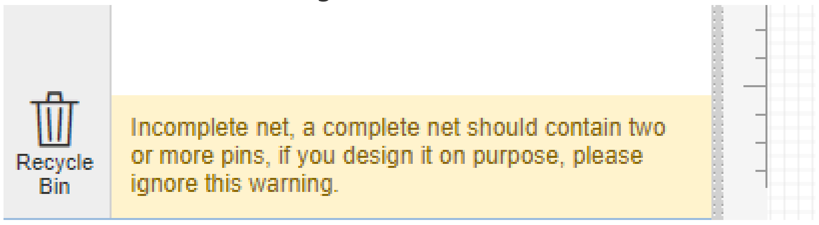
No Connect Flag

Stroke Color	#33cc33
Locked	No
Mouse-X	995
Mouse-Y	615
Mouse-DX	-38.15

In the below schematic, if you don't add a `NO Connect Flag`, there is an error flag in the nets collection of the design manager.

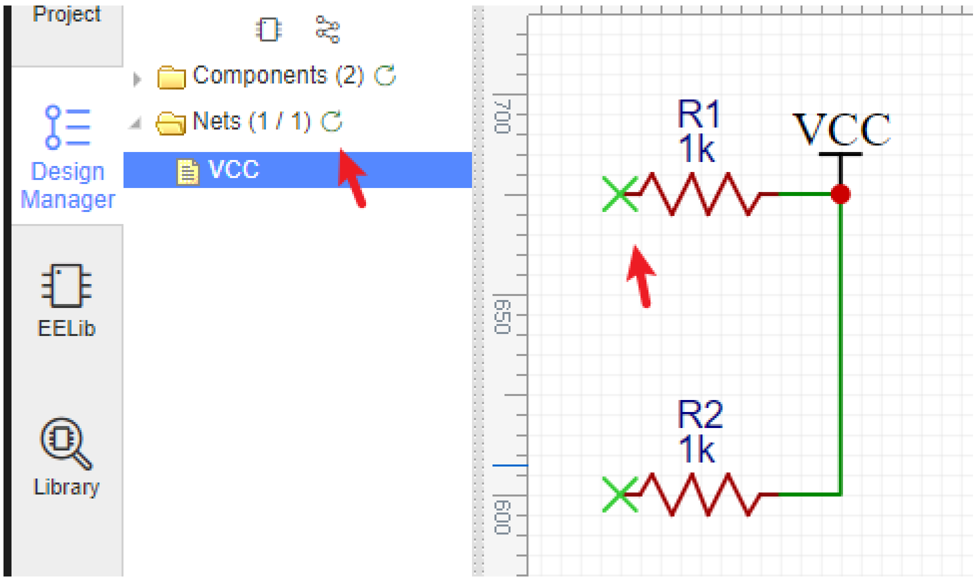


And will show the warning:



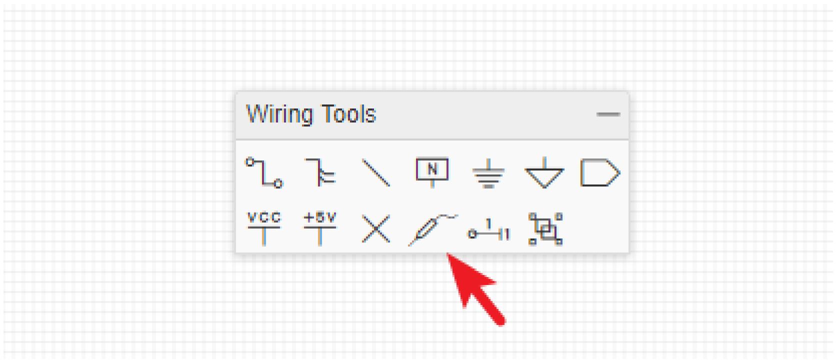
Incomplete net, a complete net should contain two or more pins, if you design it on purpose, please ignore this warning.

After adding a `NO Connect Flag`, and then refresh the Nets folder, the error disappears.

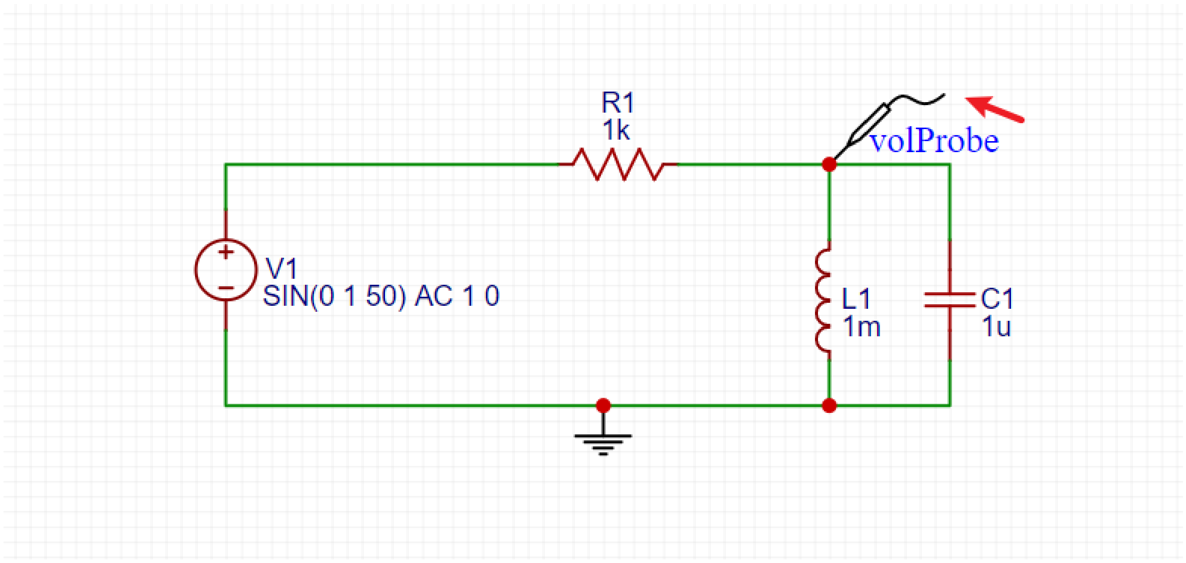


Note: *`NO Connect Flag` only works on the symbol's pin directly.*

Voltage Probe



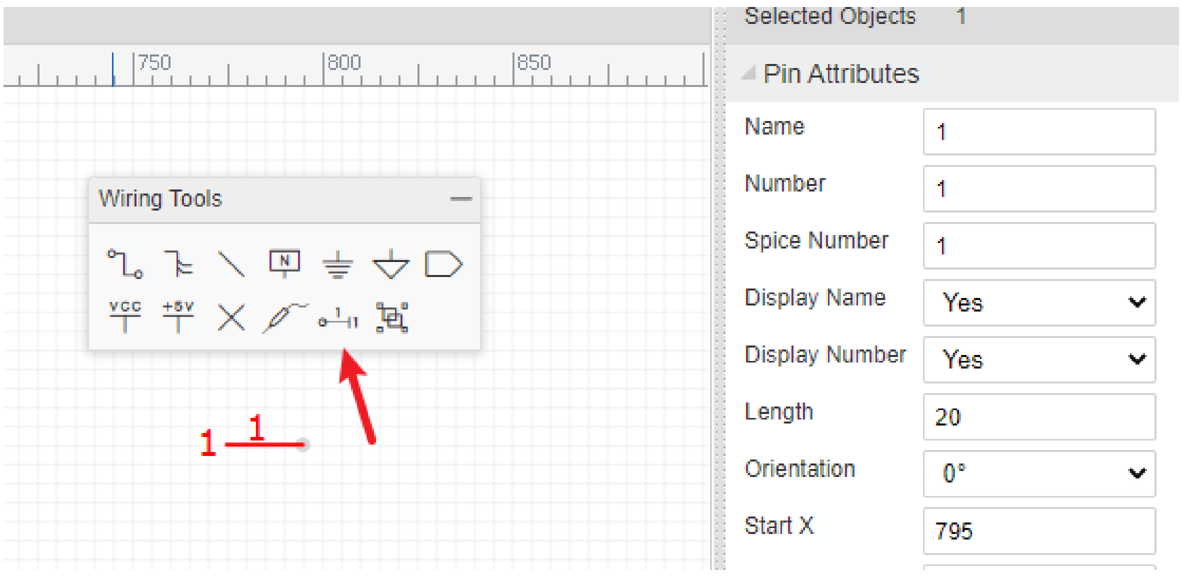
EasyEDA provides a simulation feature for the schematic. After the simulation is running, you will see the waveform where you placed the voltage probes in the circuit.



For more detail about the simulation, please check the Simulation section.

Pin

When you create a new symbol in schematic and schematic lib, you must use `Pin` to create pins for the new symbol, otherwise your symbol can't be wired with wires.


Selected Objects 1

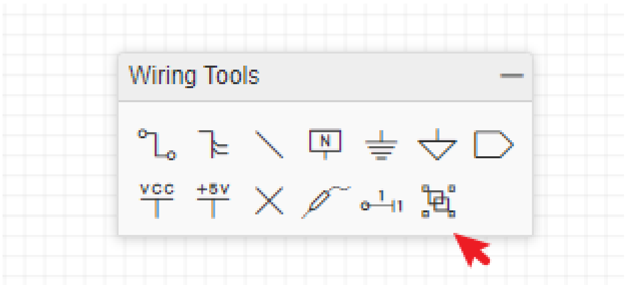
Pin Attributes

Name	1
Number	1
Spice Number	1
Display Name	Yes
Display Number	Yes
Length	20
Orientation	0°
Start X	795

For more information please refer to the **Symbol Library - Create Symbol** section.

Group/Ungroup Symbol

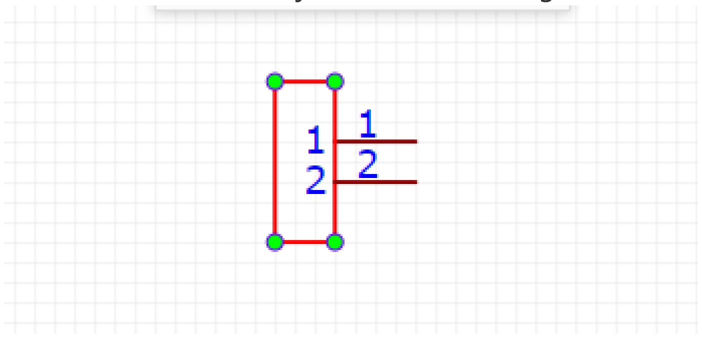
On the **Wiring Tools** palette there is the **Group/Ungroup Symbol...** button.



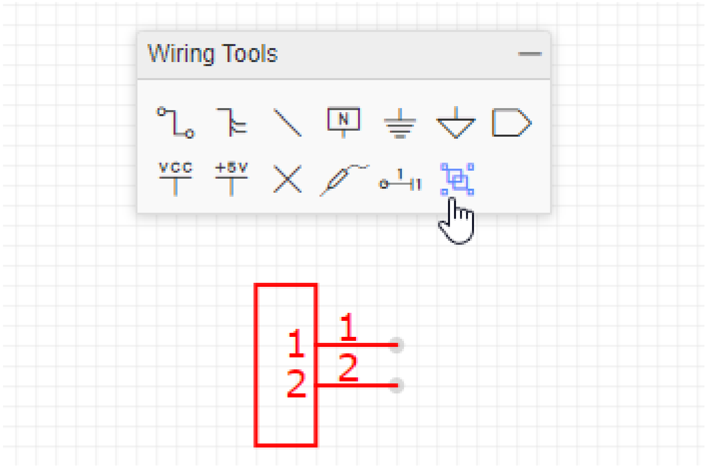
Just like the **Symbol Wizard**, this tool is also for you to quickly create schematic library symbols.

Here's how.

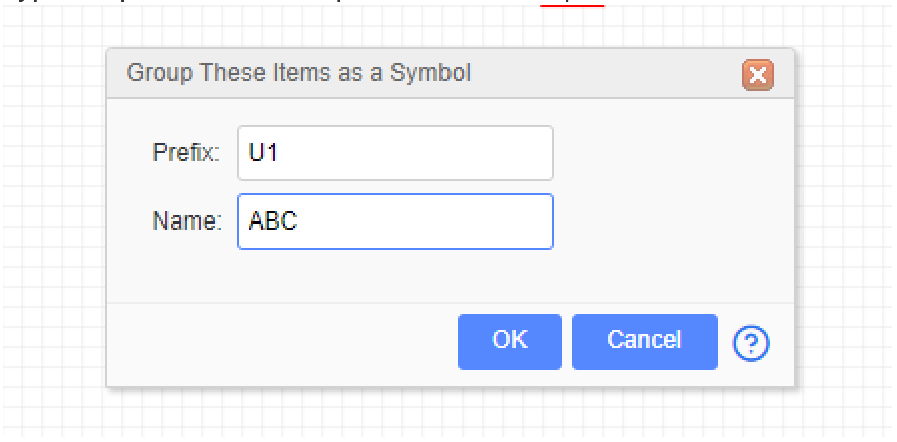
- Place Pins and other objects such as rectangle



- Select them, and click the "Group/Ungroup Symbol" icon

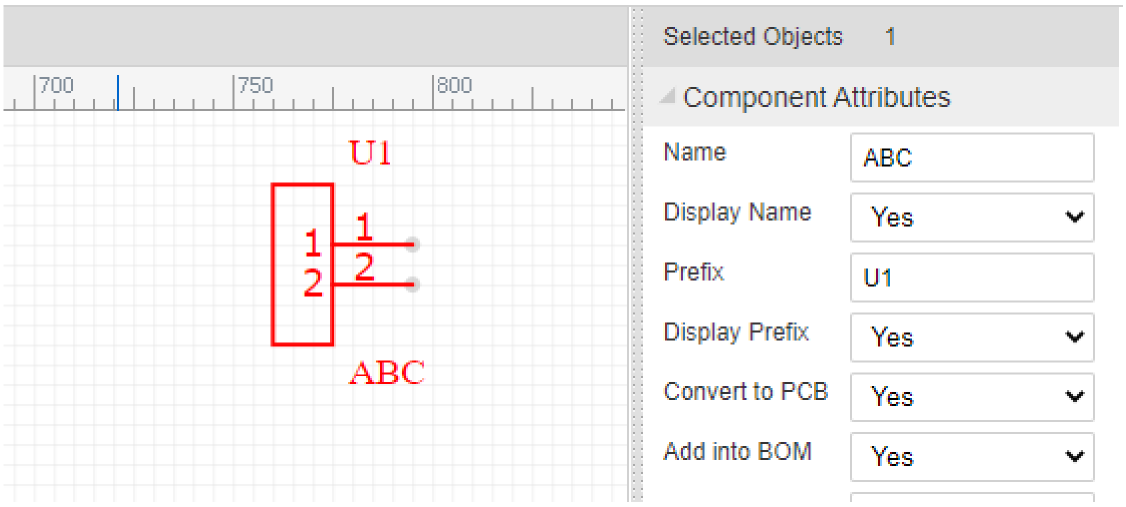


- Type the prefix and name, press OK, done. A part is created.


Group These Items as a Symbol

Prefix: U1

Name: ABC


Selected Objects 1

Component Attributes

Name	ABC
Display Name	Yes
Prefix	U1
Display Prefix	Yes
Convert to PCB	Yes
Add into BOM	Yes

So what does Ungroup do? Try selecting a symbol and then click the Group/ungroup command to see what happens!

Note:

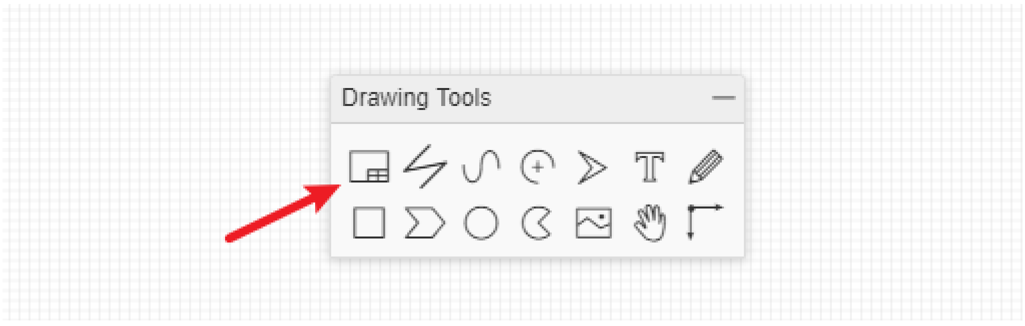
- The symbol you created in the schematic will not be saved in the personal libraries, if you want to use it repeatly, please create a Symbol via: Top Menu - File - New - Symbol.

Drawing Tools

Sheet Setting

It is now possible to add design notes to the frame and the frame selection, for example A4, which can assist in aligning and improve the look of printed schematics and PCB designs.

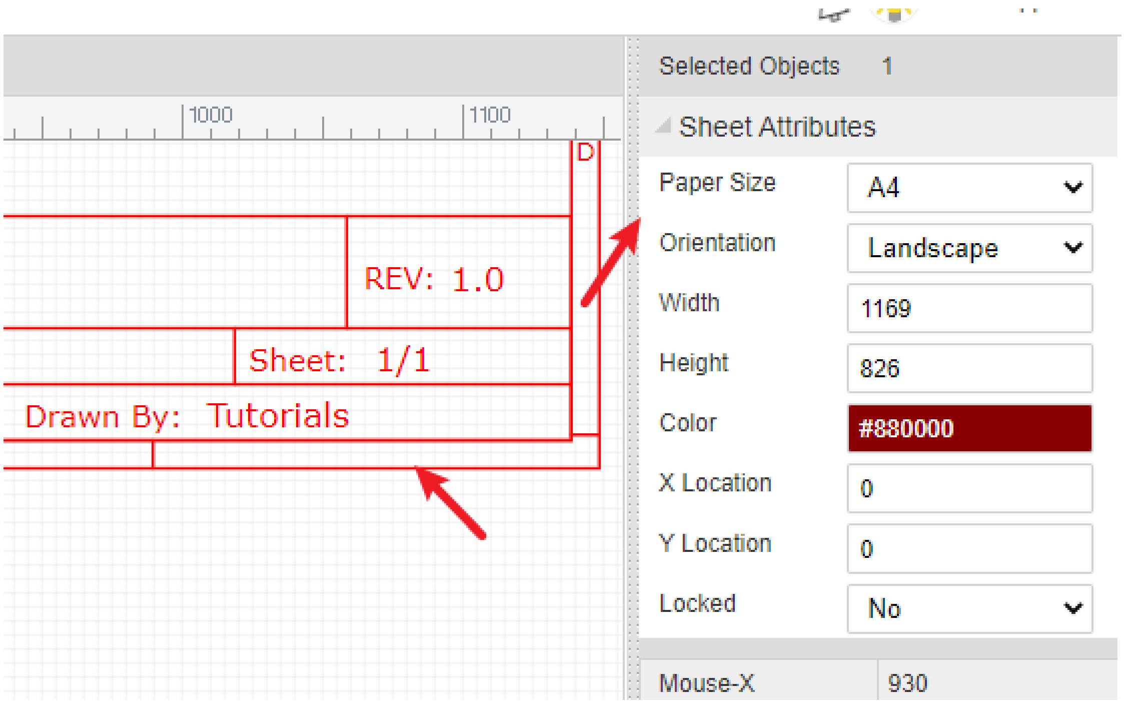
Click the frame/drawing/document button like in the image below:



And you can edit the blue text when you've selected the text attributes or double clicked it.

The bottom right zone can be selected and dragged or the frame can be dragged and deleted.

When you've selected the bottom right zone, you can edit the sheet attributes:



Selected Objects 1

Sheet Attributes

Paper Size	A4
Orientation	Landscape
Width	1169
Height	826
Color	#880000
X Location	0
Y Location	0
Locked	No

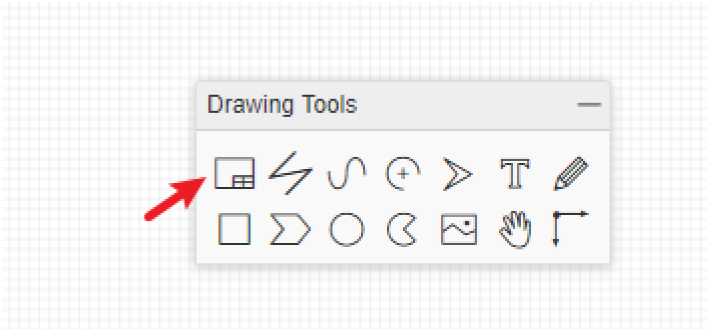
Mouse-X 930

Custom Sheet

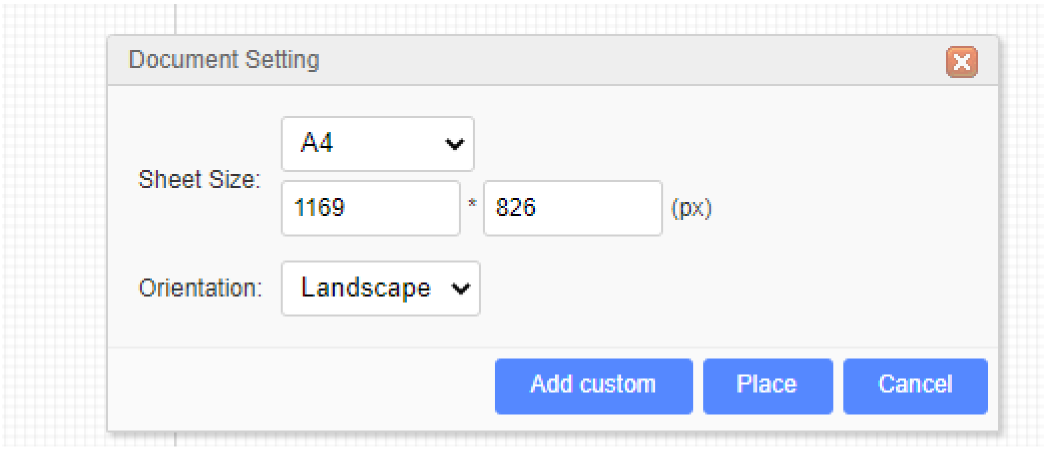
EasyEDA supports the schematic diagram drawing frame required by custom. At present, custom drawings need to be placed manually, and automatic reference of custom drawings is not supported when creating new schematic diagram.

How to create:

1. Click the "Sheet Setting" button at "Drawing Tool".



2. Click "Add Custom" button.


Document Setting

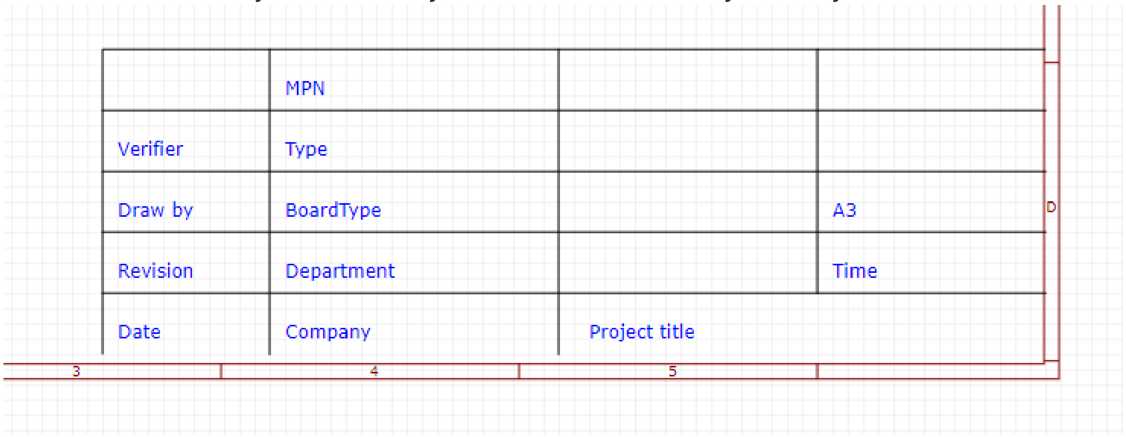
Sheet Size: A4

1169 * 826 (px)

Orientation: Landscape

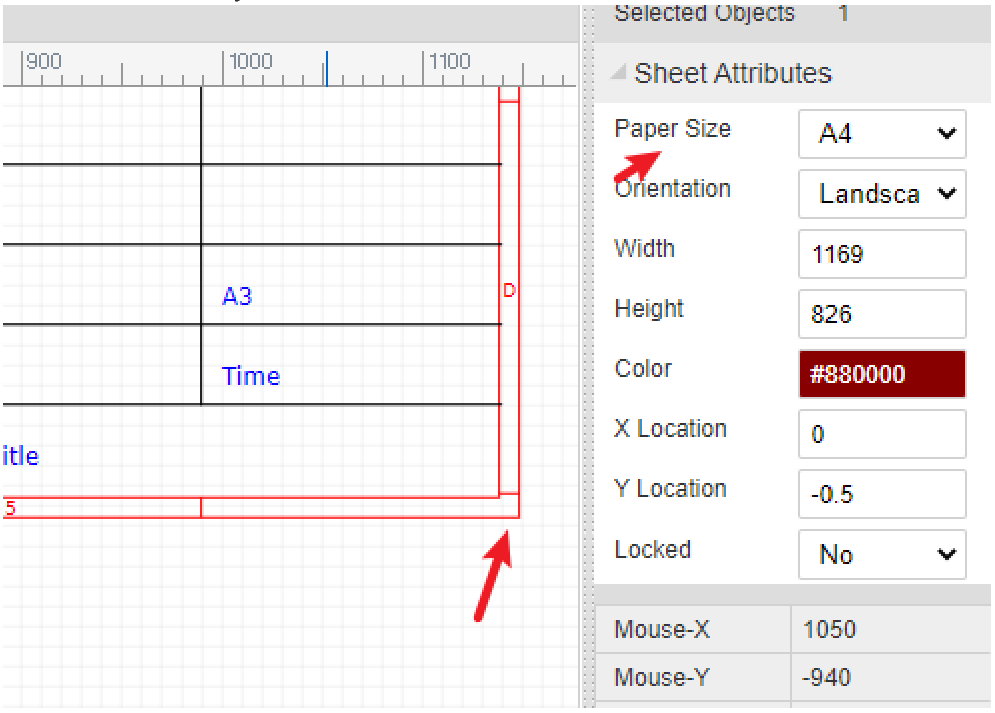
Add custom | Place | Cancel

3. It will create a new symbol editor, you can edit the table by line as you want, as below:



	MPN		
Verifier	Type		
Draw by	BoardType		A3
Revision	Department		Time
Date	Company	Project title	

4. Select the outline, you can edit its size.



Selected Objects 1

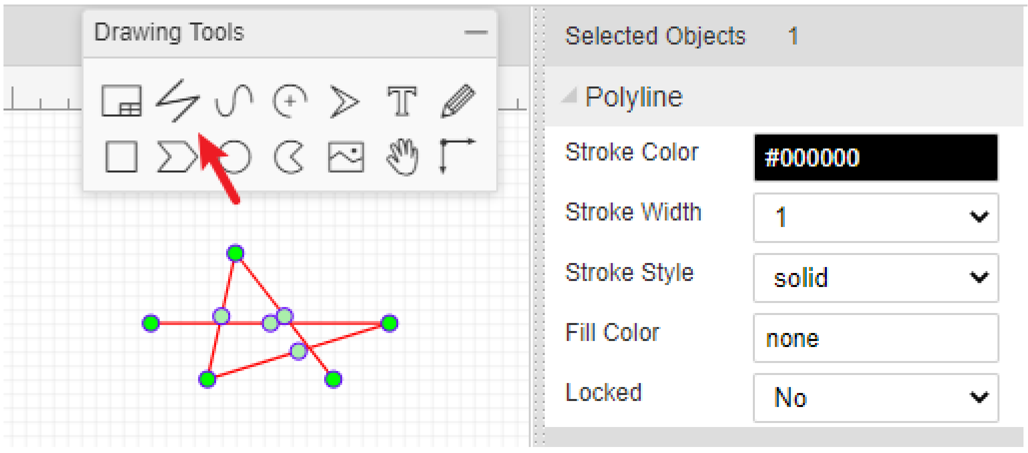
Sheet Attributes

Paper Size	A4
Orientation	Landsca
Width	1169
Height	826
Color	#880000
X Location	0
Y Location	-0.5
Locked	No
Mouse-X	1050
Mouse-Y	-940

5. Save it. You can place it in schematic such as a part at "Library".

Line

In the Schematic editor, you can draw a line with any direction. You can change its attribute as in the image below:



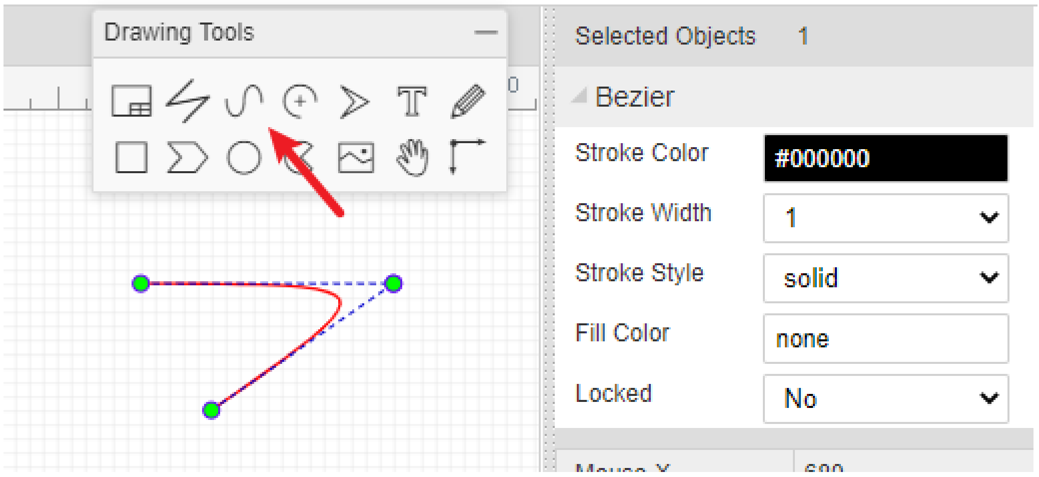
Selected Objects 1

Polyline

Stroke Color	#000000
Stroke Width	1
Stroke Style	solid
Fill Color	none
Locked	No

Bezier

With this tool, you can draw a pretty cool pattern.



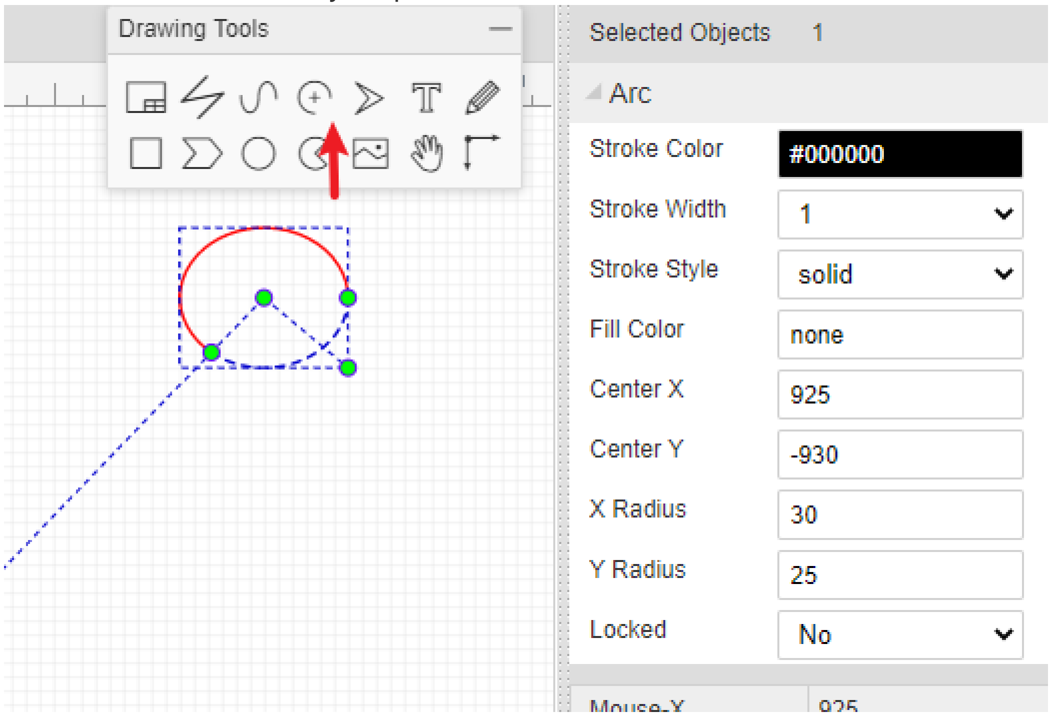
Selected Objects 1

Bezier

Stroke Color	#000000
Stroke Width	1
Stroke Style	solid
Fill Color	none
Locked	No

Arc

You can draw the arc of any shape.



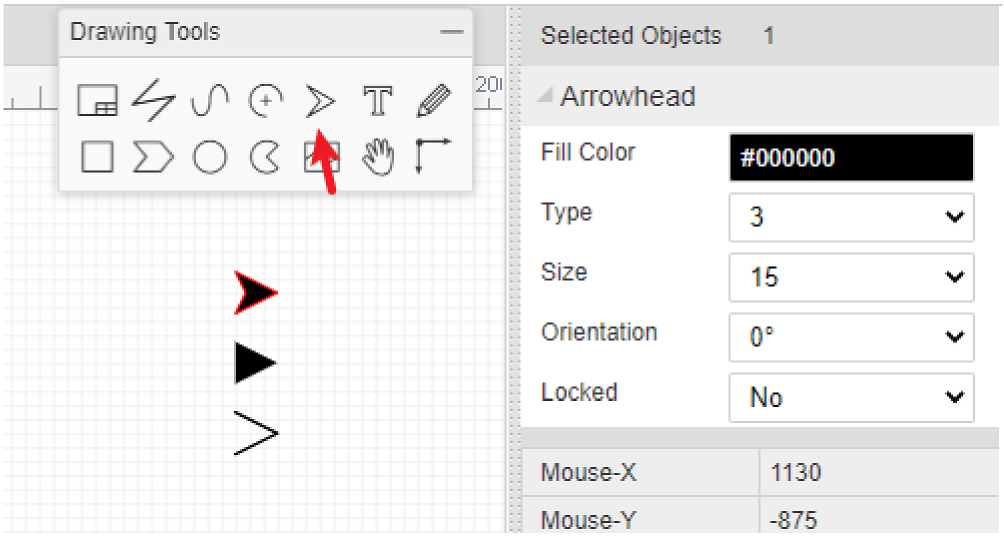
Selected Objects 1

Arc

Stroke Color	#000000
Stroke Width	1
Stroke Style	solid
Fill Color	none
Center X	925
Center Y	-930
X Radius	30
Y Radius	25
Locked	No

Arrow Head

You can add arrow head to marking text or important part.



Selected Objects 1

Arrowhead

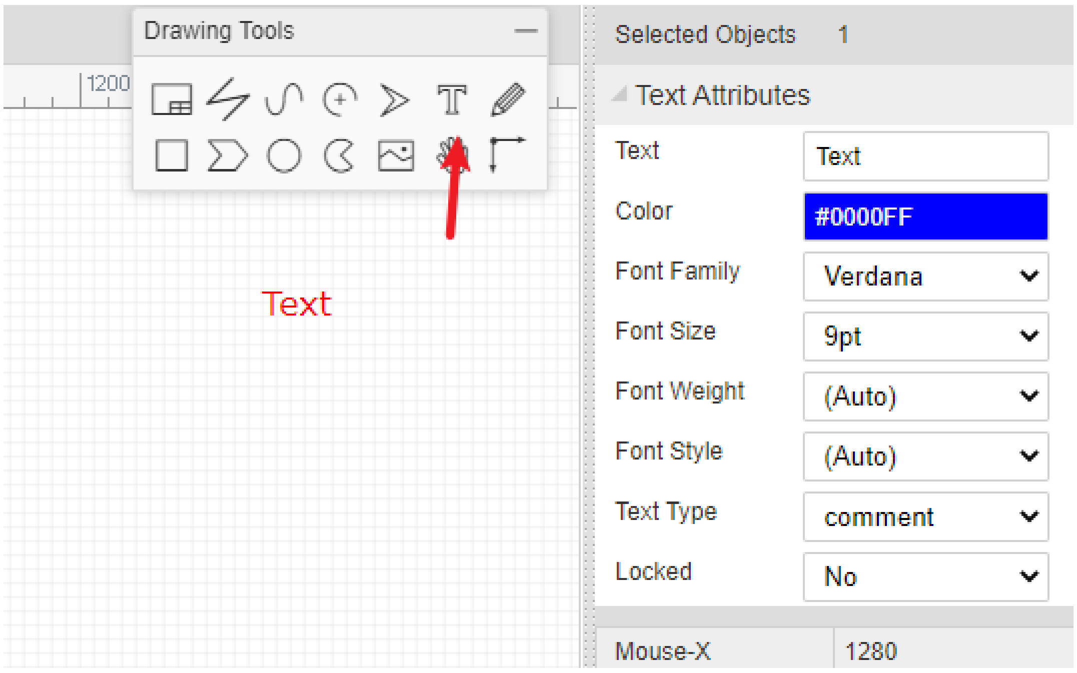
Fill Color	#000000
Type	3
Size	15
Orientation	0°
Locked	No
Mouse-X	1130
Mouse-Y	-875

Text

Text attributes provide many parameters for setting:

- **Text**: You can change text in inner box or double click the text. For every new text, the default text is `Text`.
  -**Color**: Defines text color.
  -**Font-family**: It provides 12 fonts for choosing.
  -**Font-Size**: Defines Text size.
  -**Font-weight**: Defines Text weight.
  -**Font-Style**: It contains (auto), normal, italic.
  -**Text type**: types include comment and spice.

The editor will remember your last text parameters.



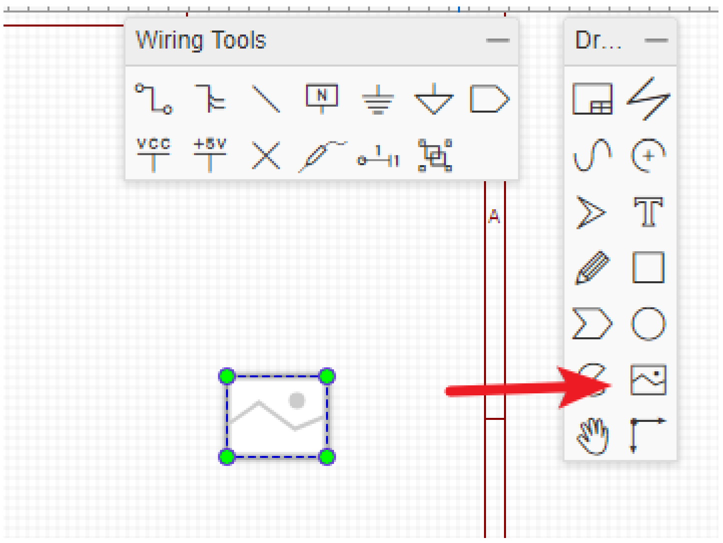
Selected Objects 1

Text Attributes

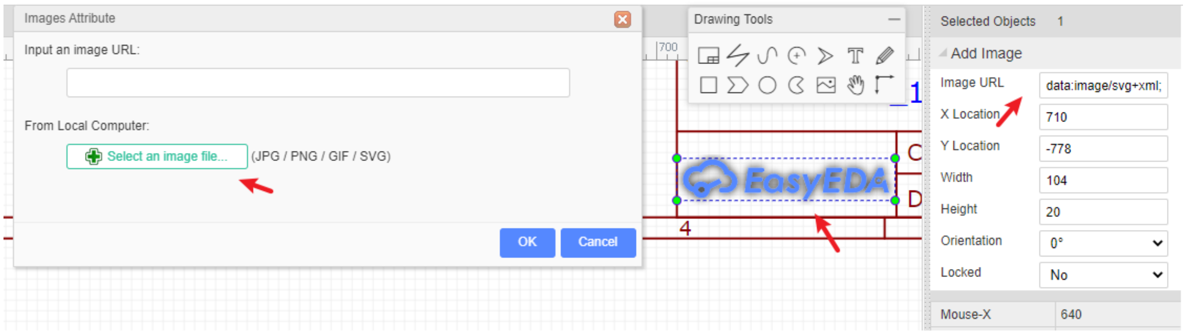
Text	Text
Color	#0000FF
Font Family	Verdana
Font Size	9pt
Font Weight	(Auto)
Font Style	(Auto)
Text Type	comment
Locked	No
Mouse-X	1280

Image

When you select Image from the Drawing Tools palette, an image place holder will be inserted into the canvas:



Select the place holder, so you can see the image's attributes in the right hand Properties panel:



Images Attribute

Input an image URL:

From Local Computer:

Select an image file... (JPG / PNG / GIF / SVG)

OK Cancel

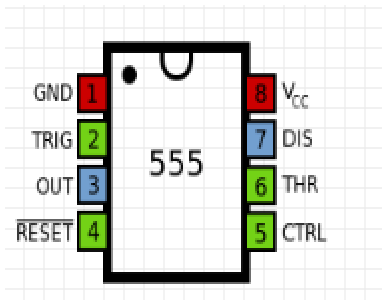
Selected Objects 1

Add Image

Image URL	data:image/svg+xml;
X Location	710
Y Location	-778
Width	104
Height	20
Orientation	0°
Locked	No
Mouse-X	640

Set the URL of your image. For example, setting the URL to:

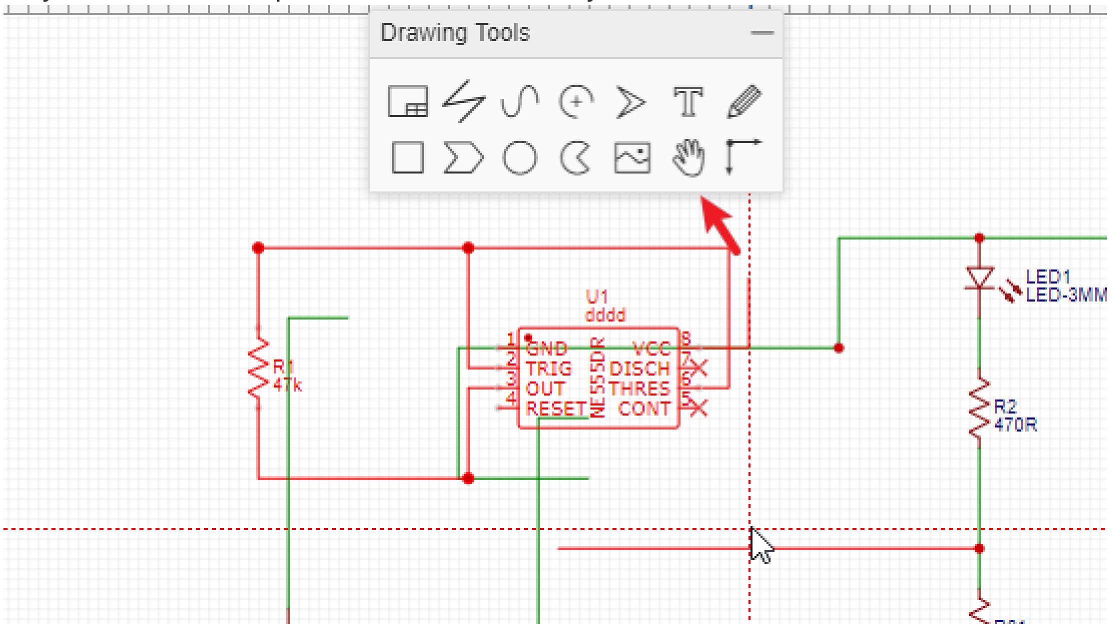
http://upload.wikimedia.org/wikipedia/commons/thumb/c/c7/555_Pinout.svg/220px-555_Pinout.svg.png will make your image look like this:



Please note: at present, EasyEDA cannot host images, so you need to upload your images to an image sharing site.

Drag

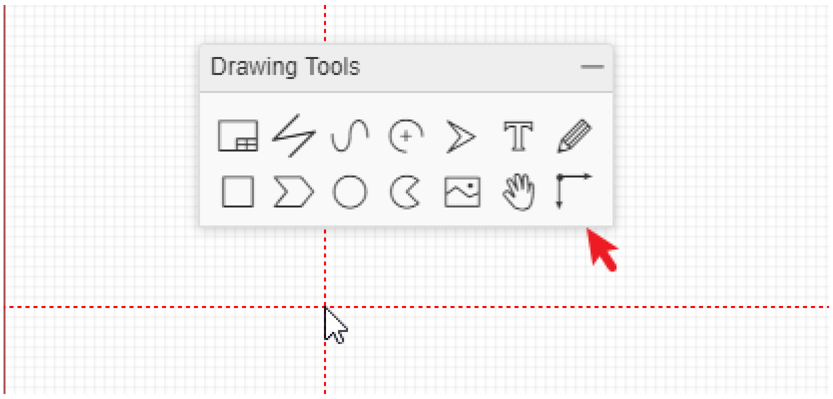
If you want to move some kind of parts and wires, you can use drag, hotkey D.
Or you can select the parts and wires area firstly and move them.



Canvas Origin

Canvas origin default is set at left top corner of the schematic sheet, but you can set it where you want via Canvas Origin.

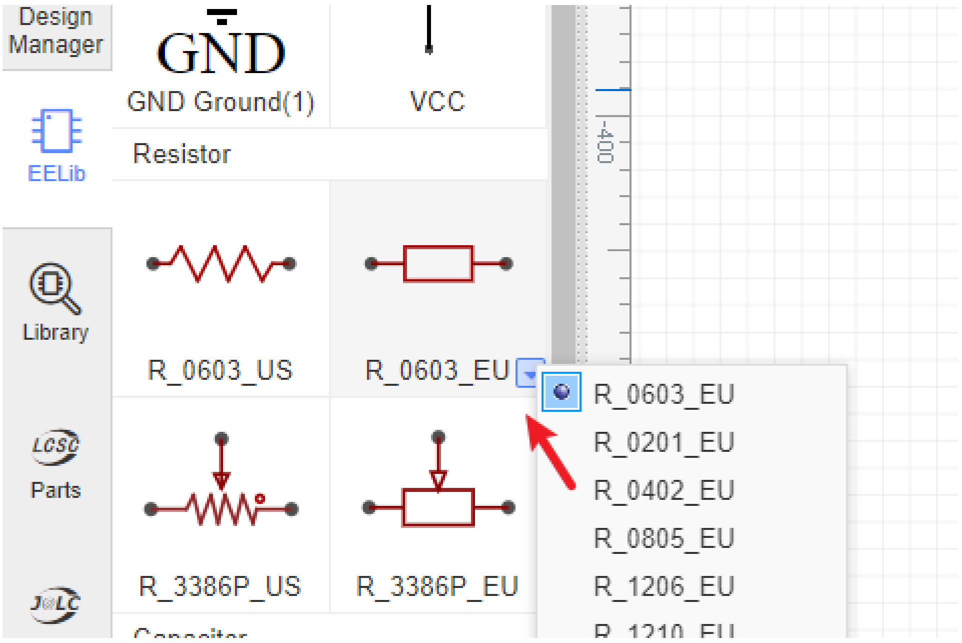
For another way to set canvas origin, you can try **Top Menu> Place > Canvas Origin**.



Libraries

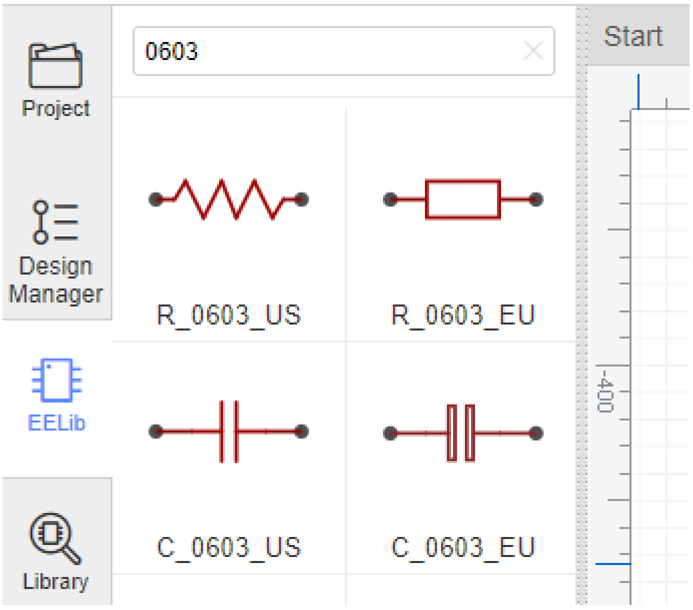
EELib

That contains ready made symbols for a wide range of components and which can be simulated.



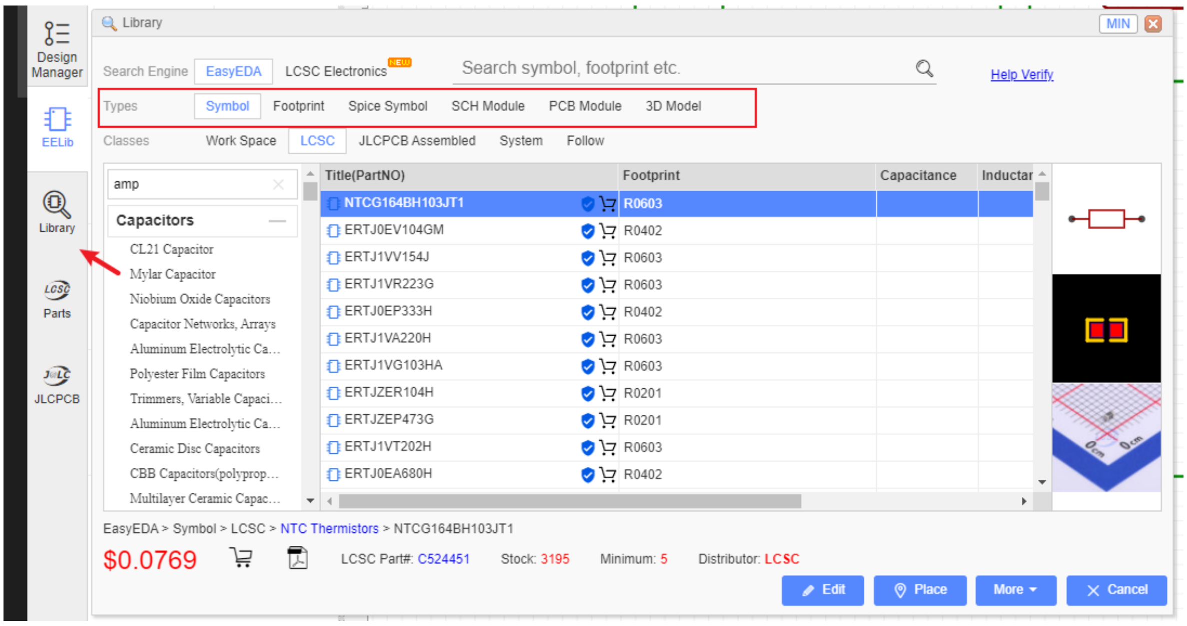
Many of these components have optional US and EU style symbols, we split them, so you can select those you like. Click on the drop down list or right click to popup the context menu, it contains many footprints or parameters. EasyEDA will remember your choices for the next time.

Don't forget to use Filter to locate a component fastly. For example, you just need to type `0603` to find all of resistors:



Library

EasyEDA provide a lot of libraries, you can find them at "Left-hand Panel - Library", hotkey "SHIFT+F", at here you can search library from LCSC, system, user contributed etc.



Title(PartNO)	Footprint	Capacitance	Inductar
NTCG164BH103JT1	R0603		
ERTJ0EV104GM	R0402		
ERTJ1VV154J	R0603		
ERTJ1VR223G	R0603		
ERTJ0EP333H	R0402		
ERTJ1VA220H	R0603		
ERTJ1VG103HA	R0603		
ERTJZER104H	R0201		
ERTJZEP473G	R0201		
ERTJ1VT202H	R0603		
ERTJ0EA680H	R0402		

EasyEDA > Symbol > LCSC > NTC Thermistors > NTCG164BH103JT1

$0.0769 LCSC Part#: C524451 Stock: 3195 Minimum: 5 Distributor: LCSC

Type

- Symbol: Schematic symbols
- Spice Symbol: Symbols for spice simlation
- Footprint: PCB footprints, PCB pattern.
- SCH Modules: Schematic modules, a part of the circuit design. It can not assign the PCB module, doesn't like the schematic Symbol can assign the footprint . when it be placed on the schematic, it will be separated.
- PCB Modules: As like as Schematic modules.
- 3D Model: It is bind with footprint via "3D Model Manager".

Classes

- Work Space: It include your personal parts and your teams' parts.
- LCSC: EasyEDA online part store LCSC.com parts(Officail Parts). It will add new libraries everyday
- LCSC Assembled: JLCPCB Assembled parts. All JLCPCB assembly parts will contain a SMT icon, that means this part can be JLCPCB assemble.
- System: EasyEDA system parts, it comes from open source libraries, such as Kicad libraries, company public libraries, user contributions.
- Follow: If you follow a user at EasyEDA(You can follow a user at him/her user page), you can view and use his/her libraries.
- User Contributed: When you searching a part, maybe you can find it at this class. At EasyEDA, all libraries are public. the detail you can refer at: Contribute

We add an "JLCPCB Assembled" Components option of the Parts, It's easy to choose which component can be assembled by JLCPCB. Yes, JLCPCB will provide the assembly service. the more information please refer at: How to order a SMT order

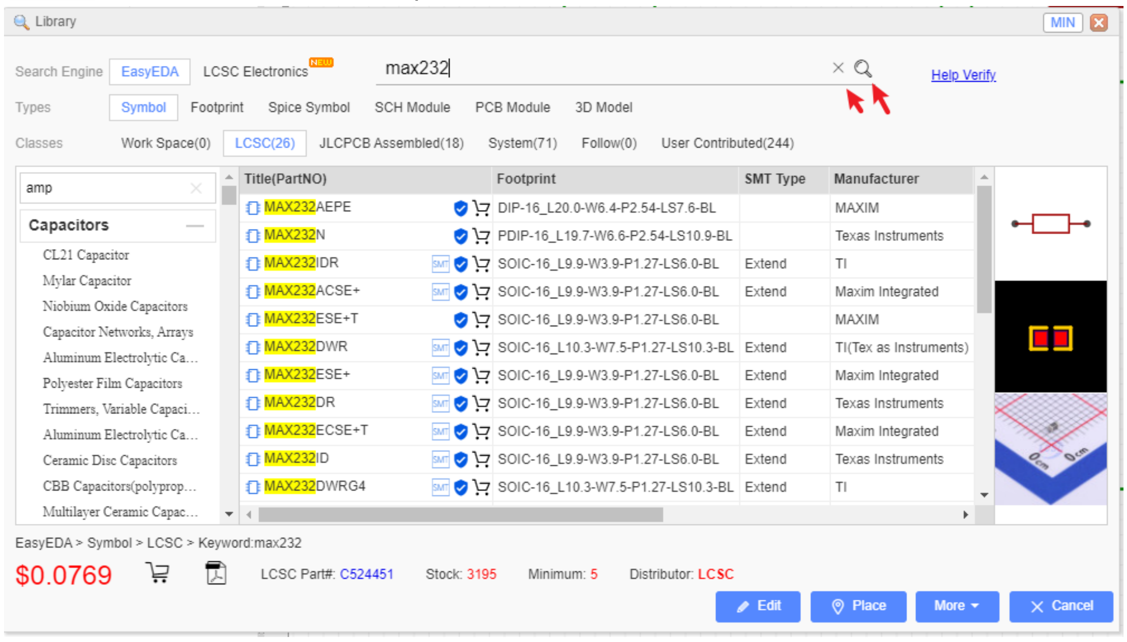
Search Engine - EasyEDA

Simply type your part number or symbol's name to Search. before searching, you must choose the "Type" first.

and then click the "Table of contents" to open the categories list to choose your components.

From there you can scroll up and down to browse parts from each category.

- If you know the component's name
  Suppose you want to find the **MAX232** (which converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits). Simply type `Max232` into the Search box and press Enter:



Library

Search Engine: EasyEDA | LCSC Electronics — max232 — Help Verify

Types: Symbol | Footprint | Spice Symbol | SCH Module | PCB Module | 3D Model

Classes: Work Space(0) | LCSC(26) | JLCPCB Assembled(18) | System(71) | Follow(0) | User Contributed(244)

amp

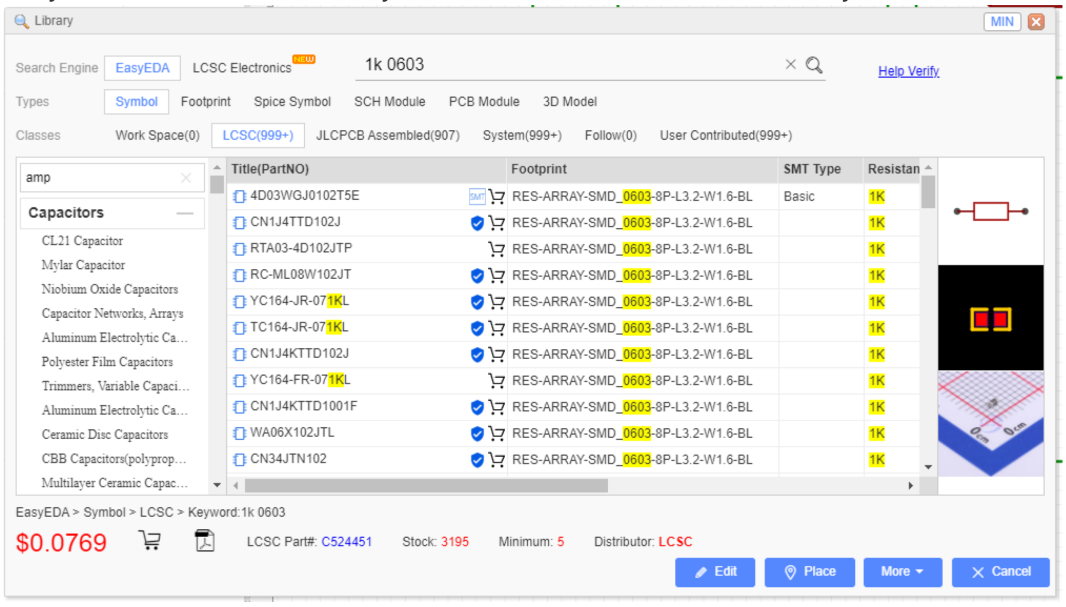
Capacitors: CL21 Capacitor, Mylar Capacitor, Niobium Oxide Capacitors, Capacitor Networks, Arrays, Aluminum Electrolytic Ca..., Polyester Film Capacitors, Trimmers, Variable Capaci..., Aluminum Electrolytic Ca..., Ceramic Disc Capacitors, CBB Capacitors(polyprop..., Multilayer Ceramic Capac...

Title(PartNO)	Footprint	SMT Type	Manufacturer
MAX232AEPE	DIP-16_L20.0-W6.4-P2.54-LS7.6-BL		MAXIM
MAX232N	PDIP-16_L19.7-W6.6-P2.54-LS10.9-BL		Texas Instruments
MAX232IDR	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	TI
MAX232ACSE+	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	Maxim Integrated
MAX232ESE+T	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL		MAXIM
MAX232DWR	SOIC-16_L10.3-W7.5-P1.27-LS10.3-BL	Extend	TI(Tex as Instruments)
MAX232ESE+	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	Maxim Integrated
MAX232DR	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	Texas Instruments
MAX232ECSE+T	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	Maxim Integrated
MAX232ID	SOIC-16_L9.9-W3.9-P1.27-LS6.0-BL	Extend	Texas Instruments
MAX232DWRG4	SOIC-16_L10.3-W7.5-P1.27-LS10.3-BL	Extend	TI

EasyEDA > Symbol > LCSC > Keyword:max232

$0.0769 — LCSC Part#: C524451 — Stock: 3195 — Minimum: 5 — Distributor: LCSC

- If you don't know the component's name
  For example, you want to find a resistor which value is 1kohm, footprint is 0603, at Libraries you can follow below steps:
  - 1.Choose the library type
  - 2.Typing the keyword such as `1k 0603`
  - 3.Click the search button
  - 4.Select the class you which is wanted of the result
  - 5.If you don't need the search you need to remove all the search keywords



Library

Search Engine: EasyEDA | LCSC Electronics — 1k 0603 — Help Verify

Types: Symbol | Footprint | Spice Symbol | SCH Module | PCB Module | 3D Model

Classes: Work Space(0) | LCSC(999+) | JLCPCB Assembled(907) | System(999+) | Follow(0) | User Contributed(999+)

amp

Capacitors: CL21 Capacitor, Mylar Capacitor, Niobium Oxide Capacitors, Capacitor Networks, Arrays, Aluminum Electrolytic Ca..., Polyester Film Capacitors, Trimmers, Variable Capaci..., Aluminum Electrolytic Ca..., Ceramic Disc Capacitors, CBB Capacitors(polyprop..., Multilayer Ceramic Capac...

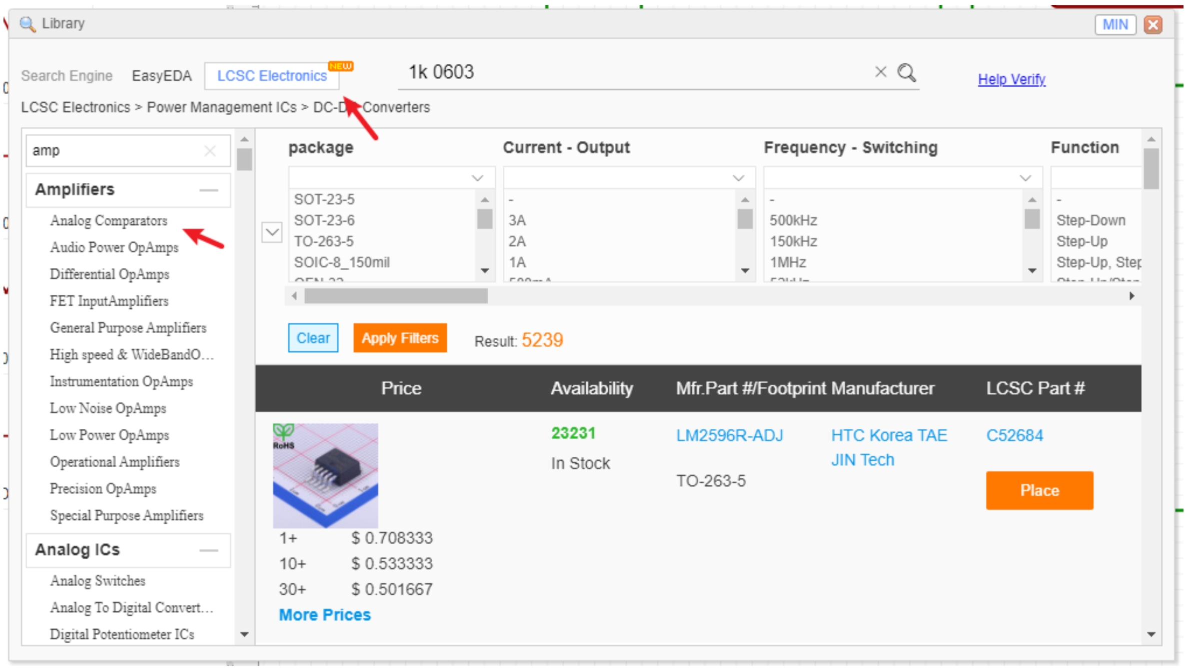
Title(PartNO)	Footprint	SMT Type	Resistan
4D03WGJ0102T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Basic	1K
CN1J4TTD102J	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
RTA03-4D102JTP	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
RC-ML08W102JT	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
YC164-JR-071KL	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
TC164-JR-071KL	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
CN1J4KTTD102J	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
YC164-FR-071KL	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
CN1J4KTTD1001F	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
WA06X102JTL	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K
CN34JTN102	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL		1K

EasyEDA > Symbol > LCSC > Keyword:1k 0603

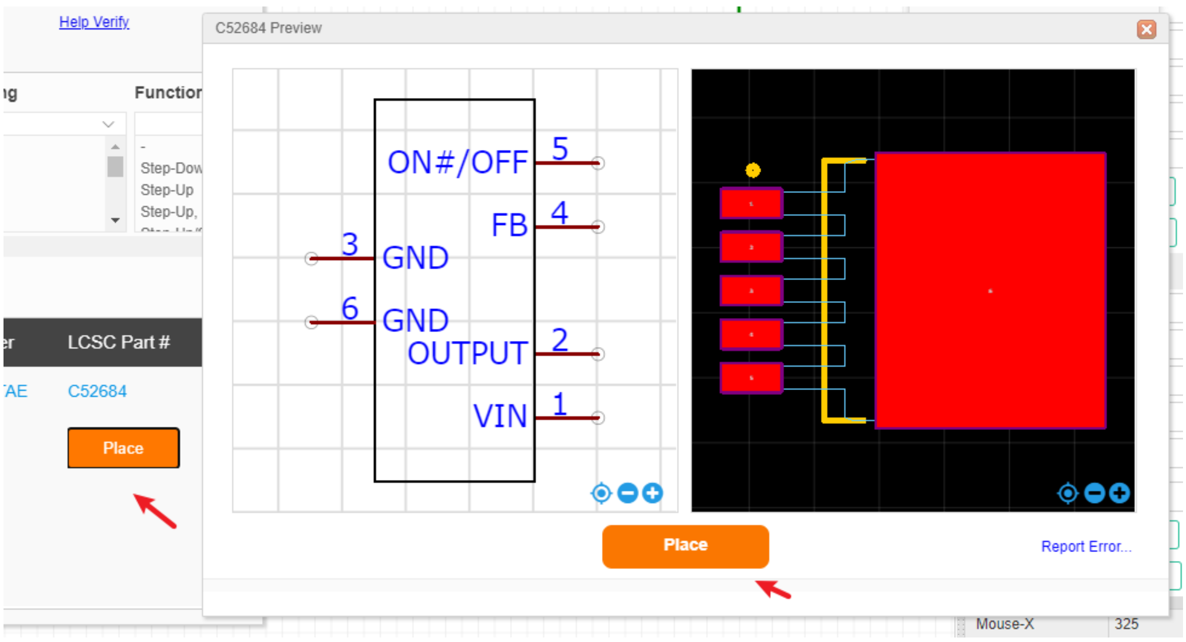
$0.0769 — LCSC Part#: C524451 — Stock: 3195 — Minimum: 5 — Distributor: LCSC

Search Engine - LCSC Electronics

When you want to find some parts by clearly parameter, you should try "Search Engine - LCSC Electronics", it all most same as LCSC.com.

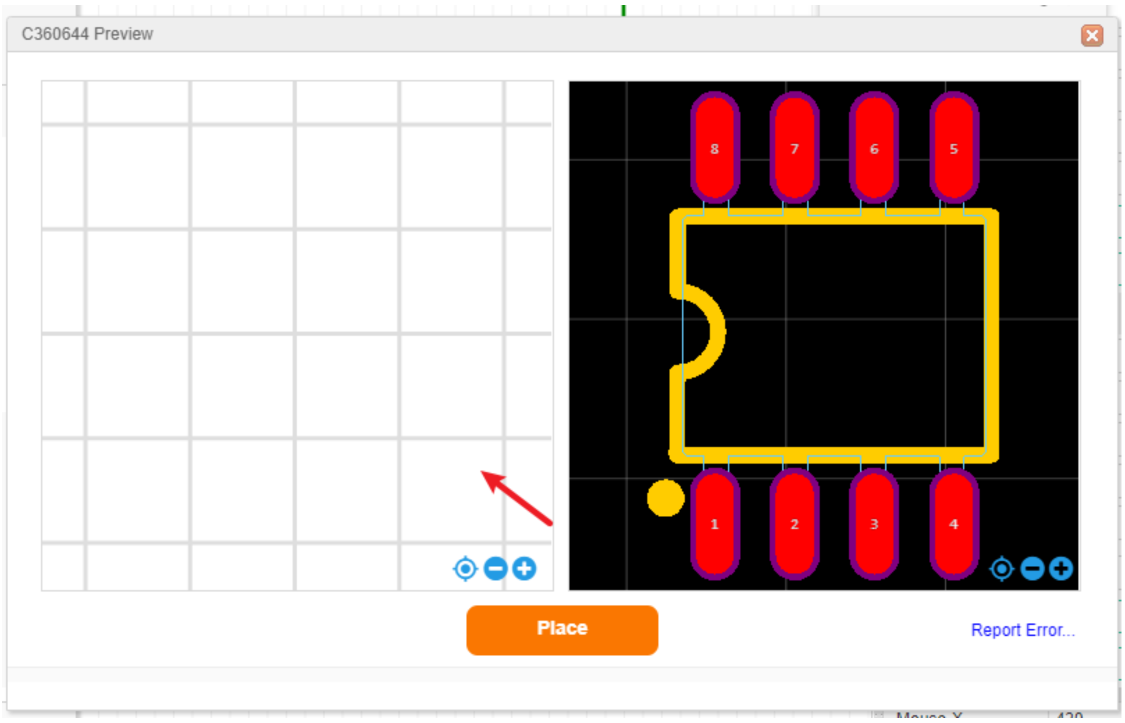


When you find out part, and you can place into the schematic:



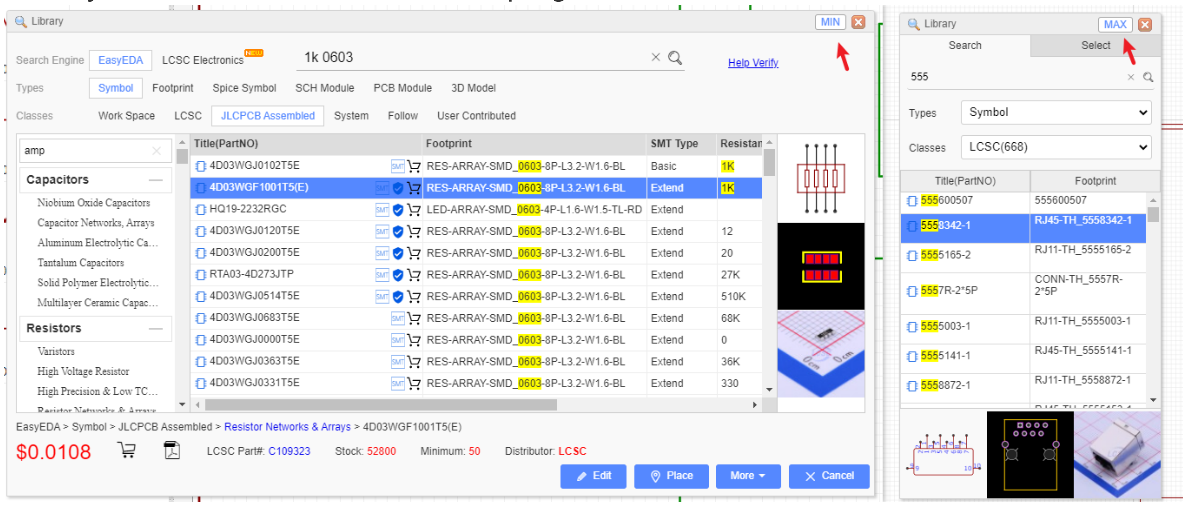
Notice:

- *The subpart can not be preview at Preview dialog window, if you find out this, you need to change to "Search Engine - EasyEDA" to place this part.*



Max and Min mode

If you want to place without close the "Library" dialog, you can change dialog mode to Min mode, just click the Min button at the top-right corner.



1k 0603

Title(PartNO)	Footprint	SMT Type	Resistan
4D03WGJ0102T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Basic	1K
4D03WGF1001T5(E)	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	1K
HQ19-2232RGC	LED-ARRAY-SMD_0603-4P-L1.6-W1.5-TL-RD	Extend	
4D03WGJ0120T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	12
4D03WGJ0200T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	20
RTA03-4D273JTP	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	27K
4D03WGJ0514T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	510K
4D03WGJ0683T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	68K
4D03WGJ0000T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	0
4D03WGJ0363T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	36K
4D03WGJ0331T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	330

EasyEDA > Symbol > JLCPCB Assembled > Resistor Networks & Arrays > 4D03WGF1001T5(E)

$0.0108 LCSC Part#: C109323 Stock: 52800 Minimum: 50 Distributor: LCSC

555

Title(PartNO)	Footprint
555600507	555600507
5558342-1	RJ45-TH_5558342-1
5555165-2	RJ11-TH_5555165-2
5557R-2*5P	CONN-TH_5557R-2*5P
5555003-1	RJ11-TH_5555003-1
5555141-1	RJ45-TH_5555141-1
5558872-1	RJ11-TH_5558872-1

Operations

When you hover the mouse over the picture of the Schematic symbol or PCB footprint, you will find a toolbar with "Edit", "Place", "More" buttons.

Place:

For parts you use infrequently, you don't need to Favorite them; just Place it into your canvas directly. Or you can double click the library to place.

Note:

- *EasyEDA supports multi-documents so please make sure that you are placing the part into the right (active) document. The active document is the one with the highlighted tab.*
- *You can't place a Schematic symbol into a PCB file, or a PCB Footprint into a schematic.*
- *EasyEDA will try the best to make sure the library is correct, but it still has incorrect parts, if you find any incorrect parts please let us known. suggested order a sample first before*

ordering a big order.

Edit:

If you want to create your own version of a symbol or footprint then you can open an existing part from the library to use as a template, edit it and then save it to your local **Work Space** library in **Library** of the Navigation Panel.

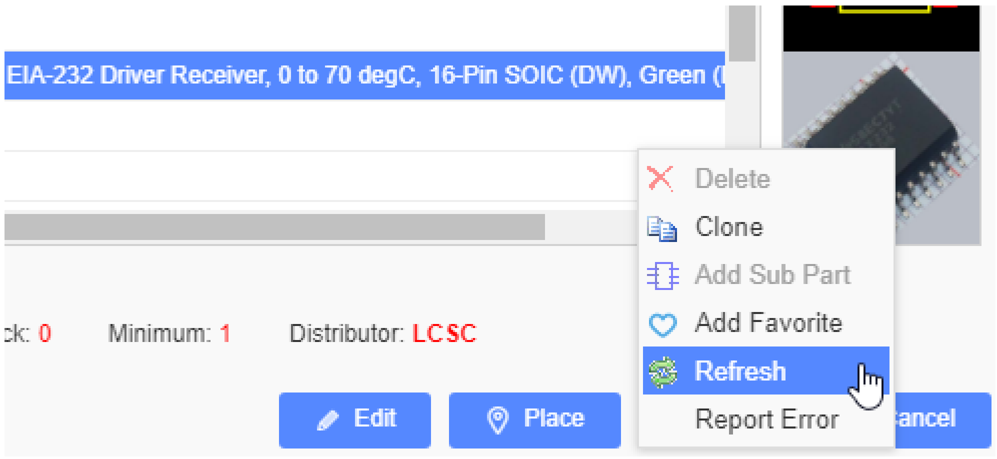
More:

We can't promise that every component in the library is free of errors so please check all symbols and footprints carefully before you commit to a PCB order.

If you do find a mistake in a component, please use the `Report Error`, so that we can fix it.

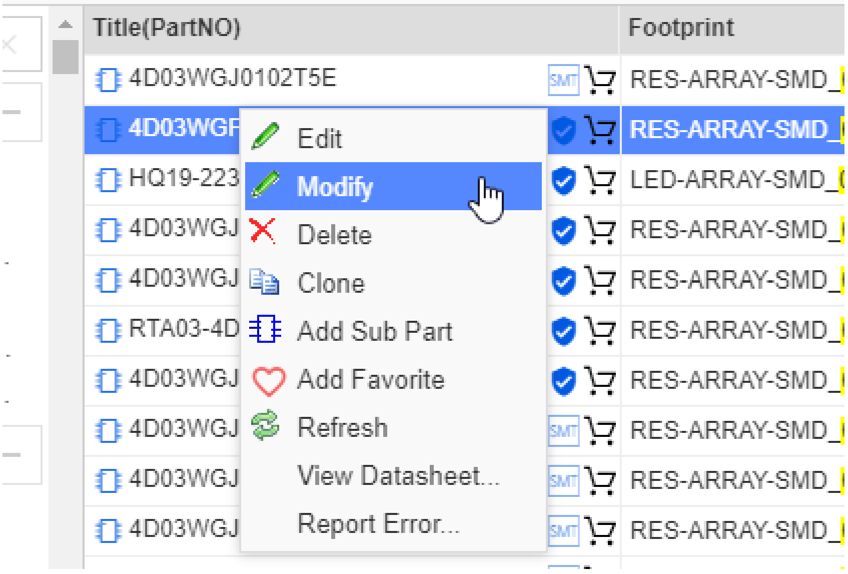
Components with sub parts (multi-device footprints).

When you find a component with sub-parts, you can't Place or Edit it, but you can Favorite and Clone it as your own part, which you can then edit.



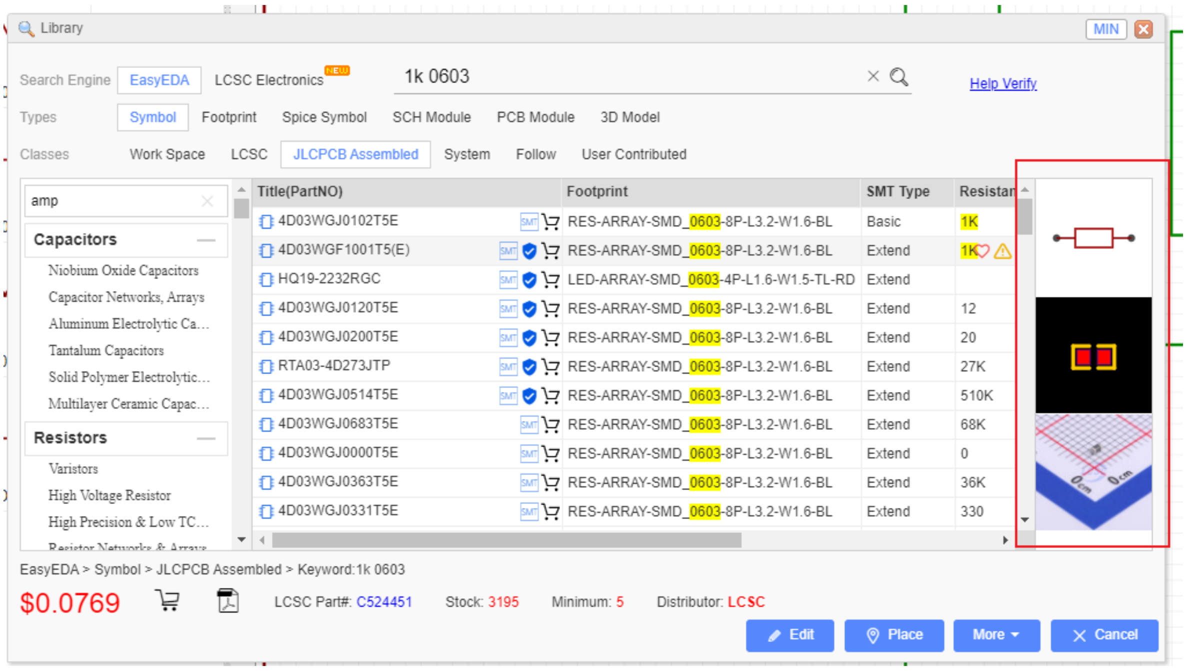
Right-Click

When you right-click the part list, you can edit its tags, add favorite etc.



Preview Image

Every library when you click, you can check its preview image, such as symbol, footprint, production picture. Click the the image you can open it quickly.


Library

Search Engine: EasyEDA | LCSC Electronics NEW — 1k 0603 — Help Verify

Types: Symbol | Footprint | Spice Symbol | SCH Module | PCB Module | 3D Model

Classes: Work Space | LCSC | JLCPCB Assembled | System | Follow | User Contributed

amp

Capacitors: Niobium Oxide Capacitors, Capacitor Networks, Arrays, Aluminum Electrolytic Ca…, Tantalum Capacitors, Solid Polymer Electrolytic…, Multilayer Ceramic Capac…

Resistors: Varistors, High Voltage Resistor, High Precision & Low TC…, Resistor Networks & Arrays

Title(PartNO)	Footprint	SMT Type	Resistan
4D03WGJ0102T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Basic	1K
4D03WGF1001T5(E)	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	1K
HQ19-2232RGC	LED-ARRAY-SMD_0603-4P-L1.6-W1.5-TL-RD	Extend	
4D03WGJ0120T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	12
4D03WGJ0200T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	20
RTA03-4D273JTP	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	27K
4D03WGJ0514T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	510K
4D03WGJ0683T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	68K
4D03WGJ0000T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	0
4D03WGJ0363T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	36K
4D03WGJ0331T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	330

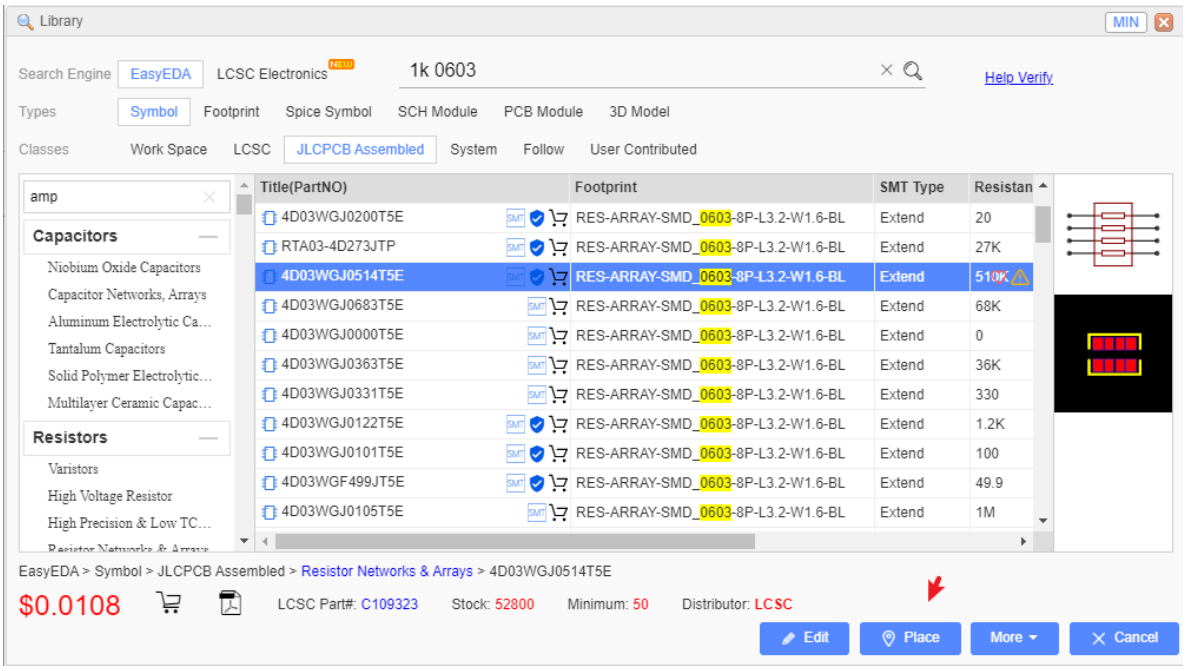
EasyEDA > Symbol > JLCPCB Assembled > Keyword:1k 0603

$0.0769 LCSC Part#: C524451 Stock: 3195 Minimum: 5 Distributor: LCSC

Edit | Place | More | Cancel

Placing Components

Find the component which you plan to place to your schematic at "Libraries", then move your mouse to the canvas and left click. If you want to add more, just left click again. To end the current sequence of placements, right click once or press `ESC`.


Library

Search Engine: EasyEDA | LCSC Electronics NEW — 1k 0603 — Help Verify

Types: Symbol | Footprint | Spice Symbol | SCH Module | PCB Module | 3D Model

Classes: Work Space | LCSC | JLCPCB Assembled | System | Follow | User Contributed

amp

Capacitors: Niobium Oxide Capacitors, Capacitor Networks, Arrays, Aluminum Electrolytic Ca…, Tantalum Capacitors, Solid Polymer Electrolytic…, Multilayer Ceramic Capac…

Resistors: Varistors, High Voltage Resistor, High Precision & Low TC…, Resistor Networks & Arrays

Title(PartNO)	Footprint	SMT Type	Resistan
4D03WGJ0200T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	20
RTA03-4D273JTP	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	27K
4D03WGJ0514T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	510K
4D03WGJ0683T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	68K
4D03WGJ0000T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	0
4D03WGJ0363T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	36K
4D03WGJ0331T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	330
4D03WGJ0122T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	1.2K
4D03WGJ0101T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	100
4D03WGF499JT5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	49.9
4D03WGJ0105T5E	RES-ARRAY-SMD_0603-8P-L3.2-W1.6-BL	Extend	1M

EasyEDA > Symbol > JLCPCB Assembled > Resistor Networks & Arrays > 4D03WGJ0514T5E

$0.0108 LCSC Part#: C109323 Stock: 52800 Minimum: 50 Distributor: LCSC

Edit | Place | More | Cancel

Don't try to Drag and Drop a component to the canvas EasyEDA team thinks that Click-Click to place components will be easier to use than a Click-Drag mode.

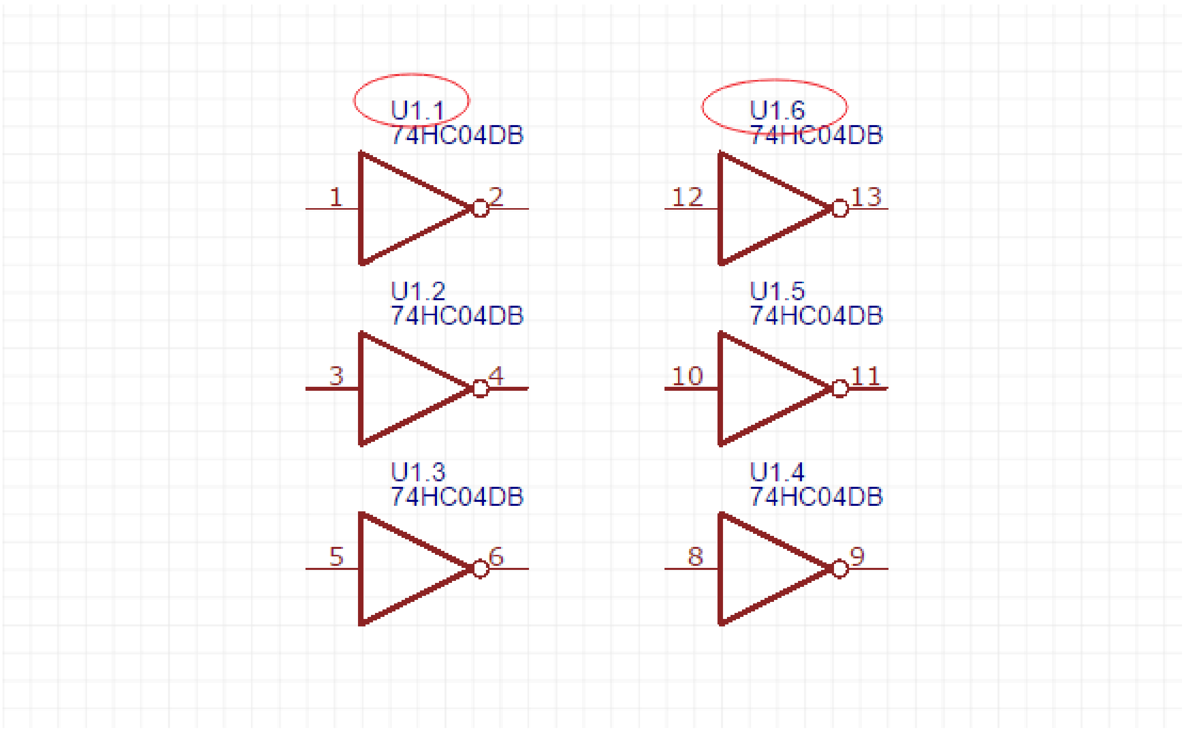
Multi-part Components

The number of pins on some components can be quite large. That's why it's easier to divide such a component into several parts or functional blocks, it calls multi-parts or subparts.

As a simple example, there are six gates in the 74HC04 Hex Inverter component. To avoid clutter in the schematic, GND and VCC pins of such components are usually served by a separate part of the component. This is really convenient as it doesn't interfere the working process with logical parts. The NetLabel names of VCC and GND Pin are usually hidden.

When placing the 74HC04 on a schematic, it will look like the screenshot below.

Note: *The component Prefix will be in form of: U?.1, U?.2 etc.*



If you click the father-part and place on the schematic, the remaining subparts will be placing one by one, if you click the one of the subpart, you will placing U1.1, U2.1, U3.1 etc.

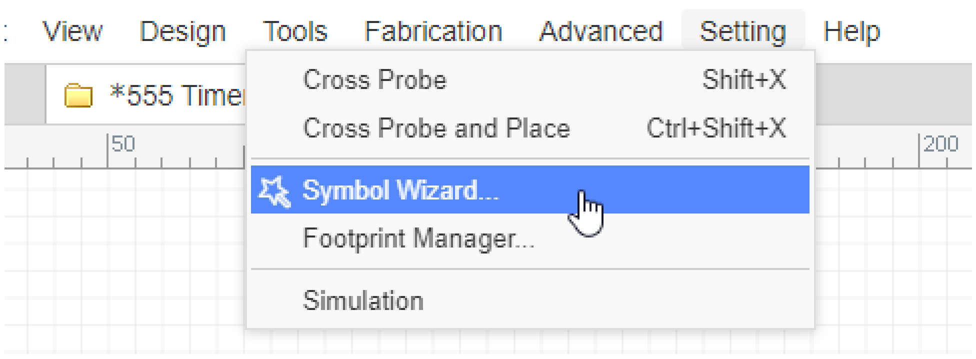
How to create multi-part(subpart) please refer Create Symbol

Schematic Symbol Wizard

How many times have you hit a schematic capture roadblock because you couldn't find a component symbol?

Well, in EasyEDA that would be never because the Schematic **Symbol Wizard** provides a quick and easy way to create a general schematic library symbol.

Via: **Top Menu > Tools > Symbol Wizard** in a new schematic symbol or sheet document.

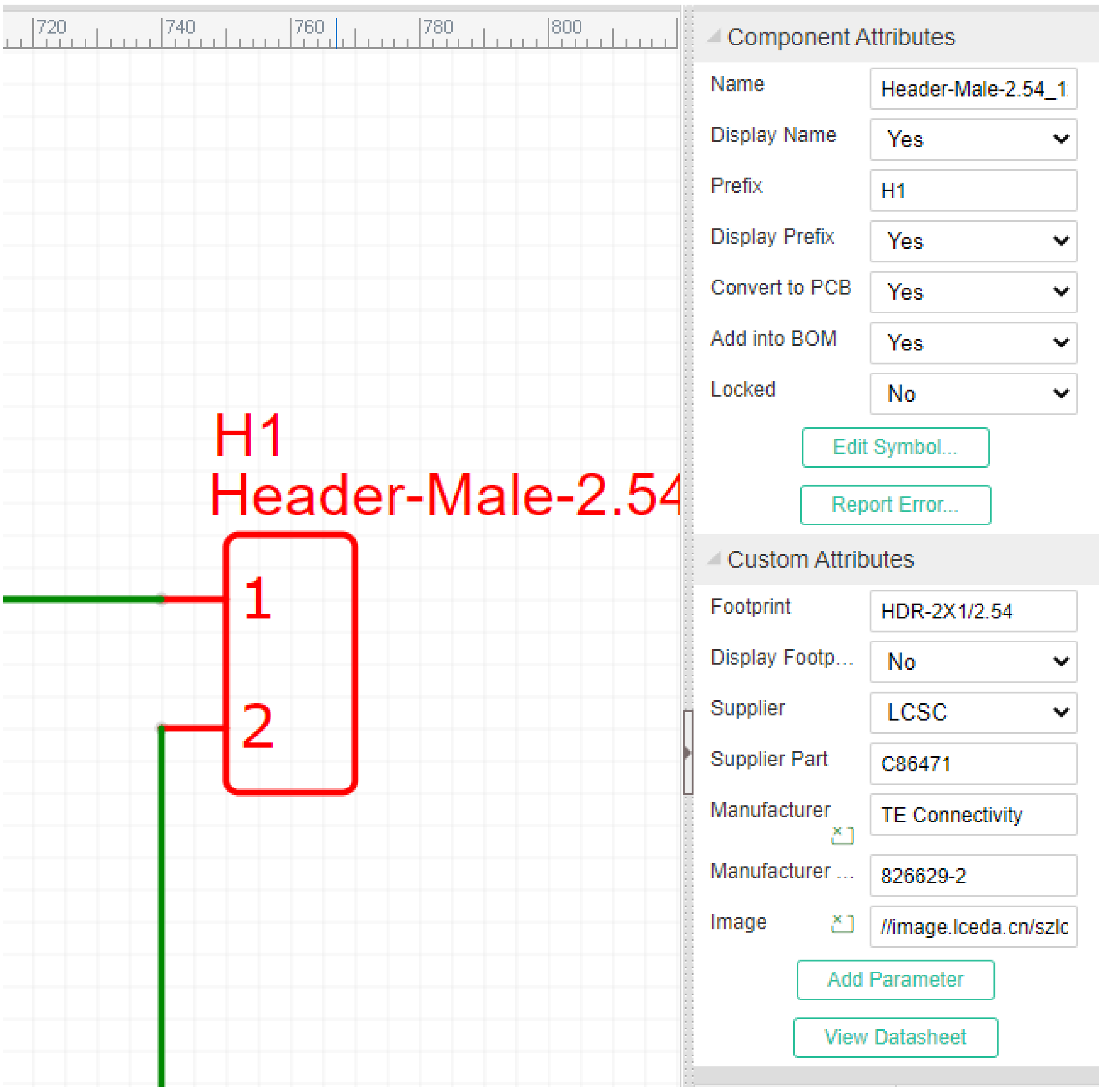


The professional function please refer at Schematic Symbol Wizard

Component Attributes

Component Attributes

After selecting a component, you can find the component's attributes in the right hand Properties panel.



Component Attributes

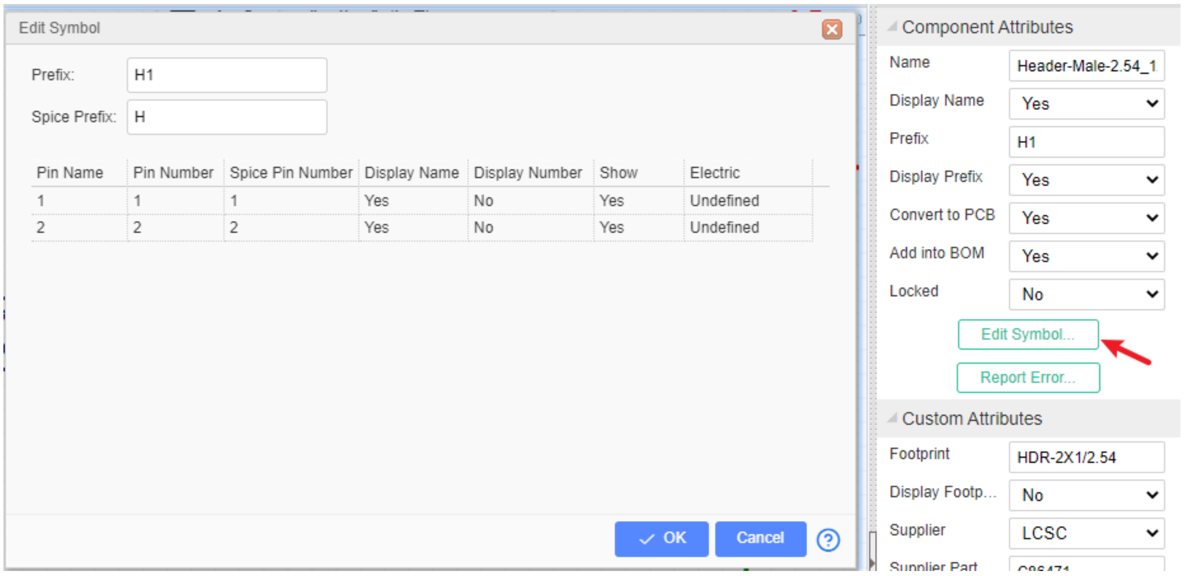
Name	Header-Male-2.54_1
Display Name	Yes
Prefix	H1
Display Prefix	Yes
Convert to PCB	Yes
Add into BOM	Yes
Locked	No

Custom Attributes

Footprint	HDR-2X1/2.54
Display Footp...	No
Supplier	LCSC
Supplier Part	C86471
Manufacturer	TE Connectivity
Manufacturer ...	826629-2
Image	//image.lceda.cn/szlc

1.Component Attributes:

You can change the **Prefix** and **Name** here， And make them **visible** or **invisible**.
If you want edit this component, you can click **Edit Symbol**.


Edit Symbol

Prefix: H1

Spice Prefix: H

Pin Name	Pin Number	Spice Pin Number	Display Name	Display Number	Show	Electric
1	1	1	Yes	No	Yes	Undefined
2	2	2	Yes	No	Yes	Undefined

Component Attributes

Name	Header-Male-2.54_1
Display Name	Yes
Prefix	H1
Display Prefix	Yes
Convert to PCB	Yes
Add into BOM	Yes
Locked	No

Custom Attributes

Footprint	HDR-2X1/2.54
Display Footp...	No
Supplier	LCSC

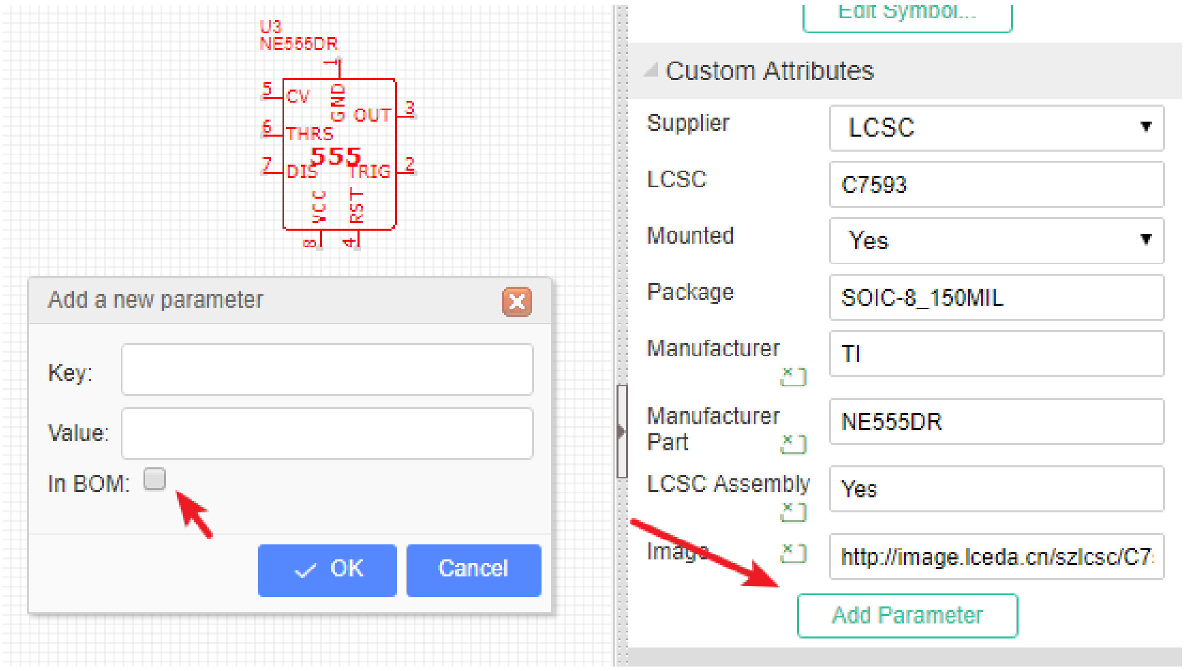
If the component's property "Convert to PCB" is set as "No", it will not appear at footprint manager.

2.Custom Attributes:

You can change *component's supplier*, *change footprint*, and *add new parameter*.

Define BOM Parameters

After selected a schematic symbol, you can add a parameter, and you can mark it as `In BOM`, when you export a BOM file, you can find this parameter in CSV file.



Custom Attributes

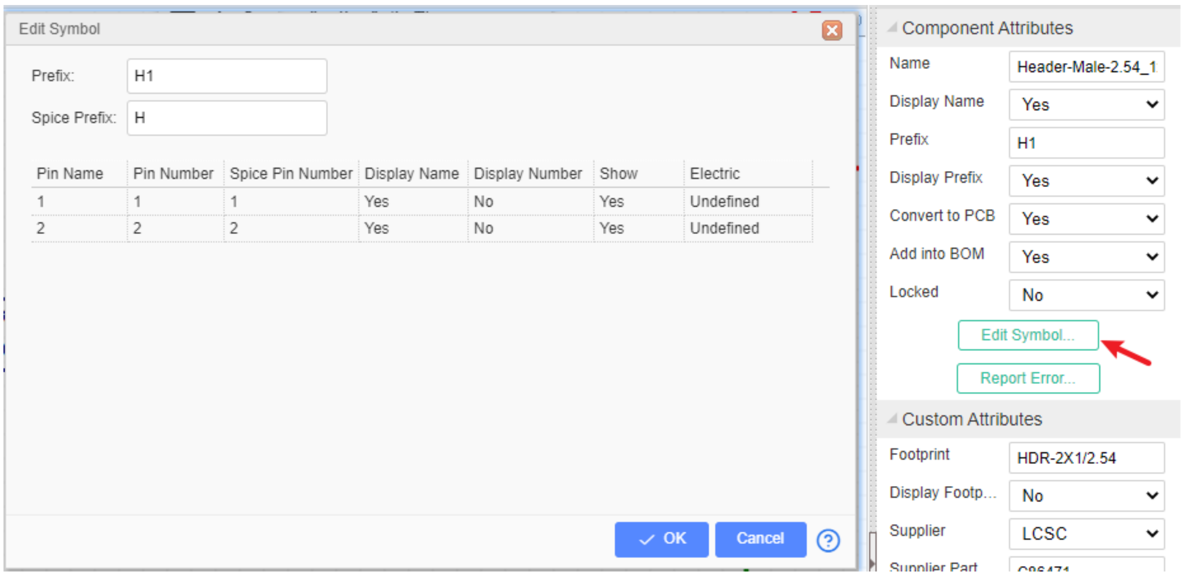
Supplier	LCSC
LCSC	C7593
Mounted	Yes
Package	SOIC-8_150MIL
Manufacturer	TI
Manufacturer Part	NE555DR
LCSC Assembly	Yes
Image	http://image.lceda.cn/szlcsc/C7

Modify Symbol Pinmap Information

When you select a component, for opening the Modify symbol information dialog, you can do:

- Or press the `I` hotkey;
- Or click the Edit Symbol on the Parts Attributes on the left panel.
- Or click the Symbol and right-click, choose the "Edit Symbol" menu.

Using this dialog you can edit the pin names and numbers, for example, to suit a different footprint or device variant. You can also enter a Spice Prefix and swap the spice Pin order to make your symbol usable in simulation.


Edit Symbol

Prefix: H1

Spice Prefix: H

Pin Name	Pin Number	Spice Pin Number	Display Name	Display Number	Show	Electric
1	1	1	Yes	No	Yes	Undefined
2	2	2	Yes	No	Yes	Undefined

Component Attributes

Name	Header-Male-2.54_1
Display Name	Yes
Prefix	H1
Display Prefix	Yes
Convert to PCB	Yes
Add into BOM	Yes
Locked	No

Custom Attributes

Footprint	HDR-2X1/2.54
Display Footp...	No
Supplier	LCSC

More detailed description of PCB and Spice Prefixes and pin numbers at next section.

Prefixes and Pin Numbers

Device and subcircuit (or hierarchical block) symbols created for use in schematics that are intended to be run as spice simulations, in addition to having a PCB Prefix that is used for the reference designator in the schematic, also have a **Spice Prefix**. They also have two sets of pin numbers: PCB pins and Spice pins.

PCB Prefix and Spice Prefix

For more information please refer at Simulation: Schematic symbols: prefixes and pin numbers

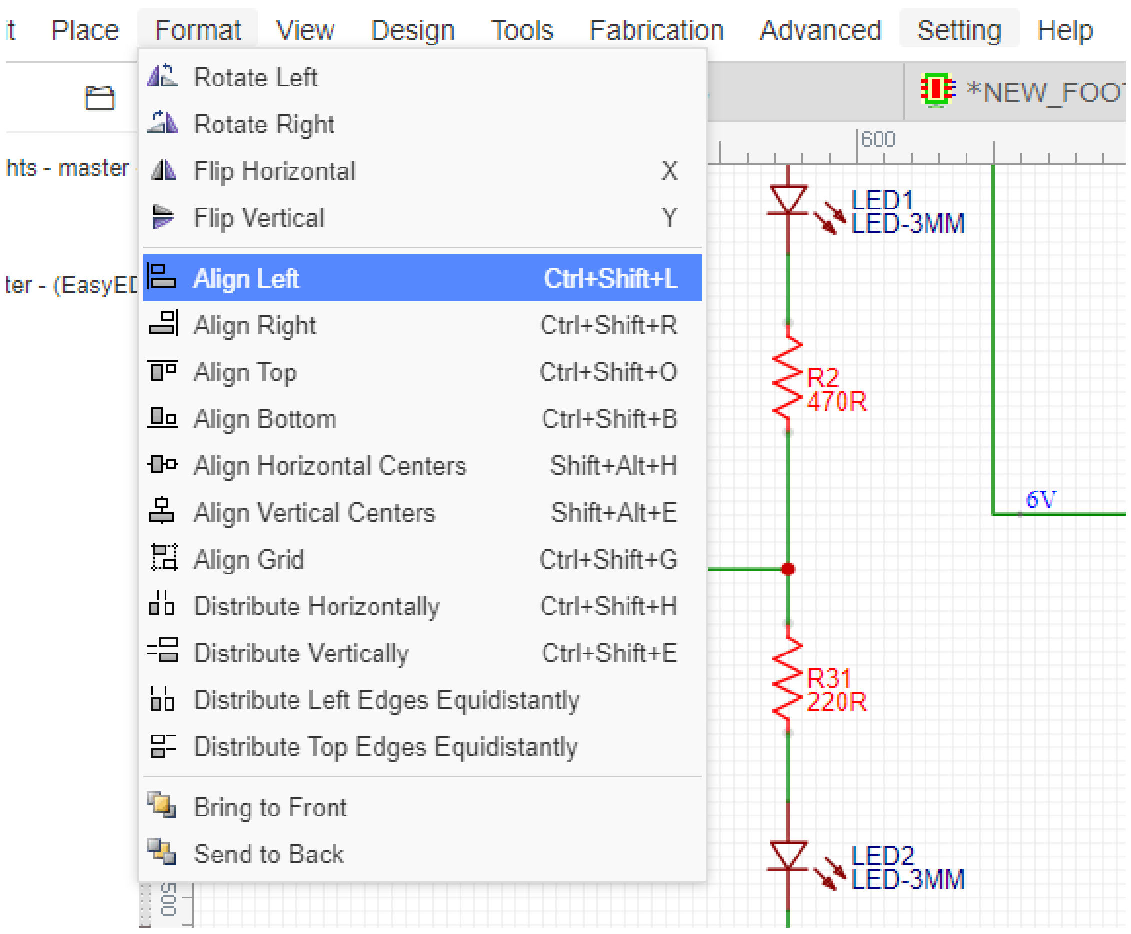
Component Adjust

Adjusting Components

About adjusting components you can:

1. Move components with your mouse
2. Move components with the arrow keys.
3. Find components with the Design Manager via the `CTRL+D` hotkey: select the component in the Design Manager to pan it to the centre of the canvas and then move it with your mouse.

4. Align the components:



Rotating the Prefix and Value (Name) of components

The default Prefix and Value (or name) of EasyEDA components are horizontal. To change them to vertical, Left click the prefix or value and when it is highlighted in **red** color, then press the **rotation** hotkey **Space** and you're done.

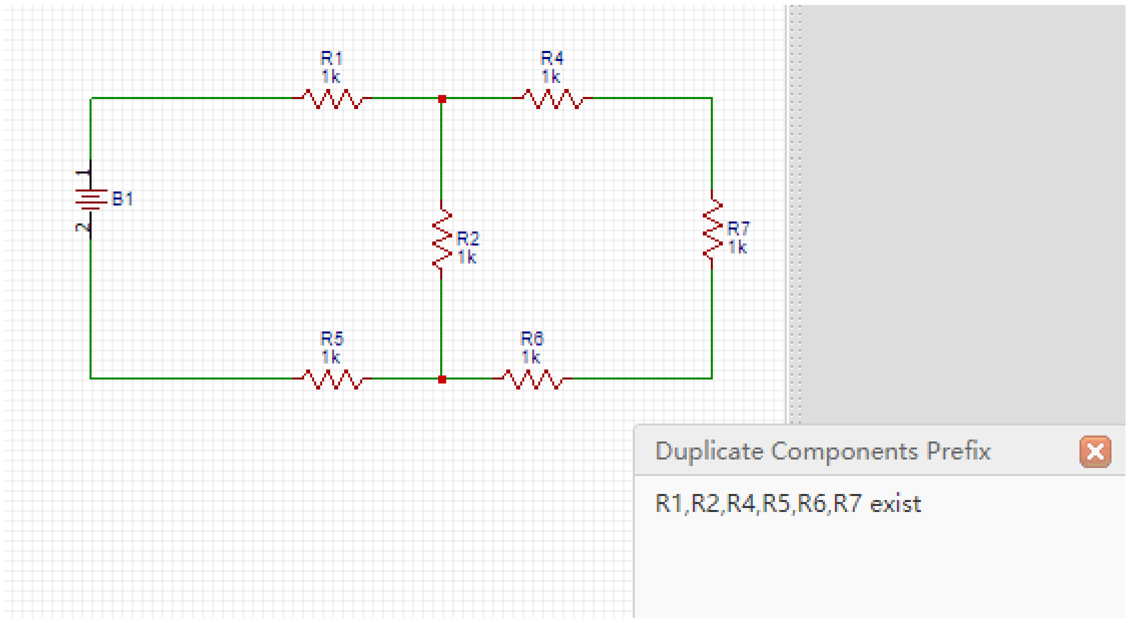
Components Prefixes

Prefix Start

In EasyEDA, at the first new schematic the prefix will start as U1/R1..etc, and EasyEDA support global unique prefix at multi-sheet now.

Prefix Conflict Error

Sometimes, if you save a sheet to another project, when you convert a project to PCB, open the Design manager or run a simulation, you will get a Prefix Conflict error message.

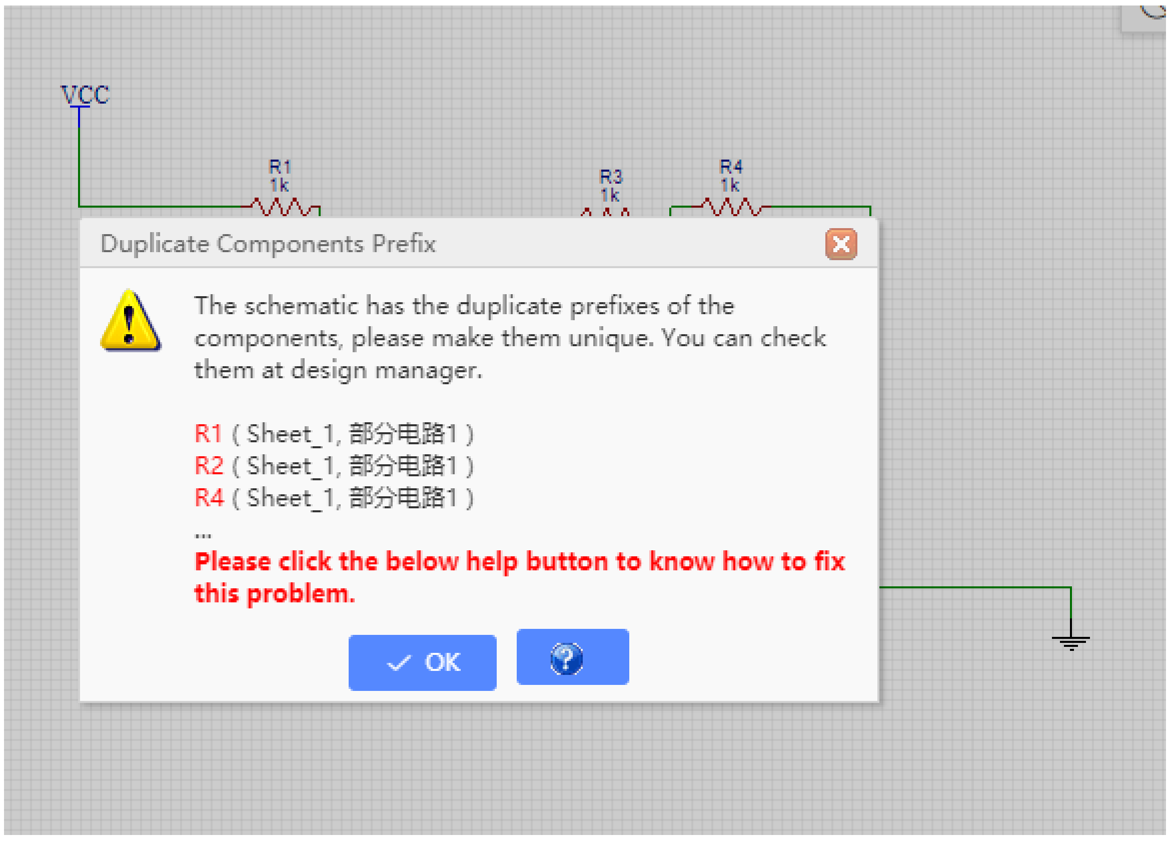


Duplicate Components Prefix

R1,R2,R4,R5,R6,R7 exist

In this schematic, you will find two components with the R4 reference designator, so you just need to change one to Rx where x is a unique number in that schematic.

It may be tempting to backup a schematic into the same project as the original, however, if an attempt is then made to do Convert Project to PCB, you will get the Prefix Conflict error for every component.



Duplicate Components Prefix

The schematic has the duplicate prefixes of the components, please make them unique. You can check them at design manager.

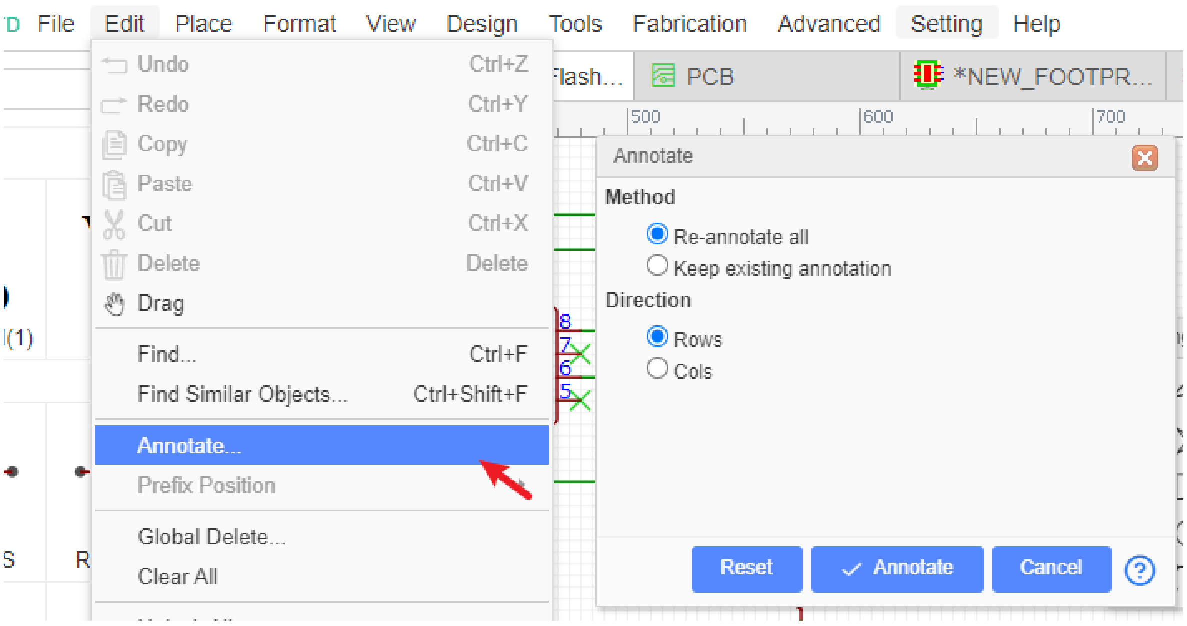
R1 (Sheet_1, 部分电路1)
R2 (Sheet_1, 部分电路1)
R4 (Sheet_1, 部分电路1)
...
Please click the below help button to know how to fix this problem.

In the above image, you can find the two identical copies of the same schematic, which when you Convert Project to PCB, EasyEDA will try to merge into a single schematic, so every item will have 2 copies.
To fix this, you just have to create a backup project and remove or better still save backup copies of your schematics to that project.

Annotate

After creating a schematic, it is quite likely that you have component Prefixes (reference designators) that are in no particular order on the canvas. You may also have duplicates. You can automatically renumber/reset all the components' prefix by using the **Annotate** function.

Via: **Top Menu > Edit > Annotate**



Various Annotate possibilities are available:

- **Re-annotate all**: resets all existing annotation and then annotates all components again from scratch;
- **Keep existing annotation**: annotates new components only (i.e. those whose reference designator finishes with ? like R? or U?).
- **Direction**: Rows annotates across the schematic in a raster pattern from top left to bottom right;
  Cols annotates down the schematic in a raster pattern from top left to bottom right.
- **Annotate**: applies the selected annotation actions.
- **Reset**: if you want to reset all the reference designators to end with '?', just click the Reset button. After that, R1 will be R?, U1 will be U? etc.

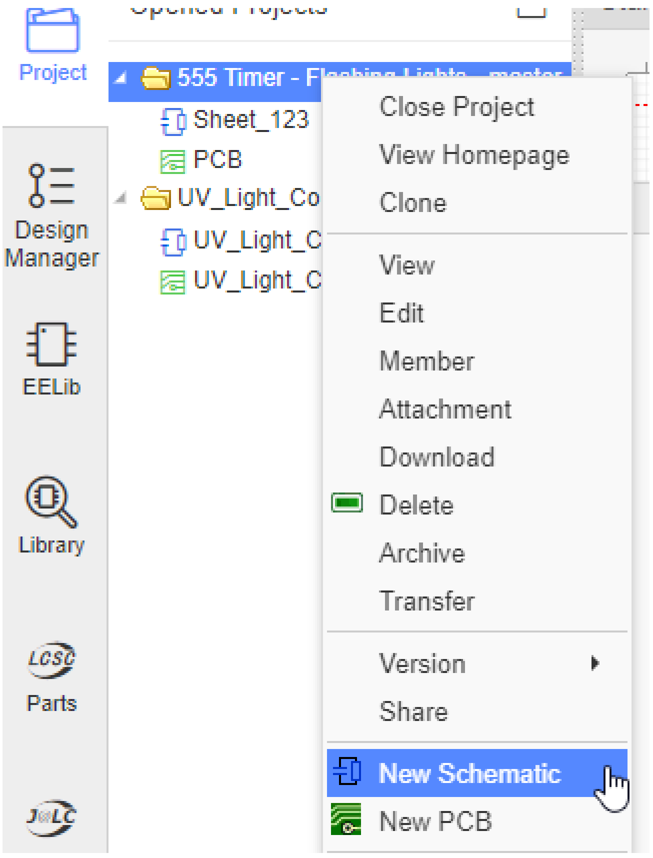
Note:

- *Reset does not reset annotation back to where it was before pressing the Annotate button.*
- *Annotation cannot be undone! if you do not accept the result: close all of the affected schematics without saving. If you do accept the result: make sure you save all of the affected schematics.*

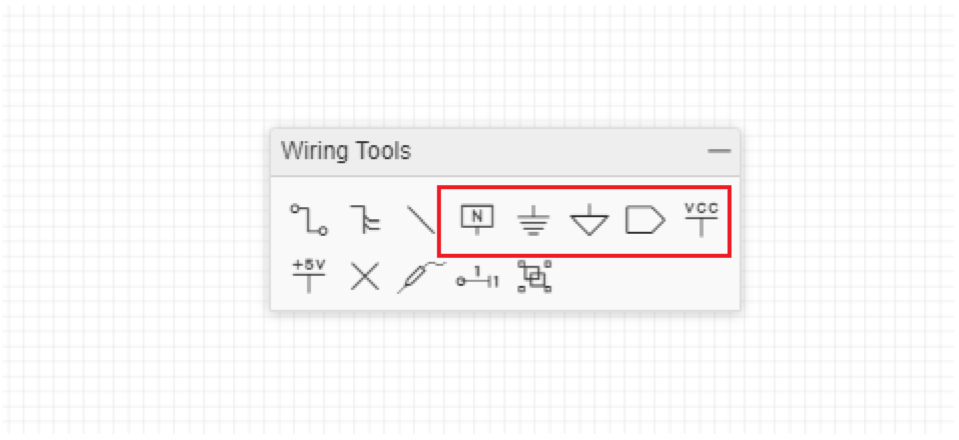
Multi-Sheet

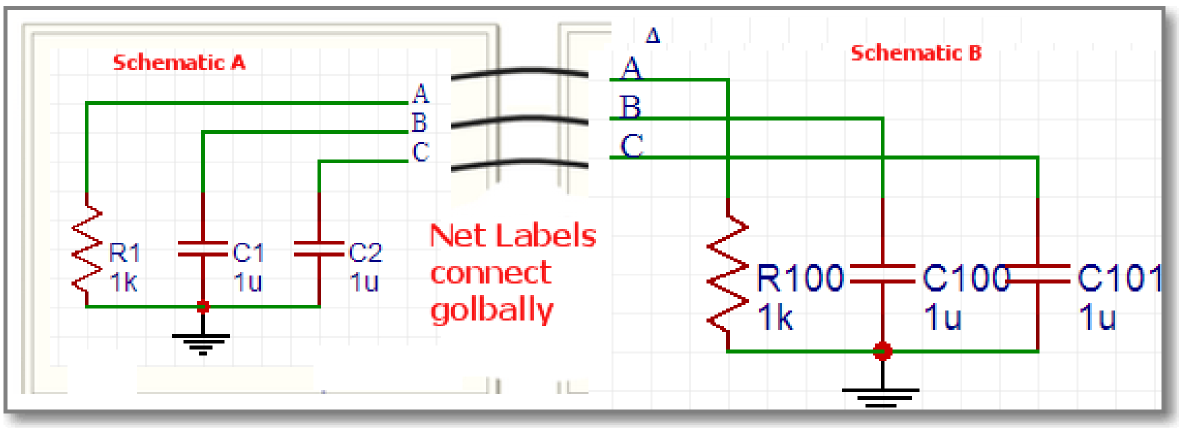
EasyEDA does not support true hierarchical designs but it does support **multi-sheet designs**.

You can put several schematics in one project with connections between made by NetLabels/netPorts. All nets in EasyEDA are global so if you create a netlabel `DATA0` in sheet A and then create a netlabel `DATA0` in sheet B, when sheet A and sheet B are in the same project, they will be connected.

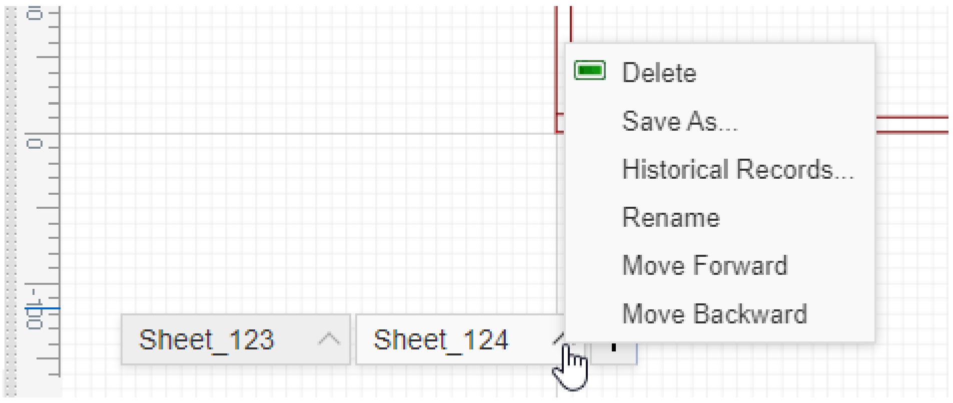


Multi-sheet designs(equivalent to a circuit spread over several pieces of paper), all schematics under the same project will be merged into one when be converted to PCB connecting in **Netlabel**, **Netflag**, **Netport**.

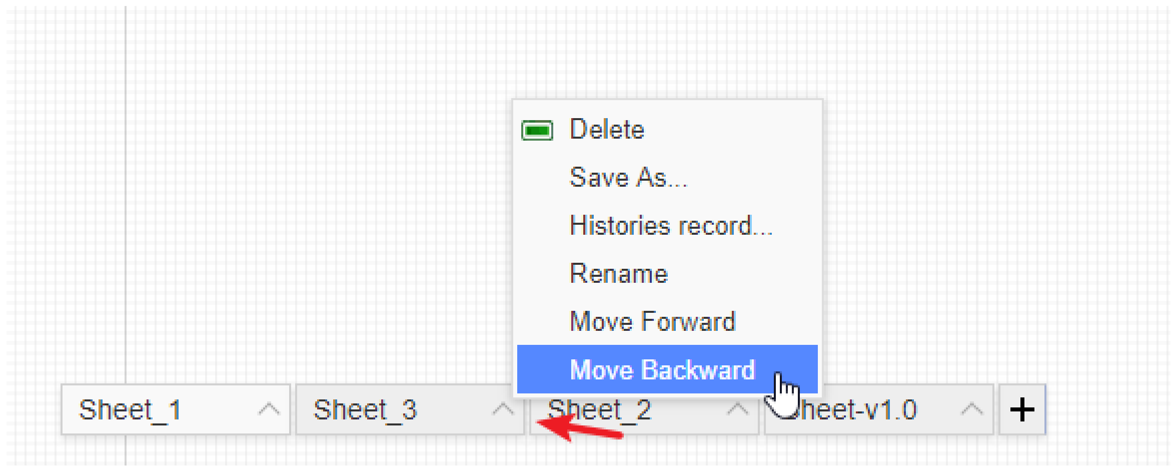




You can click the Sheet tabs on the left-down corner to switch the Sheets, and right-click the sheet tab you can "Save as", check "Histories record", "Move Forward/Backward","Rename" and "Delete" the sheet.



If you want to arrangement the sheets order, you click the menu of the sheet icon: Move Forward/Move Backward.

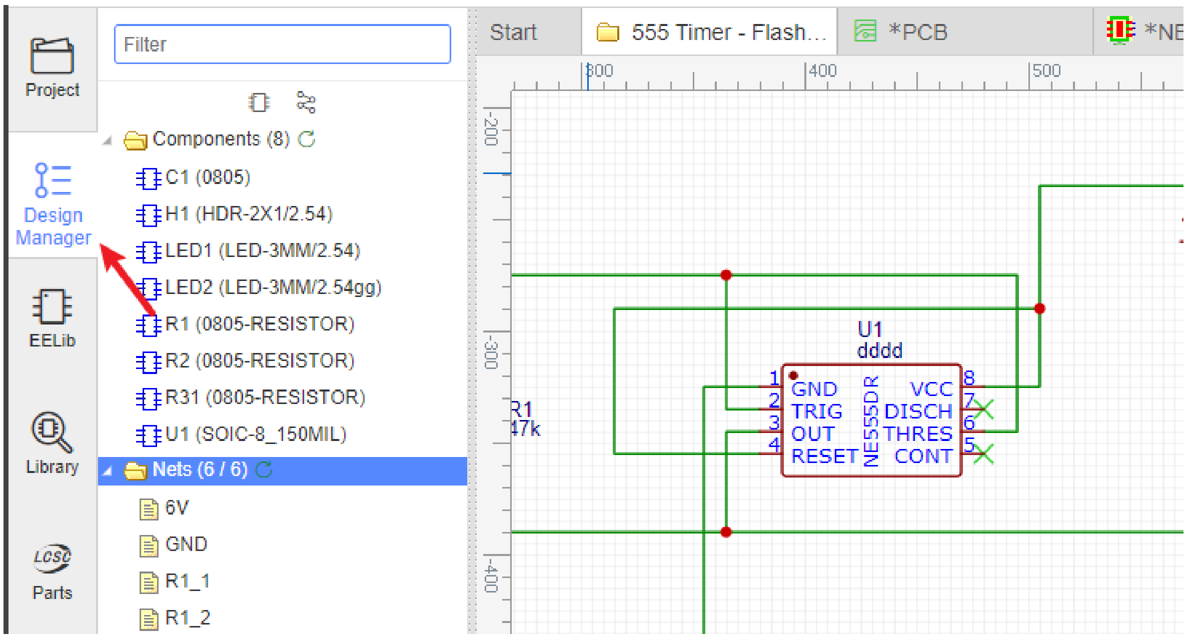


Note:
EasyEDA support global unique prefixes, when you place components in different sheet, the editor will auto annotate the prefix. If you save as a sheet to another project, please make all of the prefixes unique, if the Sheet A has a R1, and the Sheet B has a R1, then you will get a Prefix Conflict Error.

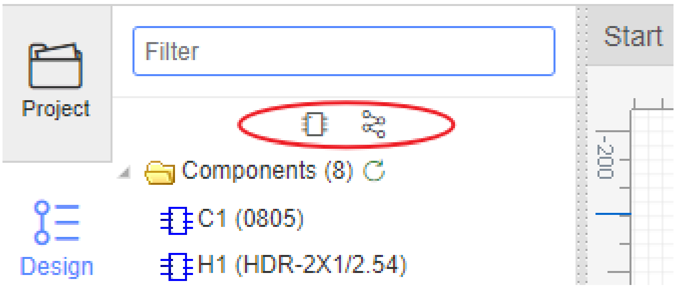
Design Manager

With large schematics it can be hard to find the components quickly. Sometimes, you may make a mistake such as wiring to a wrong component pin. So you need a tool to help you out. **Design Manager** is just the tool.

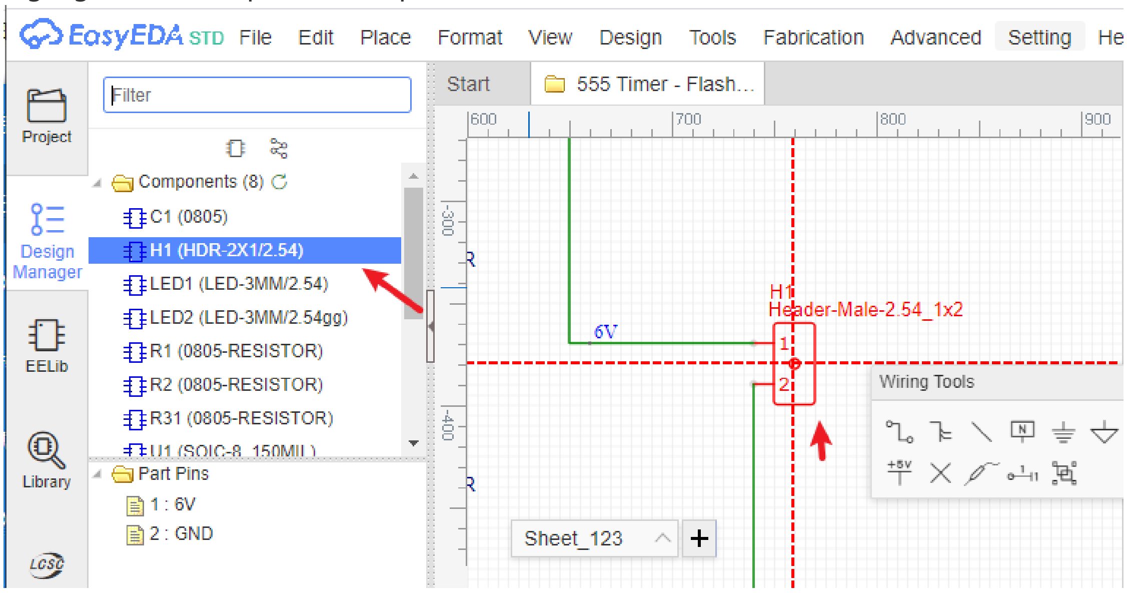
Just press the `CTRL+D` hotkey to open the Design Manager.
or click it via on the left navigation panel:



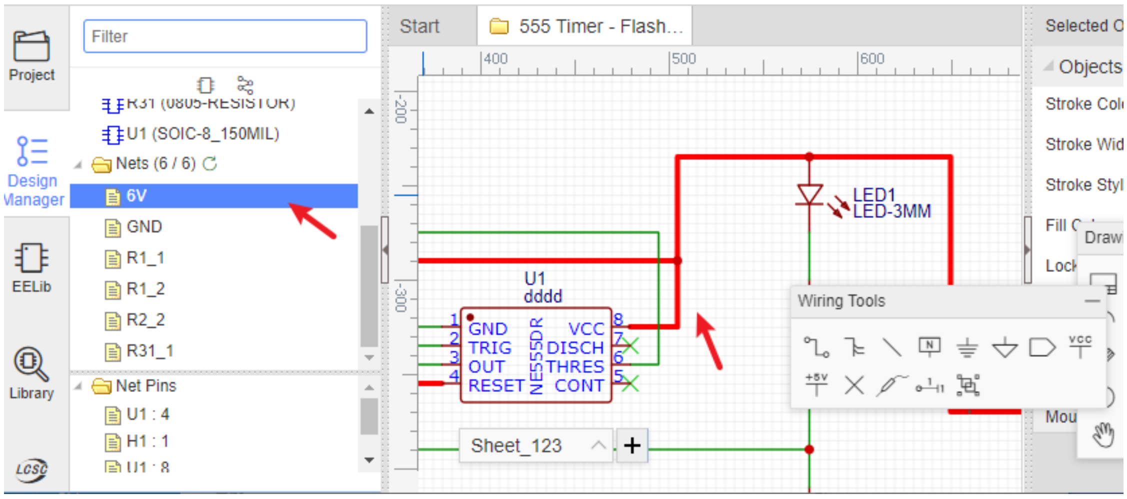
You can click the jump icon to goto the folder quickly.



- **Filter**: You can find your components or net name easily: for example, if you want to find all capacitances, you just need to type `C`;
- **Components**: Lists all the components in this schematic. Clicking on a Component item highlights that component and pans it to the center of the window.

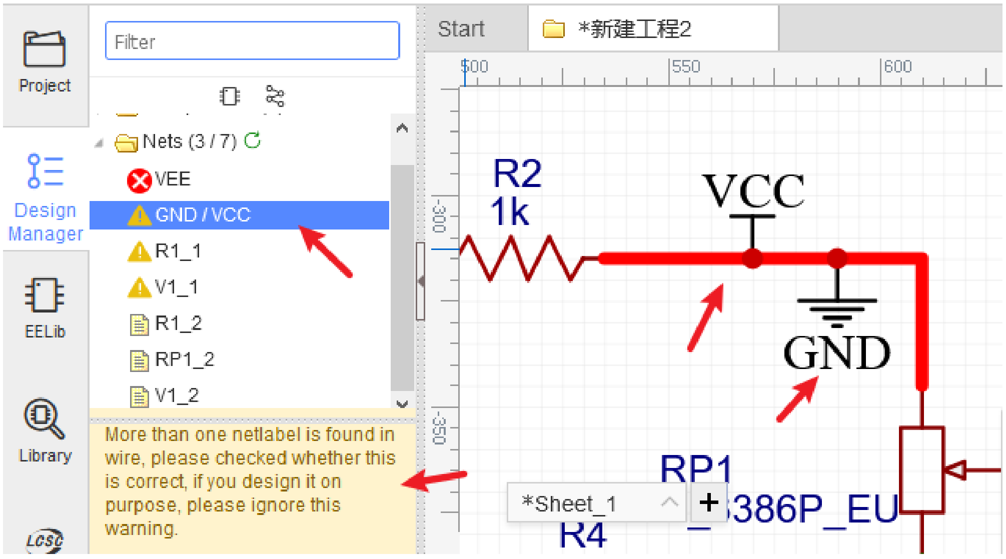


- **Nets**: Lists all the nets in this schematic. A net must connect at least two Pins, or the net name will be marked as a red error. When click the net name, the canvas wire will highlight and being large, when you click the empty space to unhighlight:



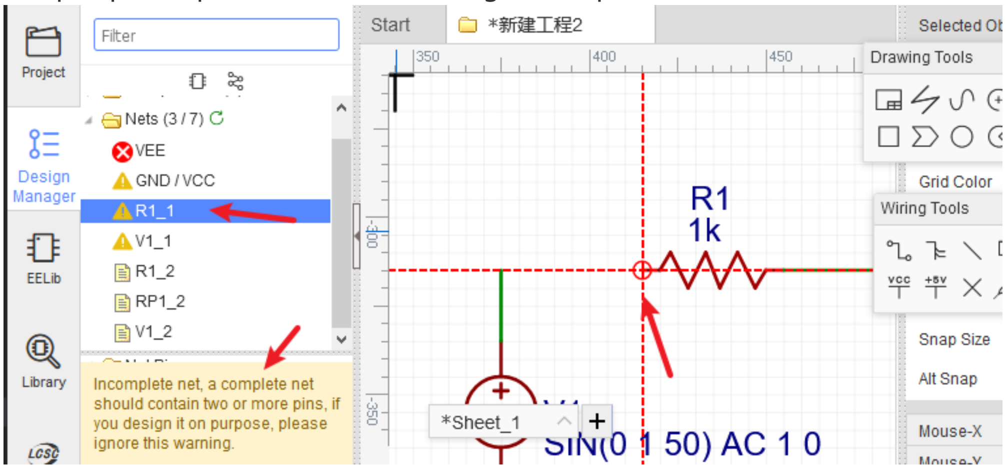
When you click the net name, you will see the tip at the bottom-left corner.

  - Net warning: It will show a prompt exclamation point icon.
    - When multiple netlabels on one wire, please check whether if it is correct or just connected by mistake. You need to click this net and find it out. If your netflag or netlabel only connect one pin, it will show warning.



More than one netlabel is found in wire, please checked whether this is correct, if you design it on purpose, please ignore this warning.

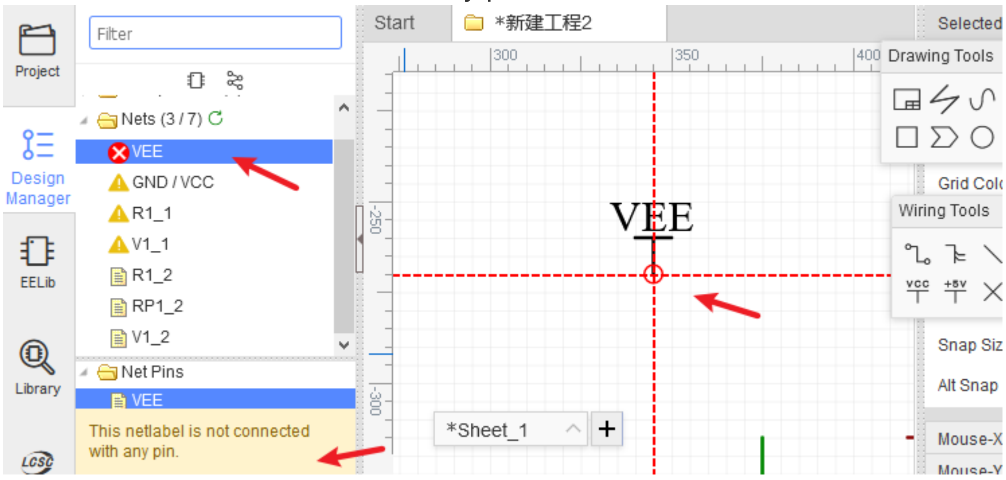
    - The part's pin doesn't place the netlabel, or doesn't connect other pins, or doesn't place No Connect Flag. A completed net must connects two and more pins, so that, you need to modify your net connection. If you don't need to use this pin, please place a No Connect Flag on the pin.



Incomplete net, a complete net should contain two or more pins, if you design it on purpose, please ignore this warning.

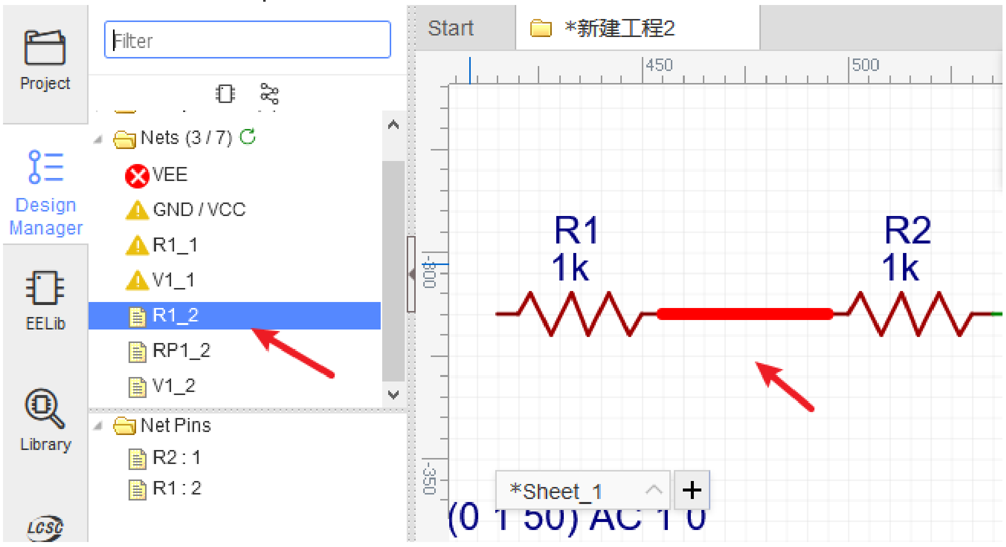
  - Net error: Prompts a red error icon.

- When Netlabel haven't connected any pins.

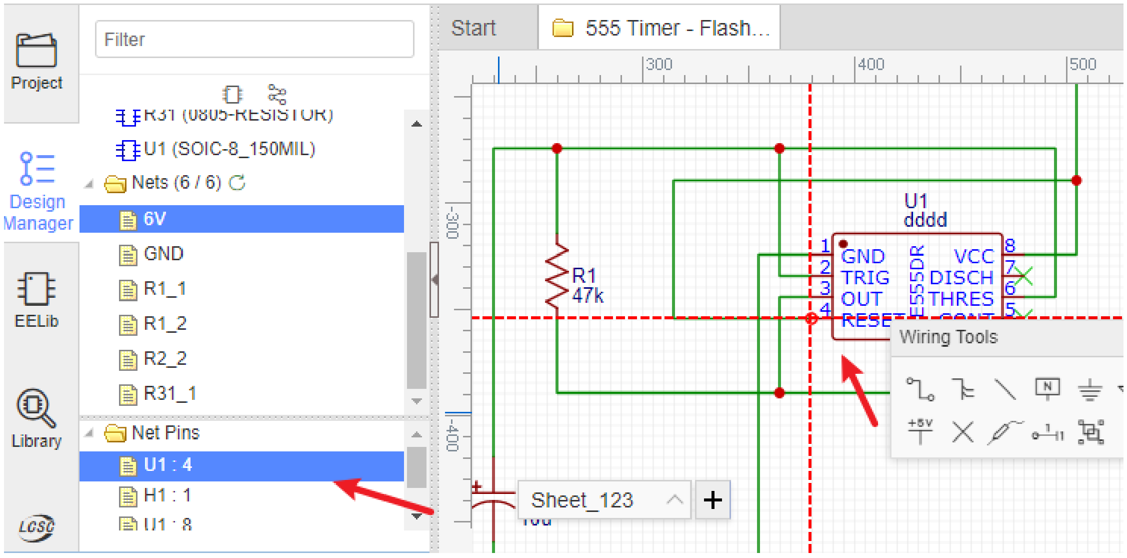


This netlabel is not connected with any pin.

- Net good: Prompt text small icon. The completed net should connect two pins and more.
  - When the net is completed.



- **Net Pins/Parts Pins**: Lists all the pins of the selected net name or components.



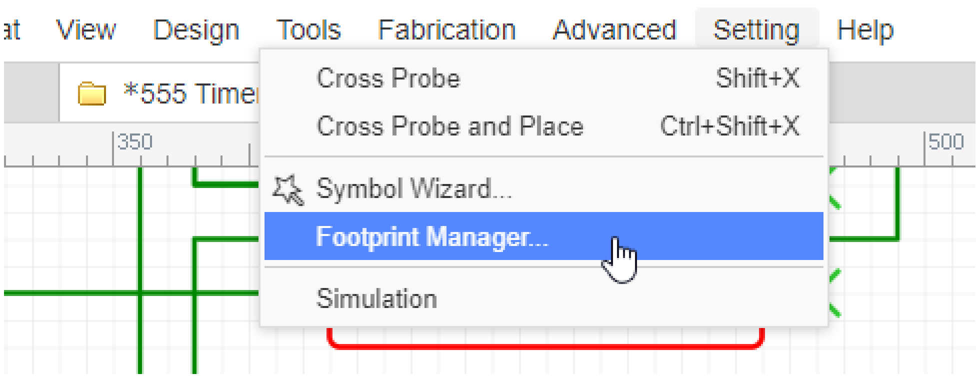
Footprint Manager

Introduction

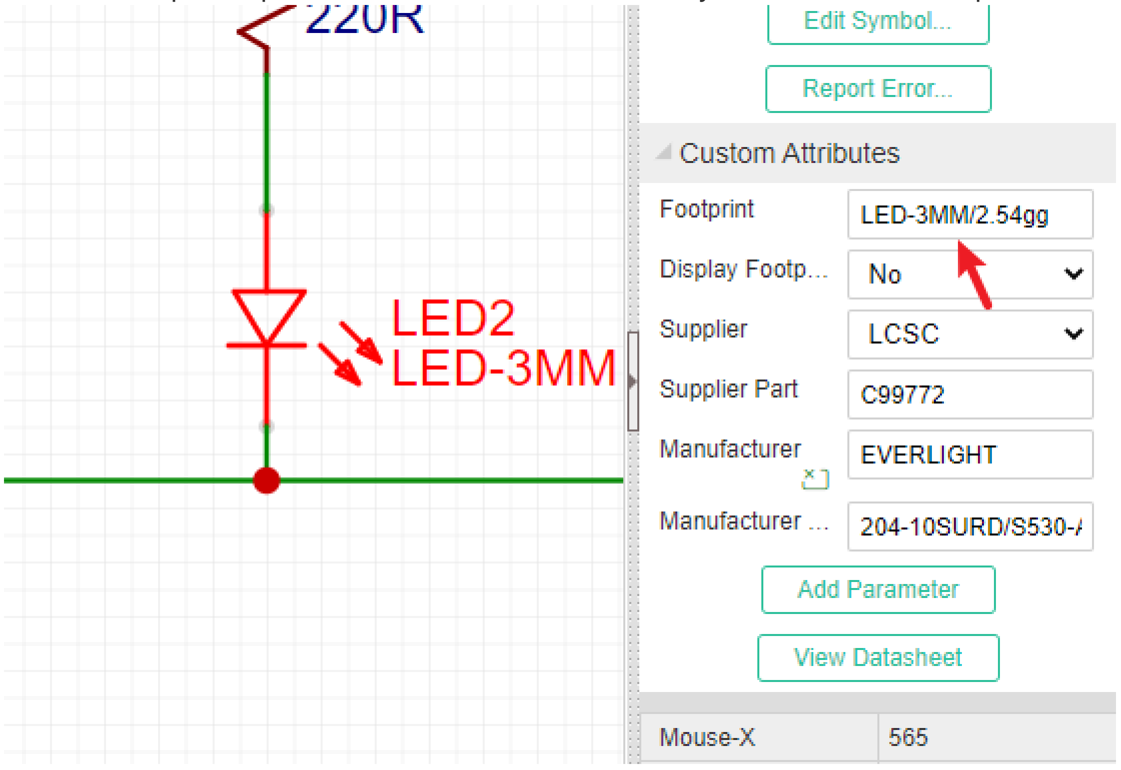
Want to batch modify components? Can't identify the corresponding relationship between component pins and footprint pins? Don't worry, EasyEDA can do this.

There are two ways to open the footprint manager:

- Click top menu, via: Top Menu - Tools - Footprint Manager



- Click the footprint input box of custom attributes when you've selected a component:



Footprint	LED-3MM/2.54gg
Display Footp...	No
Supplier	LCSC
Supplier Part	C99772
Manufacturer	EVERLIGHT
Manufacturer ...	204-10SURD/S530-/
Mouse-X	565

1. Footprint manager will check your parts footprint correct or not automatically when open it.

If the part without the footprint or this footprint doesn't exist in EasyEDA Libraries, or if the part's Pins doesn't correspond the footprint's Pads correctly, the footprint manager will show the red background alert.

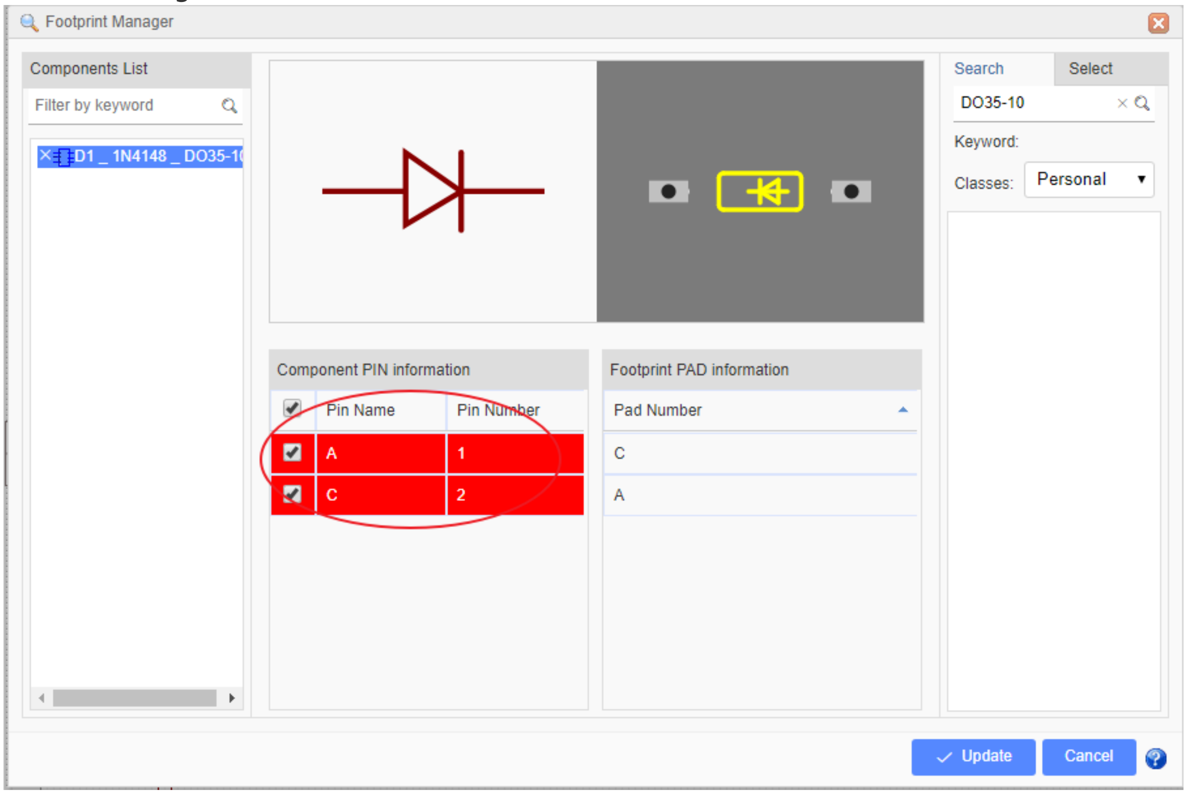
For example, If your part D1 has 2 pins,

- pin numbers are `1` and `2`,
- pin names are `A` and `C`,

but you assigned a footprint has 2 pads,

- pad number are `A` and `C`,

but the part's pin number doesn't match the pad number, so the the footprint manager will alert red background:



Component PIN information

	Pin Name	Pin Number
✓	A	1
✓	C	2

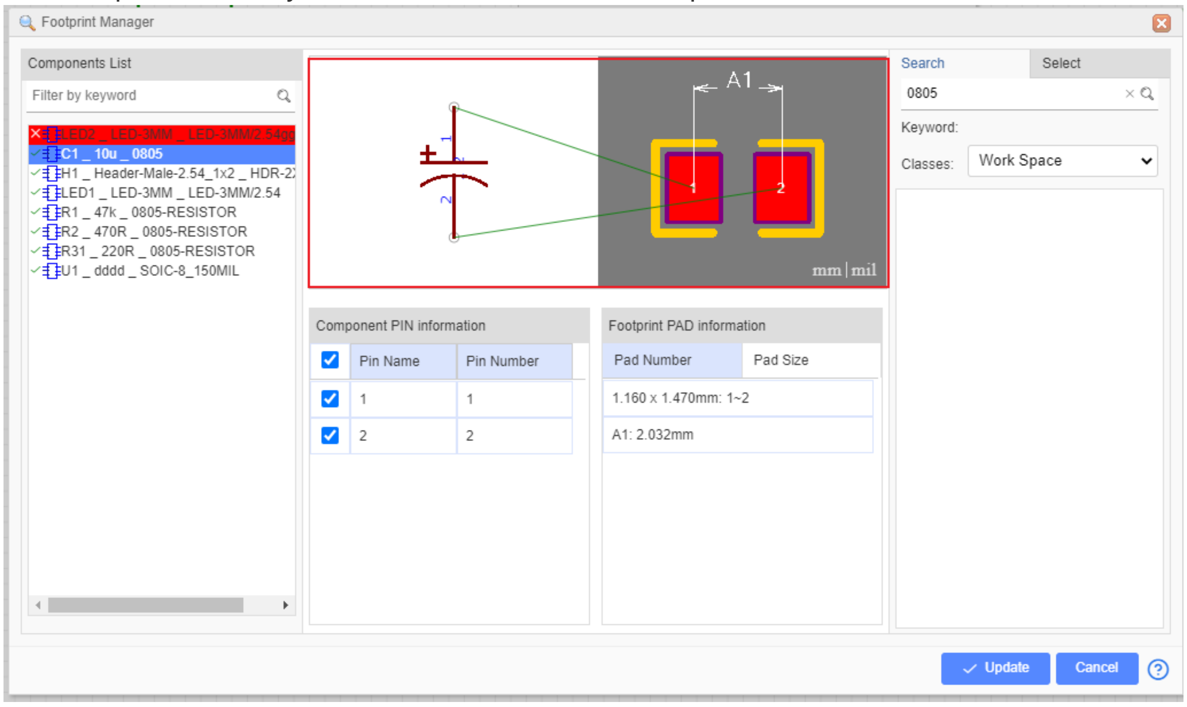
Footprint PAD information

Pad Number
C
A

In order to solve this:

- method 1: change part's pin number from 1 and 2 to A and C.
- method 2: change footprint's pad number as 1 and 2. That needs the footprint is created by you. And you can't change the Pad number in footprint manager, you need to find out the footprint at "Library > Footprints > Work Space", and then edit it.
- method 3: find an other footprint and update.

2. In the preview area, you can zoom in, zoom out and pan with mouse scroll button.


Footprint Manager
Components List
Filter by keyword
LED2 _ LED-3MM _ LED-3MM/2.54gg
C1 _ 10u _ 0805
H1 _ Header-Male-2.54_1x2 _ HDR-2
LED1 _ LED-3MM _ LED-3MM/2.54
R1 _ 47k _ 0805-RESISTOR
R2 _ 470R _ 0805-RESISTOR
R31 _ 220R _ 0805-RESISTOR
U1 _ dddd _ SOIC-8_150MIL
A1
1
2
mm | mil
Search
Select
0805
Keyword:
Classes:
Work Space
Update
Cancel

Component PIN information

✓	Pin Name	Pin Number
✓	1	1
✓	2	2

Footprint PAD information

Pad Number	Pad Size
1.160 x 1.470mm: 1~2	
A1: 2.032mm	

- **Component PIN Information**: And you can modify component's pin map information in here.
- **PCB PAD Information**:
    - **Pad Number**: You can check the footprint's pad number, but you can't modify it. when you select the component on the left side, it shows component's footprint pad number, if you selected a footprint which is searched or selected from the classes, it will show the selected footprint's pad number.
    - **Pad Size**: You can check the footprint's pads size and distance, it same as "Check Dimension" tool of footprint editor. Click the preview area unit text to change size unit.

Update footprint

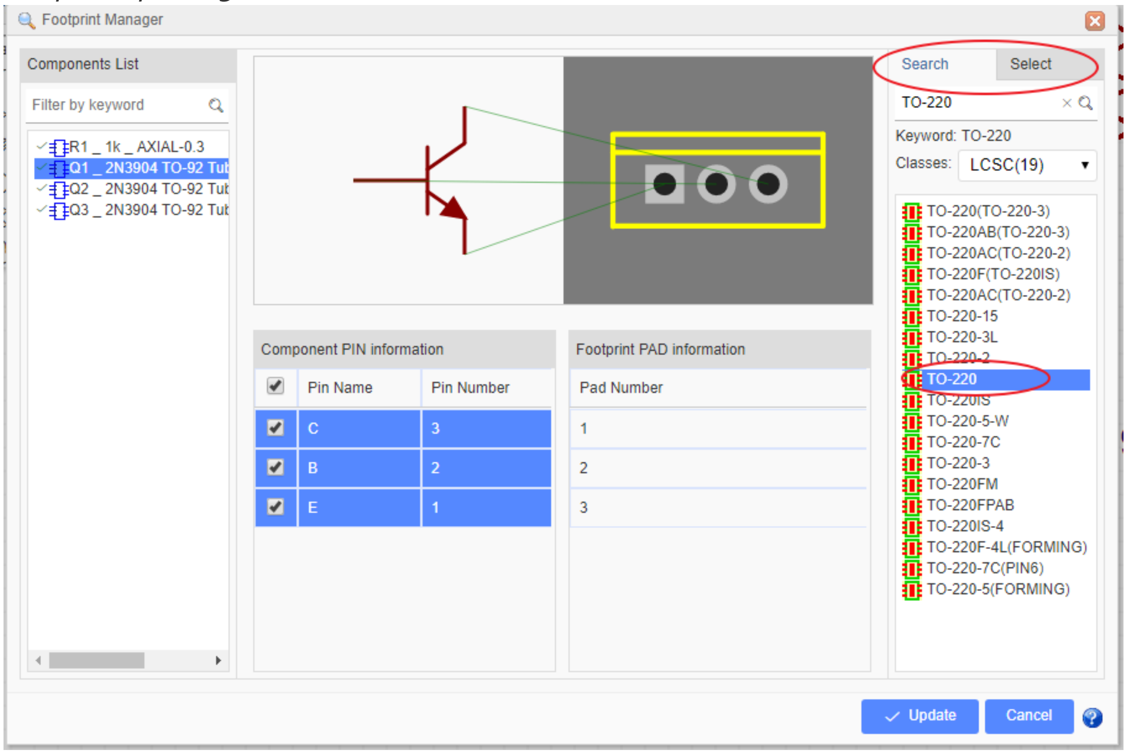
If you want to change the footprint, for example, select a component such as Q1, from **TO-92** TO **TO-220**, you just need to click in the footprint input box. EasyEDA will popup the footprint manager dialog. You can follow the instructions.

- Type **TO-220** into the search box and search, Or change to Select tab,
- Select the classes you want and select **TO-220** footprint,
- Verify it in the preview box,
- then press the **Update** button.

After that you will find you have changed the footprint to **TO-220**.

Note:

- *To ensure that you use a footprint type that is already in the EasyEDA library, it is recommended that you use this technique to change component footprints rather than just typing a footprint name directly into the footprint text input box.because of the footprint manager will add the footprint's global unique ID into the schematic when the footprint updating.*



R1 _ 1k _ AXIAL-0.3
Q1 _ 2N3904 TO-92 Tub
Q2 _ 2N3904 TO-92 Tub
Q3 _ 2N3904 TO-92 Tub

Component PIN information

✓	Pin Name	Pin Number
✓	C	3
✓	B	2
✓	E	1

Footprint PAD information

Pad Number
1
2
3

TO-220

Keyword: TO-220

Classes: LCSC(19)

TO-220(TO-220-3)
TO-220AB(TO-220-3)
TO-220AC(TO-220-2)
TO-220F(TO-220IS)
TO-220AC(TO-220-2)
TO-220-15
TO-220-3L
TO-220-2
TO-220
TO-220IS
TO-220-5-W
TO-220-7C
TO-220-3
TO-220FM
TO-220FPAB
TO-220IS-4
TO-220F-4L(FORMING)
TO-220-7C(PIN6)
TO-220-5(FORMING)

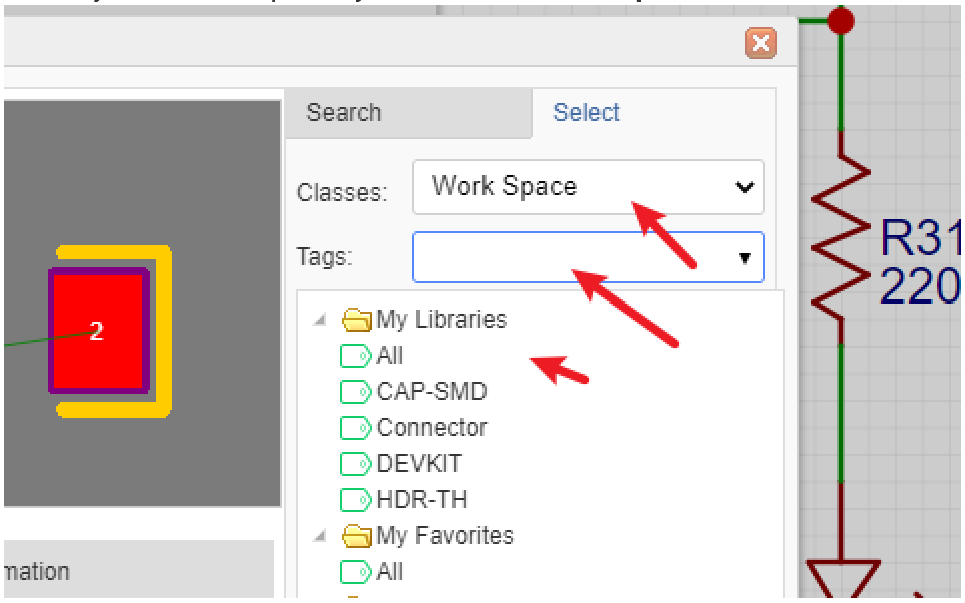
- *When you select a subpart, the others subparts will be selected too, so they will update the footprint together.*
- *If the part's property "Convert to PCB" is set as "No", it will not appear at footprint manager.*

Update in Batch

If you want to batch modify components' footprints,

- In the footprint manager dialog, you can press `CTRL + click` or `SHIFT + select` to select the components, and then select the footprint to update.
- In schematic canvas, you can frame select the commponents as you want, and then click the "footprint" attribute input box at the right-hand property panel.

To use your own footprints, you can select **Work Space** under the Select tab.



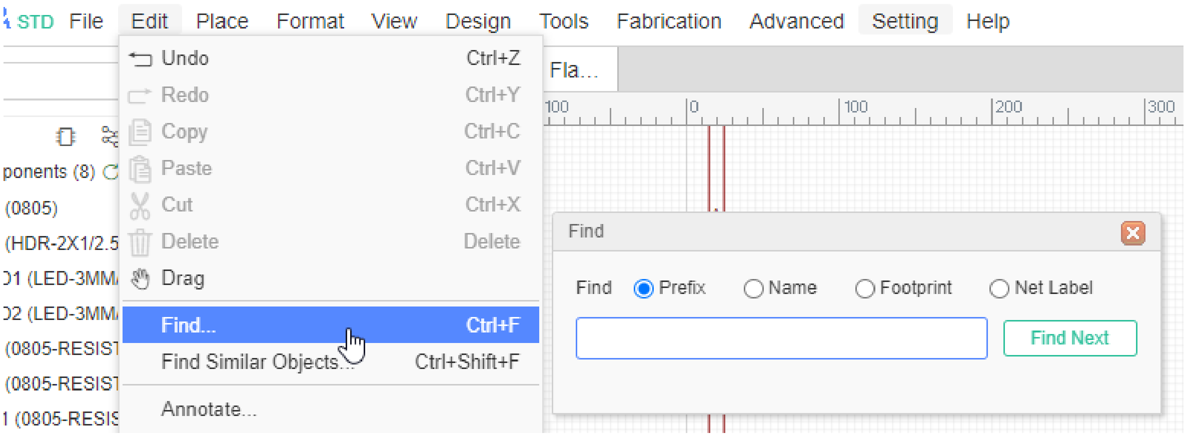
Find Similar Objects

Find Components in the Schematic

Finding individual **components** in a dense schematic can be very time consuming. EasyEDA has an easy way to find and jump to components:

Top Menu> Edit > Find...

(or `Ctrl+F`)



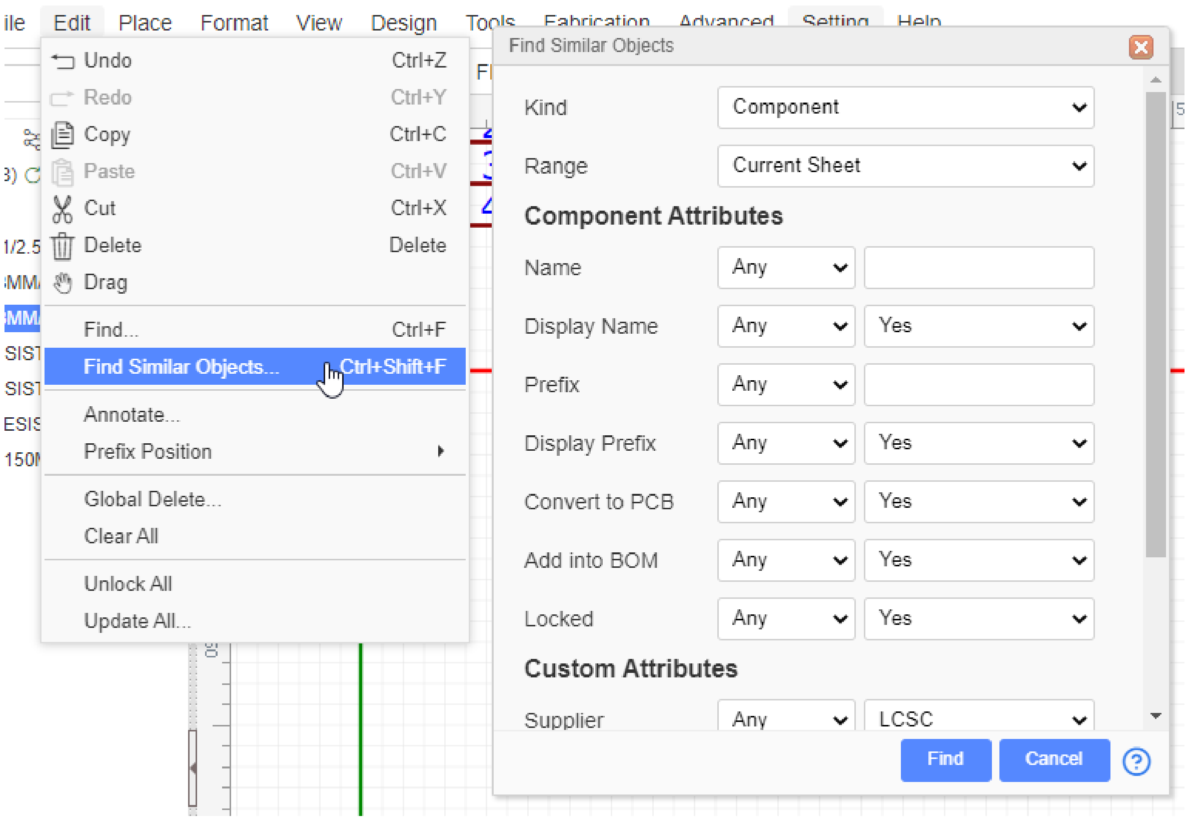
Note: *You have to click OK?in this dialog or use the Enter key.*

This feature will find, highlight and center in the window, parts by their Prefix (or reference designator). However, it cannot be used to find net names or other text in a schematic.

This is where the Design Manager comes in. the more information please refer Design Manager chapter.

Find Similar Objects

EasyEDA provide a powerful find similar tool, you can find what you want very easily.
Via **Top Menu > Edit > Find Similar Objects...**



Find Similar Objects

Kind	Component	
Range	Current Sheet	

Component Attributes

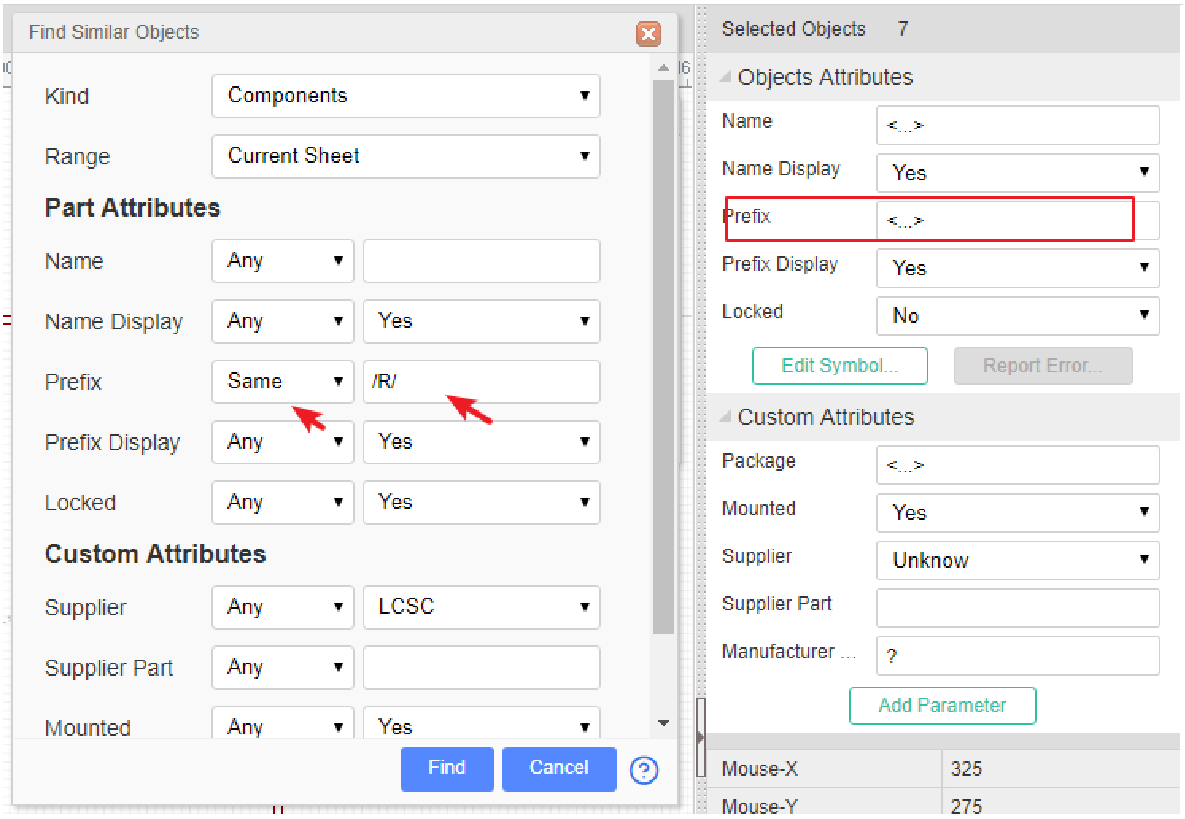
Name	Any	
Display Name	Any	Yes
Prefix	Any	
Display Prefix	Any	Yes
Convert to PCB	Any	Yes
Add into BOM	Any	Yes
Locked	Any	Yes

Custom Attributes

Supplier	Any	LCSC

Kind: Select the object what you want to find.
Range: This option only for the schematic, you can find the object for current sheet or all sheets.
Find Parameters: Any: Find any objects; Same: Only find the object which attribute same as this attribute. Different: Find the object which attribute is different than this attribute.

The input box support the Js Regular Expression, you can type `/keyword/` to find what you want, such as find all prefix which are including "R":


Find Similar Objects

Kind	Components	
Range	Current Sheet	

Part Attributes

Name	Any	
Name Display	Any	Yes
Prefix	Same	/R/
Prefix Display	Any	Yes
Locked	Any	Yes

Custom Attributes

Supplier	Any	LCSC
Supplier Part	Any	
Mounted	Any	Yes

Selected Objects 7

Objects Attributes

Name	<...>
Name Display	Yes
Prefix	<...>
Prefix Display	Yes
Locked	No

Custom Attributes

Package	<...>
Mounted	Yes
Supplier	Unknow
Supplier Part	
Manufacturer ...	?

Mouse-X	325
Mouse-Y	275

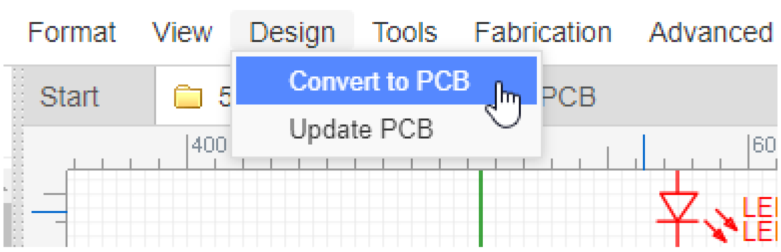
After click the "Find" button, all the siutable objects will be seleted, and the right-hand panel will show all the attributes, the different attributes will show as the `<...>`, you can change the attributes directly, and they will apply to all selected objects.

The find similar objects only support to find a part of custom attributes. Such as footprint, suppiler etc.

Convert Schematics to PCB

Convert to PCB

Most of the time, schematics are created with the aim of producing a PCB. So how do you convert your schematic to a PCB in EasyEDA? You just need to to click the PCB icon on the toolbar with the title **Convert to PCB**.

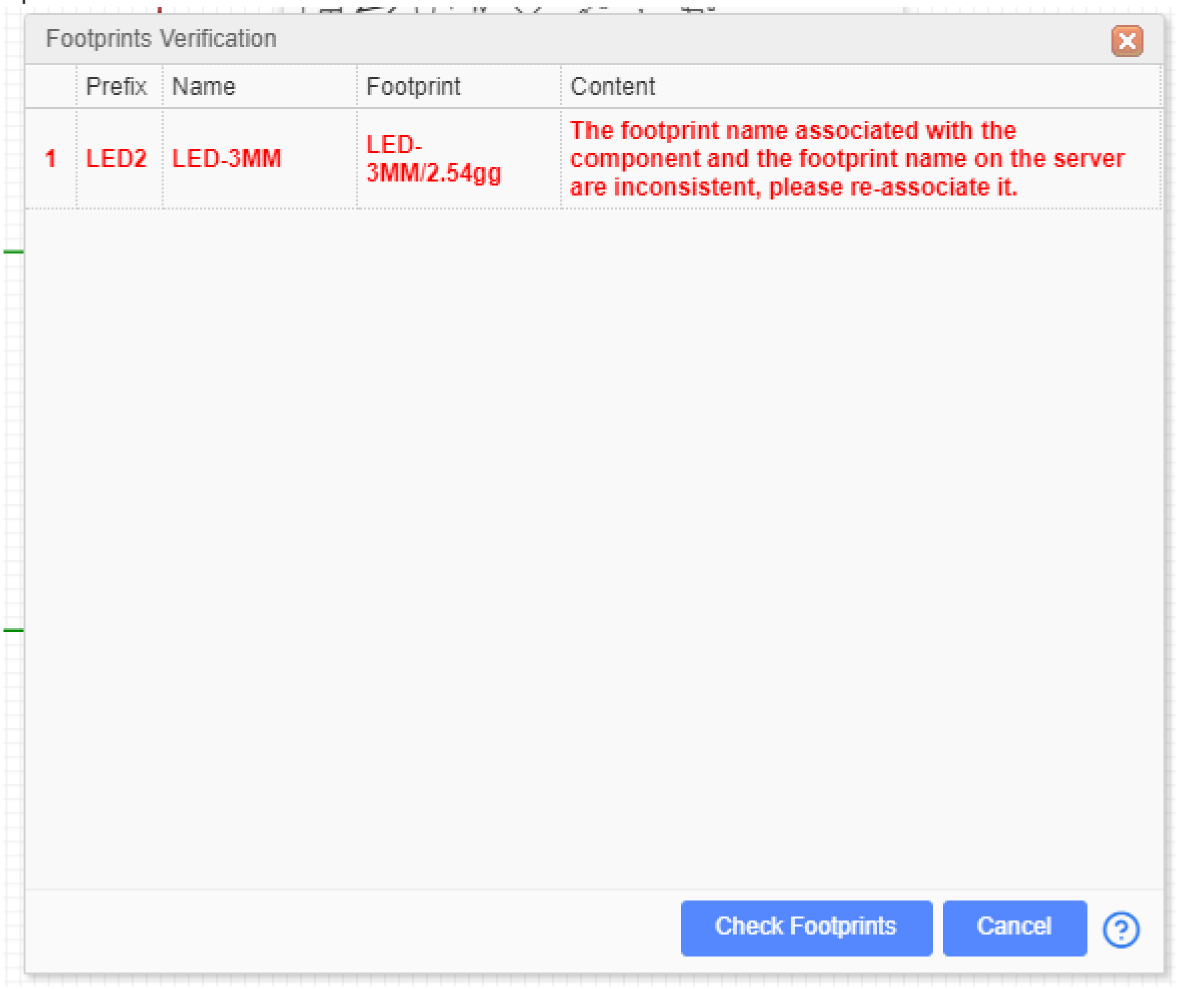


Note:

- *Before converting, you need to use the Design Manager and Footprint Manager to check all the components, nets(connection) and footprints to ensure no errors exist.*

Footprints Verification

After clicking the **Convert to PCB** button, if the project has errors the following dialog will open:



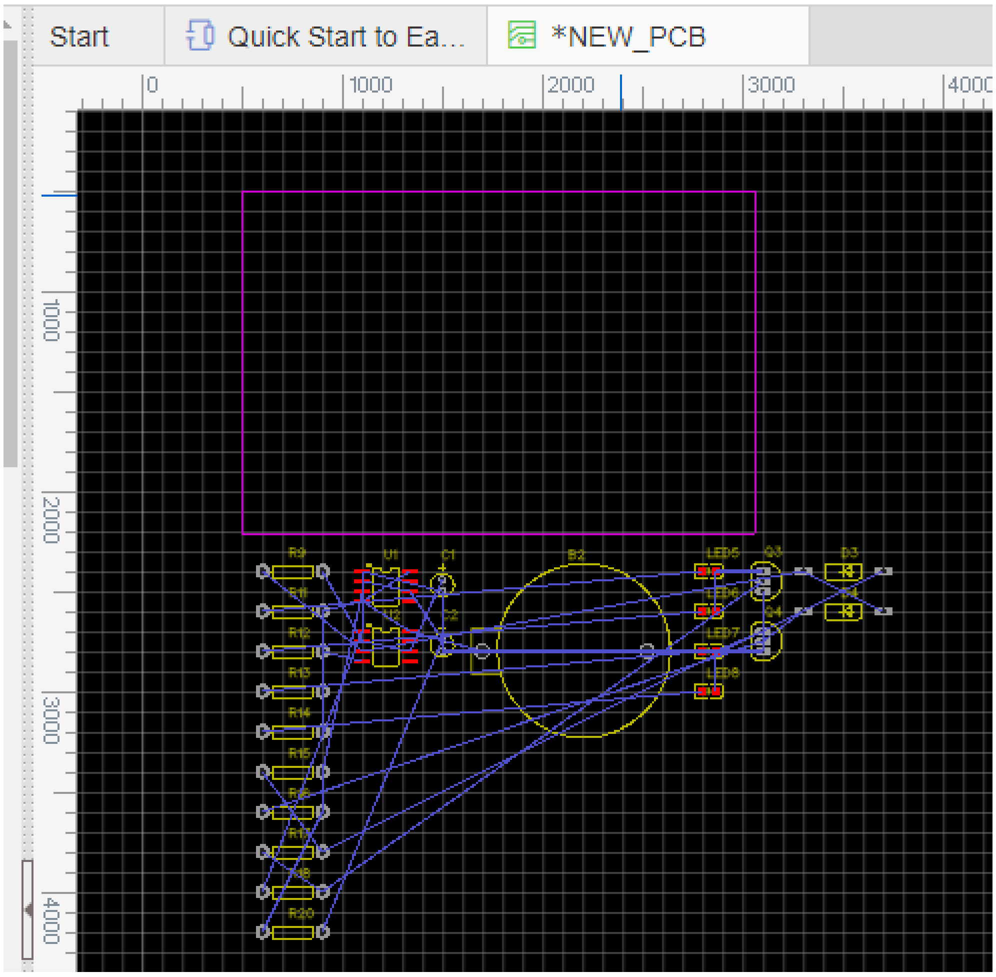
	Prefix	Name	Footprint	Content
1	LED2	LED-3MM	LED-3MM/2.54gg	The footprint name associated with the component and the footprint name on the server are inconsistent, please re-associate it.

The row in red indicates that EasyEDA can't find a PCB footprint matching the footprint that the schematic symbol is calling for.

This could be because you have made an error entering the footprint attribute in the symbol's Properties or maybe you haven't yet created a PCB footprint for the footprint that your symbol is calling for.

In this case the footprint should have been **AXIAL-0.3** but instead it is empty. To correct it you can click on the row and update the footprint **AXIAL-0.3** for it at the footprint manager.

After making any necessary corrections, click the **Convert to PCB** button and EasyEDA will automatically load all the PCB footprints into the PCB editor as shown in the image below.



This shows the footprints placed in arbitrary positions with the connections between them shown as blue Rat lines.

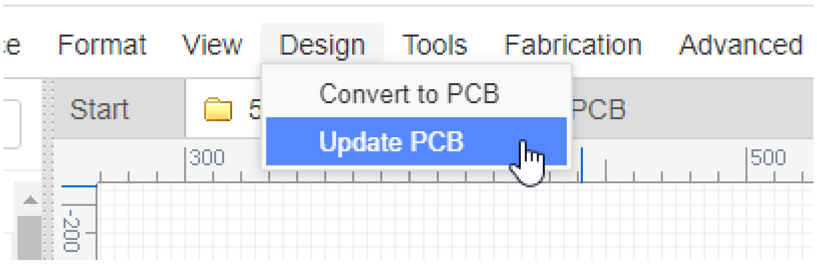
Invalid footprint

The footprint's PAD number is different from the symbol's PIN number, e.g. the diode footprint's PAD numbers are A,C but the symbol's PIN numbers are 1,2. You just need to change one to fit the other. It is case sensitive!

the changing method please refer the **Schematic - Footprint Manager** section.

Update PCB

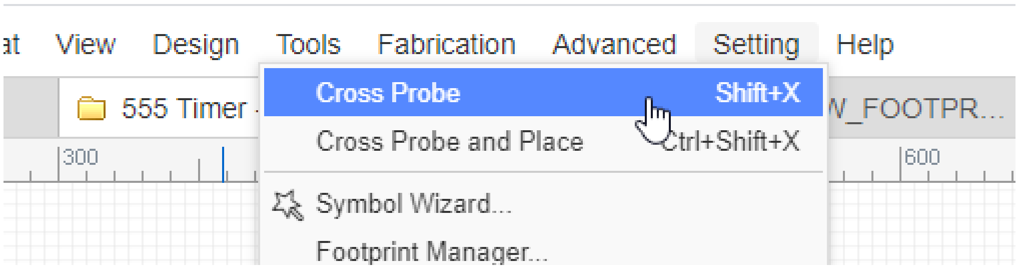
Converting a schematic to PCB can be done using the `Convert to PCB...` , but if you do modifications to the schematic, by using the `Update PCB` button you can immediately be passed forward to update the selected PCB without having the PCB editor window already open or without creating a new PCB file.



or you can use "Top Menu - Design - Import Changes" at PCB editor.

Cross Probe

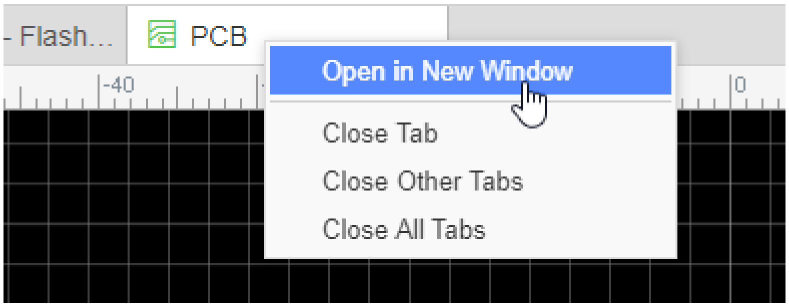
This tool is used to cross probe from chosen objects on the current schematic to its corresponding counterparts in the PCB, or from PCB Footprints to corresponding counterparts in the schematic.



Since v6.4.0, EasyEDA supports multiple windows design to cross probe.

How do it works?

1. Open schematic and PCB
2. Right-click the schematic or PCB tab, click "Open in New Window"



3. It will open this document in new window, then you can do the cross probe: Click the component, click the Design Manager list, the "Cross Probe and Place" works too.

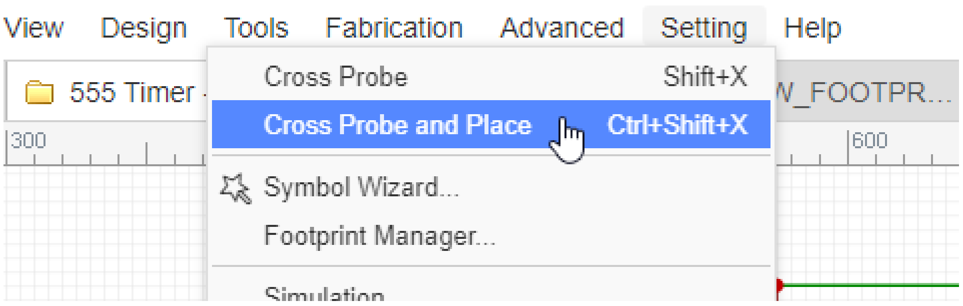
Note:

- *You need to open PCB first before using cross probe in the schematic. And don't forget to use the hotkey `SHIFT+X`.*
- *After converting the schematic to PCB, for using this function please save the PCB first.*
- *If your project has many PCBs, when you use the cross probe please open the PCB what you need manually.*

Cross Probe And Place

If your schematic have a lot of components, it will be difficult to layout the PCB , so EasyEDA provides a powerful function "Cross Probe And Place".

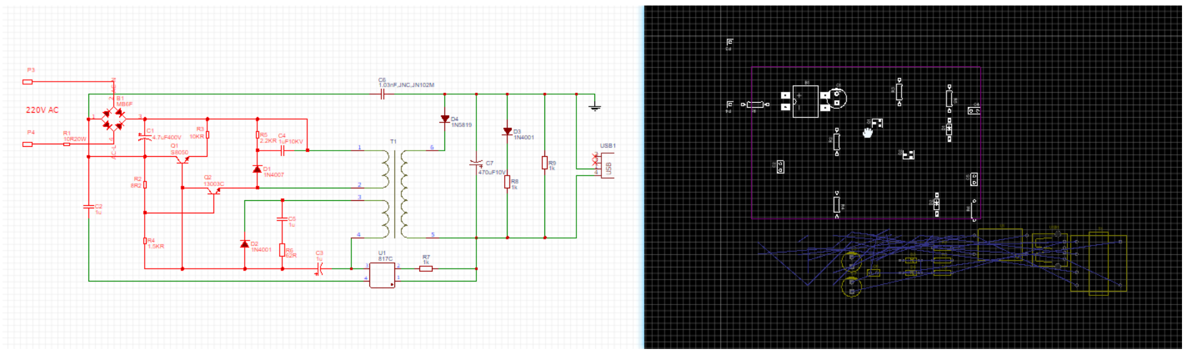
Top Menu> Tools > Cross Probe And Place



Cross Probe And Place will make the footprints' location match the schematic's parts' location as much as it possibly can.

How to use:

- Convert the schematic to PCB first, and save at current project.
- Frame select the components area by mouse in the schematic, and then click the "Cross Probe And Place", hotkey "CTRL + SHIFT + X".
- The editor will switch to the PCB, and choose the footprints as you selected for waiting for placing.
- Right click to place, and the mouse will keep the drag status, its easy for adjusting the footprints' location.



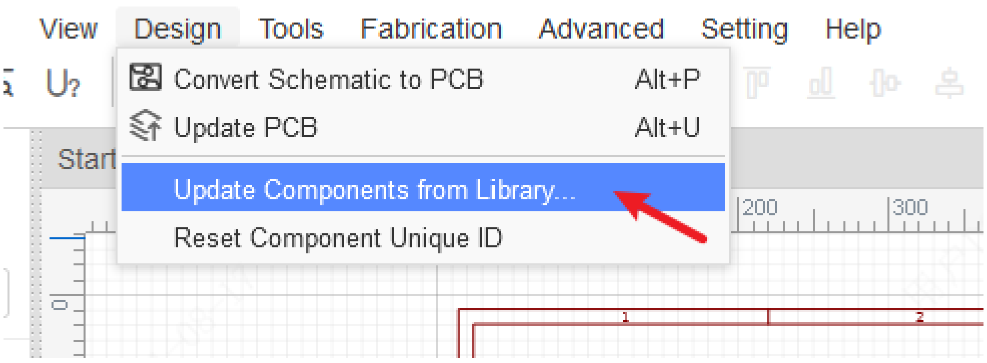
Notice:

- *You need to open PCB first before using this function in the schematic*

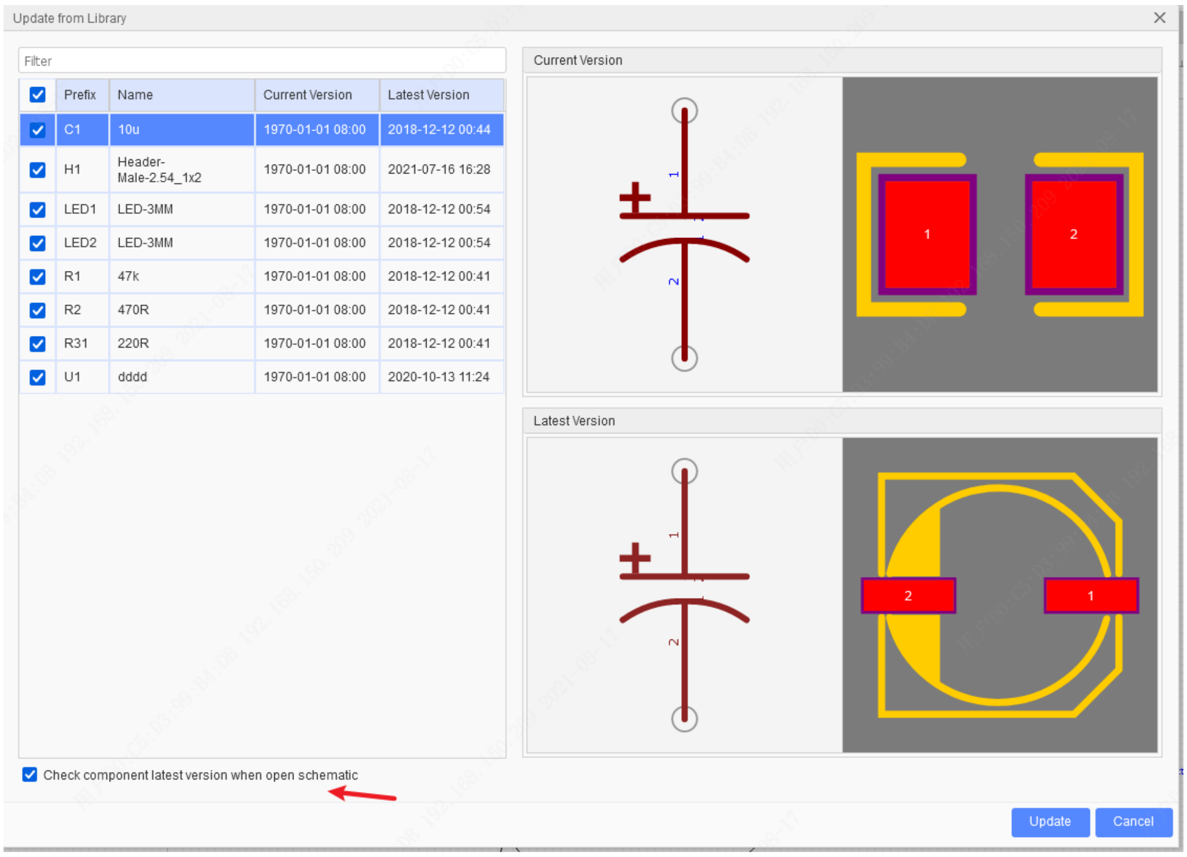
Update Components from Library

If you want to update the component at schematic when you update the parts at Library, can use this feature.

Via: Top menu - Design - Update Components from Library


View
Design
Tools
Fabrication
Advanced
Setting
Help
Convert Schematic to PCB Alt+P
Update PCB Alt+U
Update Components from Library...
Reset Component Unique ID

Click the menu you will see the update dialog, you can preview the current components and compare with latest version.



✓	Prefix	Name	Current Version	Latest Version
✓	C1	10u	1970-01-01 08:00	2018-12-12 00:44
✓	H1	Header-Male-2.54_1x2	1970-01-01 08:00	2021-07-16 16:28
✓	LED1	LED-3MM	1970-01-01 08:00	2018-12-12 00:54
✓	LED2	LED-3MM	1970-01-01 08:00	2018-12-12 00:54
✓	R1	47k	1970-01-01 08:00	2018-12-12 00:41
✓	R2	470R	1970-01-01 08:00	2018-12-12 00:41
✓	R31	220R	1970-01-01 08:00	2018-12-12 00:41
✓	U1	dddd	1970-01-01 08:00	2020-10-13 11:24

✓ Check component latest version when open schematic

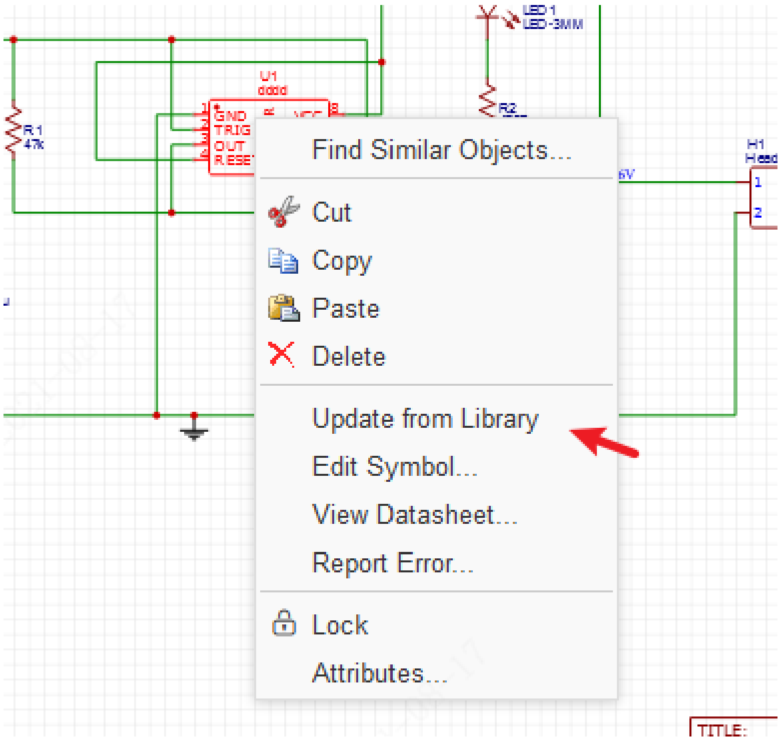
You can setting check the component latest version when open the schematic at bottom option.

Before update components, please check the parts shape, pin number, footprint carefully.

PCB has this feature too at Design menu, you can update footprint from Library.

Notice：
Since v6.4.20.7, while placing the component at the schematic, that will keep the symbol and footprint corresponding at that time, no matter you update your footprint or not, it will not impact by latest footprint as previous editor version. when you import changes, the footprint will use at that time version, will not use the latest footprint version, if you want to use the latest footprint, you need to update component first.

If you want to update single one compoennt, you can right-click it and update

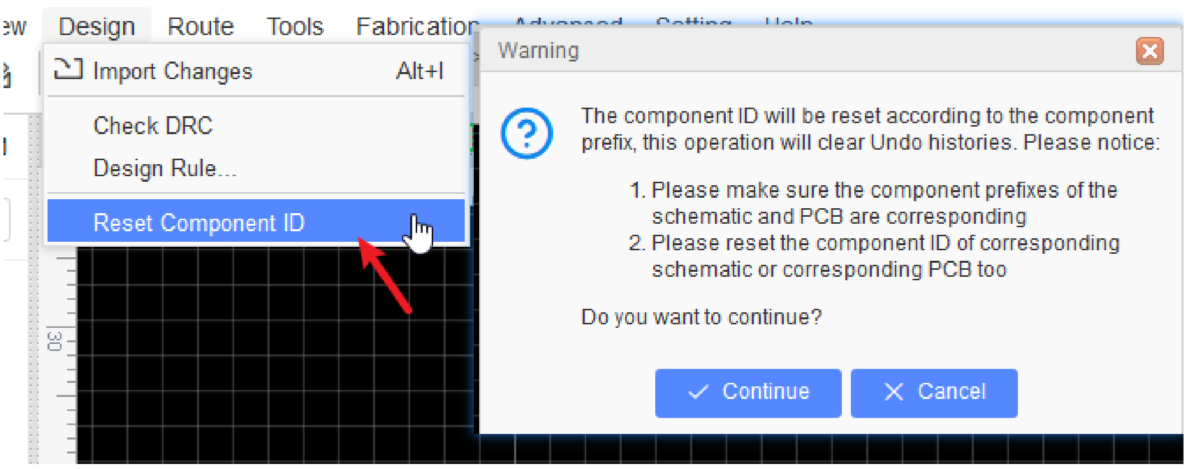


Reset Component ID

This function resets the ID of the component.

Before V6.4.7, EasyEDA was created by using component prefix to correspond schematic components to PCB components. This method would lead to the situation that after the schematic component was modified with component prefix, the old method would deleted and add new component when imported and updated into PCB, which would affect the original layout of components.
Since V6.4.7, the component ID is used for matching associations, so old files or imported third party EDA files can use this feature to reset the component ID so that the two IDs match.

Via: Top Menu - Design - Reset Component ID



The component ID will be reset according to the component prefix, this operation will clear Undo histories. Please notice:

1. Please make sure the component prefixes of the schematic and PCB are corresponding
2. Please reset the component ID of corresponding schematic or corresponding PCB too

Do you want to continue?

Notice:

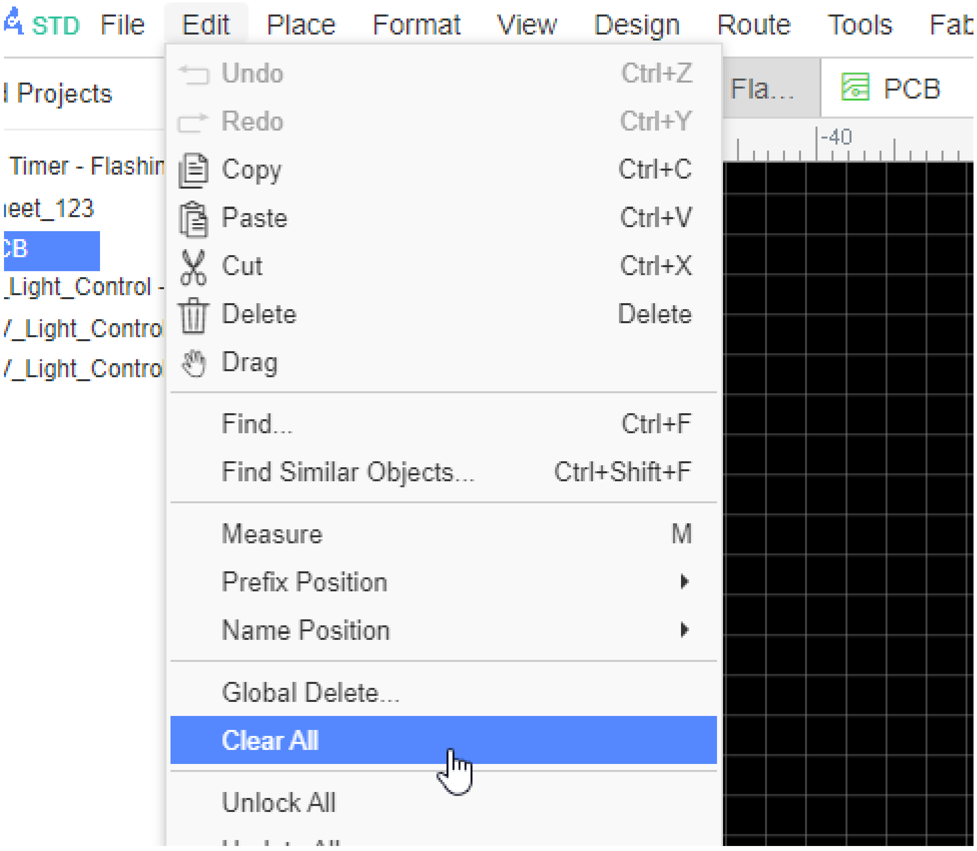
- 1. Make sure that the schematic diagram and the component prefix of the PCB match. The reset ID is reset based on the component prefix.

- 2. It is necessary to reset the component ID on both schematic diagram and PCB so that the component ID on both sides can match one by one.
- 3. For schematic diagram with subparts, it is necessary to change the component prefix of the subparts of PCB pair to U1.1 or other subparts prefixes before resets the ID, otherwise this component will still be deleted and replaced with a new one. Subsequent releases will solve this issue.

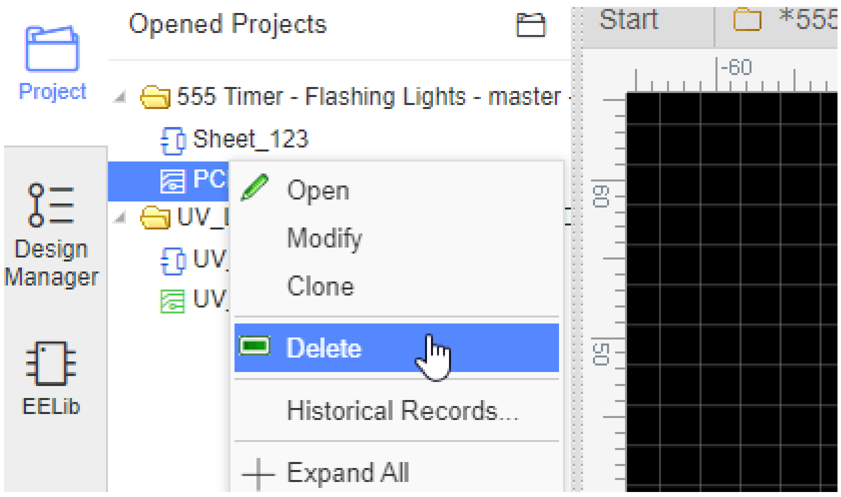
Global Delete

If you feel your schematic or PCB is mess up, need delete objects in batch, you can:

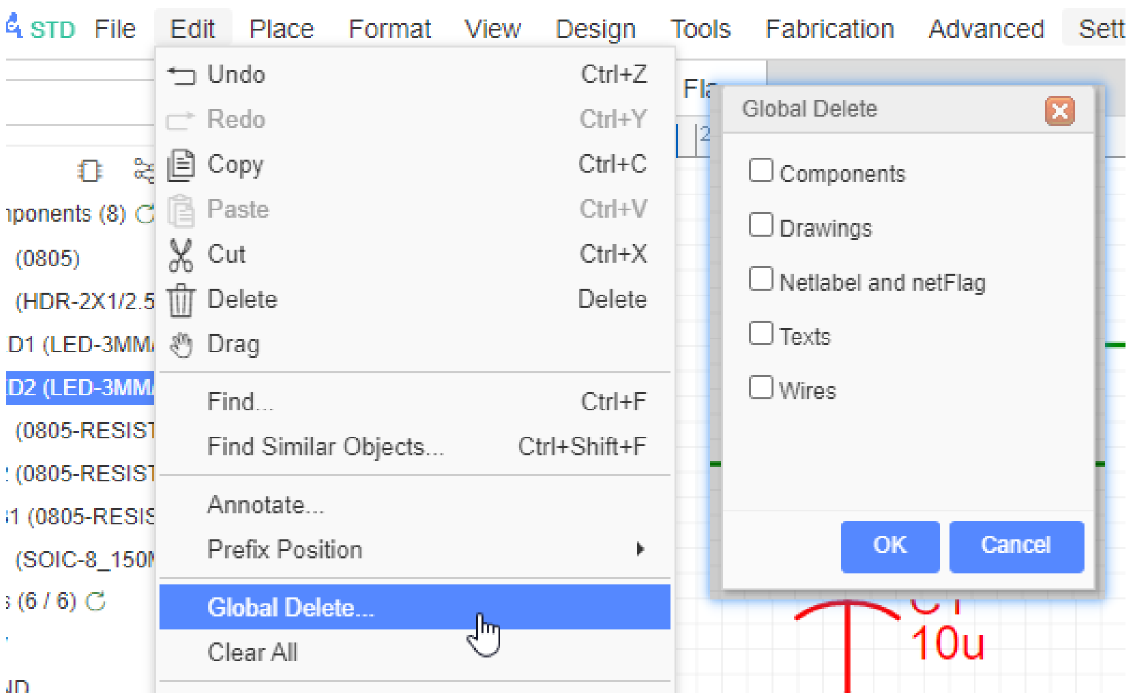
- **Top Menu** > **Edit** > **Clear All**, or CTRL + A select all and then press Delete key.



- Delete the document and create a new one.



- Using **Top Menu** > **Edit** > **Global Delete**, just delete what you want.

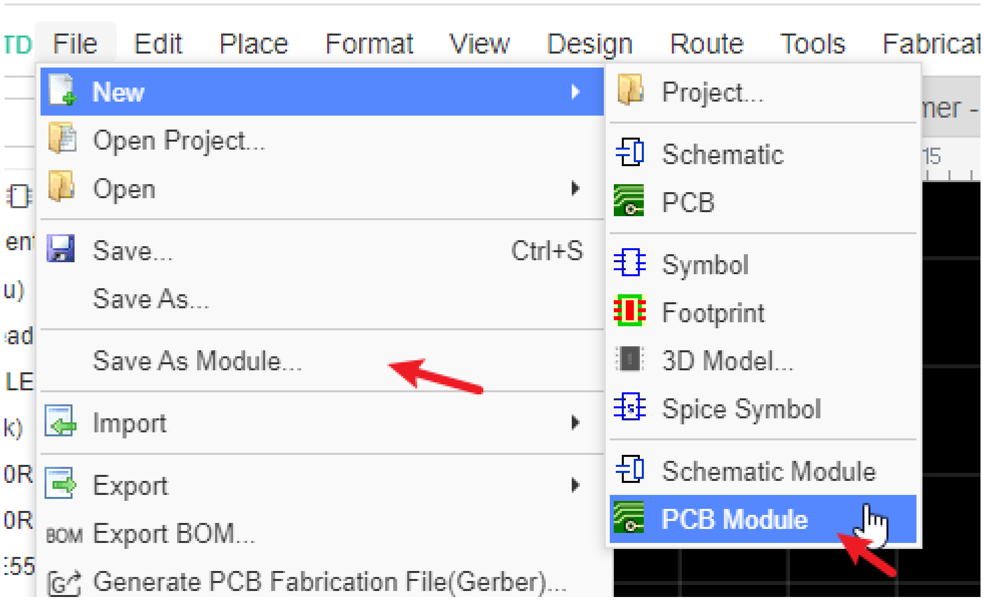


Schematic Modules

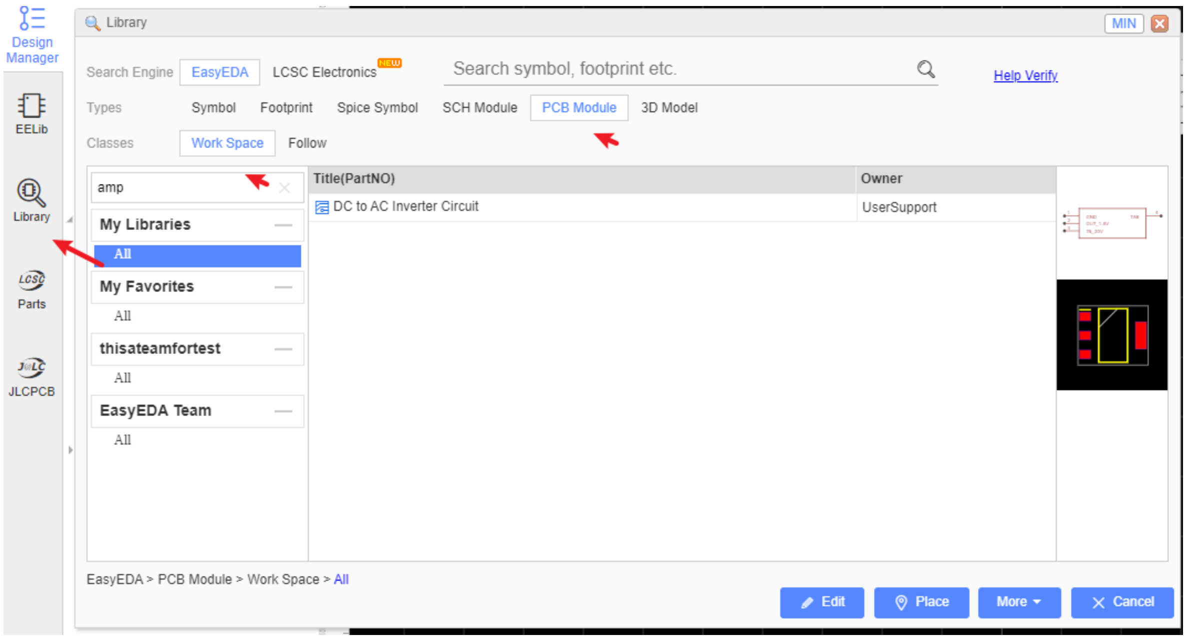
Copying codes is an easy job for coders, now copying and reusing a schematic or PCB is easy. Take a power supply unit for example, you can save this unit as a schematic module.
Via **File** > **Save as Module**:
EasyEDA support create the PCB modules, it seems schematic module.

How to Create

Via: **Save as Module** and **File** > **New** > **Schematic/PCB Module**。



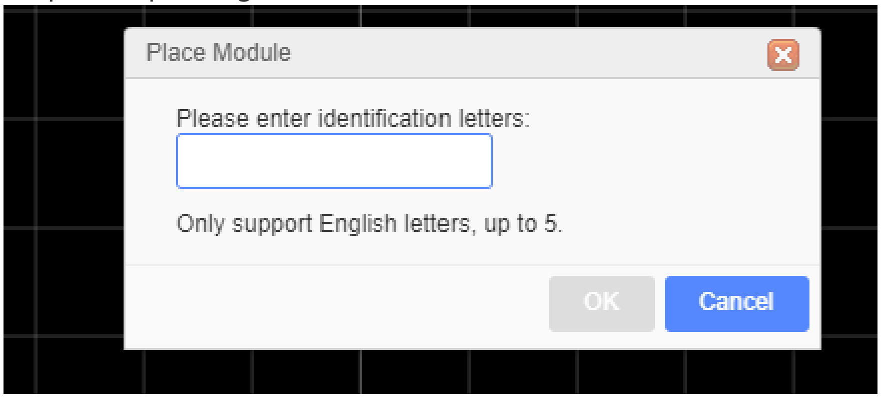
PCB module save at **Library** > **Schematic/PCB module** > **Work Space** > **My Libraries**



How to use

Since v6.4.3, after placing schematic modules and PCB modules, after Import Changes, supports to keep the layout location.

How to use:

1. Draw schematic modules and PCB modules, and ensure that their component prefix are one to one, and the footprint is also corresponding. The module's component prefix can not have question marks and duplicate prefix, such as U? or two R1.
2. Open schematic and PCB at a same project.
3. Open "Library", select the module.
4. Click the "Place" button to place the previous saved schematic module and PCB module.
5. It will pop up a window to enter English letter. The letter of schematic module should keep corresponding with PCB modules.
   
   Place Module
   Please enter identification letters:
   Only support English letters, up to 5.
   OK
   Cancel
   For example: A component at schematic module is U2, enter letter K, press OK to place into canvas, it will be KU2, then PCB module has KU2 too.
   Click "OK" and enter the placement mode. After each placement, the pop-up will continue to enter the identification letter. Make sure that the identification letters entered each time are unique.
6. When finish the module place, the PCB component unique ID will same as Schematic component unique ID, then after Import Changes, the component's location will be keep. and you can update the track's net follow the schematic netlabel too.

That implement the multipe chanel placing.

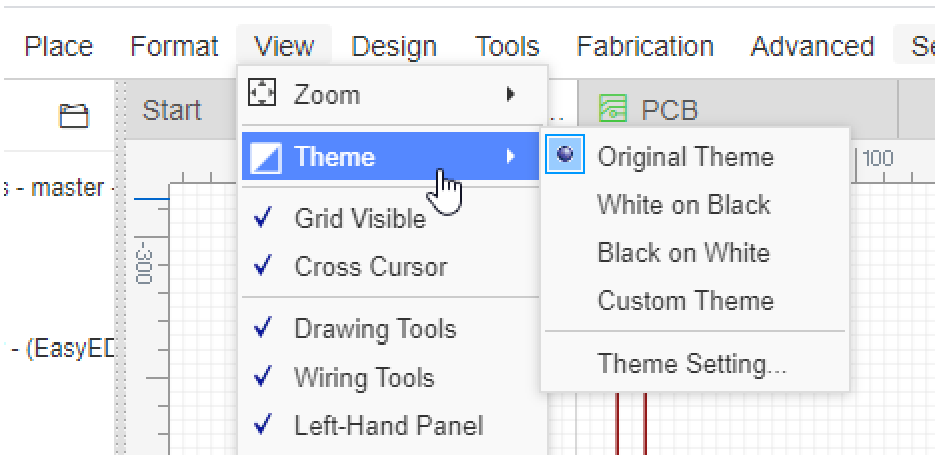
Notice:

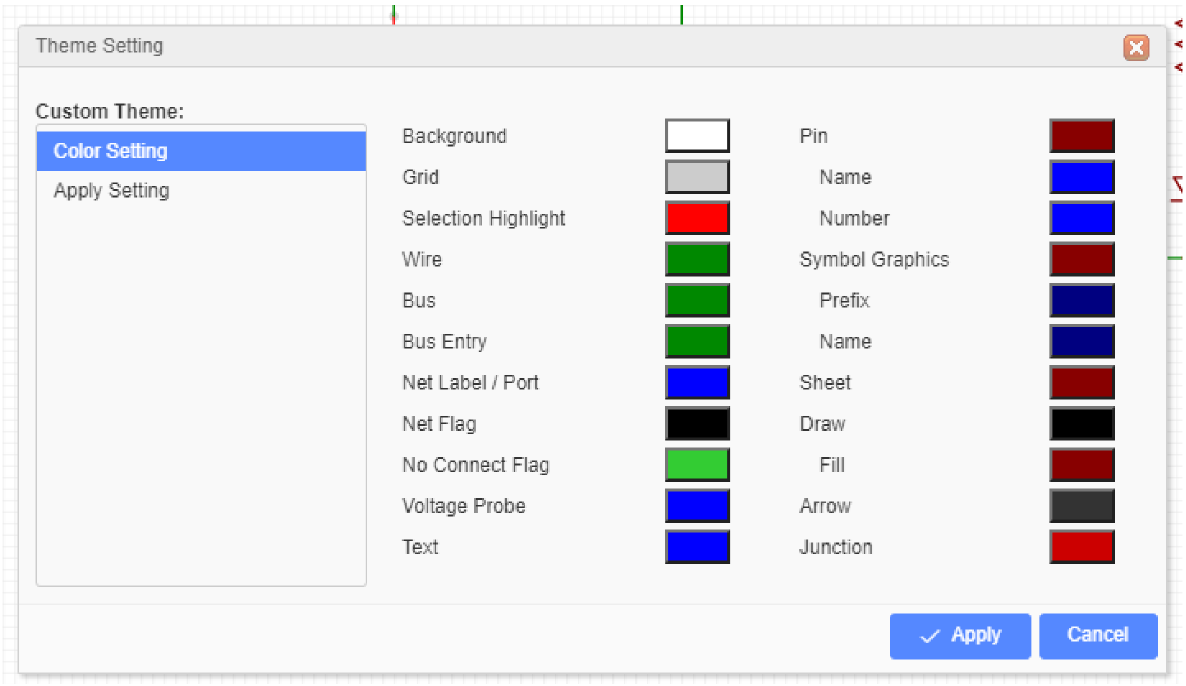
- Module composes by tracks and components, it doesn't same as symbol binding footprint, the schematic module can not binding PCB module, after placing, the module will be separated by many objects, only the symbol and footprint can be corresponding via component ID, that is why you need to make the identication letter unique for placing each time to make sure schematic module corresponding with PCB module.

Schematic Theme

EasyEDA support a powerful theme feature for the schematic design.

Via: Top Menu - View - Theme.





Background	⬜	Pin	🟥
Grid	⬜	Name	🟦
Selection Highlight	🟥	Number	🟦
Wire	🟩	Symbol Graphics	🟥
Bus	🟩	Prefix	🟦
Bus Entry	🟩	Name	🟦
Net Label / Port	🟦	Sheet	🟥
Net Flag	⬛	Draw	⬛
No Connect Flag	🟩	Fill	🟥
Voltage Probe	🟦	Arrow	⬛
Text	🟦	Junction	🟥

Original Theme: The default theme, only works for the new part placing.

White on Black: White on Black, the objects will be white, the background will be black.

Black on White: Black on White.

User Defind: When change to this theme style, the schematic will follow your theme options "My theme".

My Theme: Custom theme, which is stored locally in the browser and it will be synchronized to the server. When click apply, this theme will be applied to the current schematic. Next time you open the schematic, the theme of the schematic will be a custom theme.

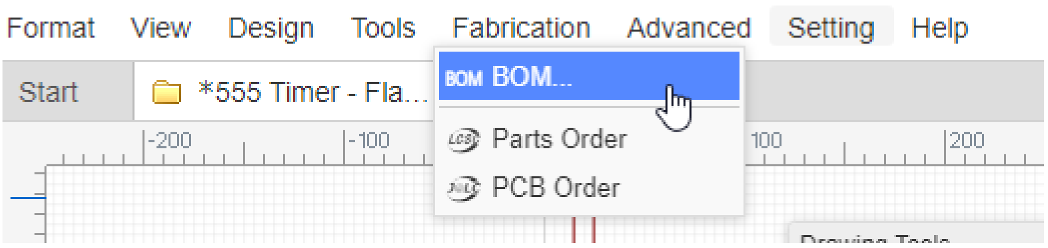
My theme Settings: You can apply "My theme" on: 1. Creating New Schematic, 2.Opening Existed Schematic.

If you used any theme for the schematic, you need to UNDO to go back previous color theme. The "Original Theme" can't help.

Your schematic theme will synchronized to the server by default.

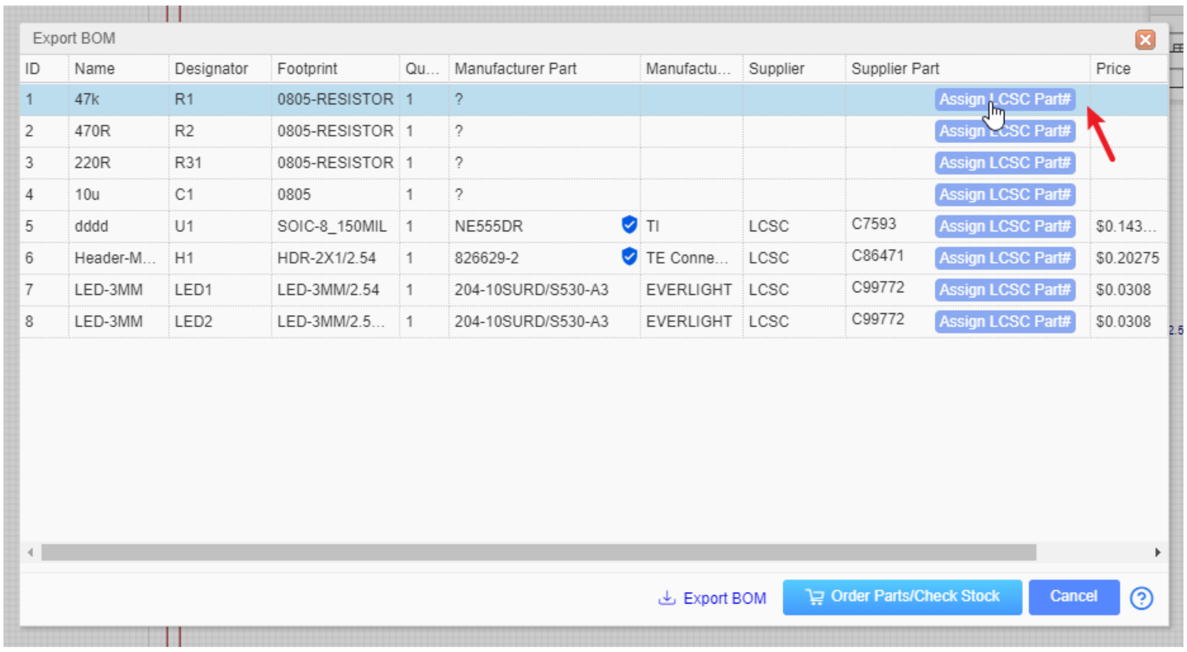
Export BOM

You can export the Bill of Materials (BOM) for the schematic (Document) and PCB, via: "Top Menu - File - Export BOM", or "Top Menu - Fabriaction - BOM".



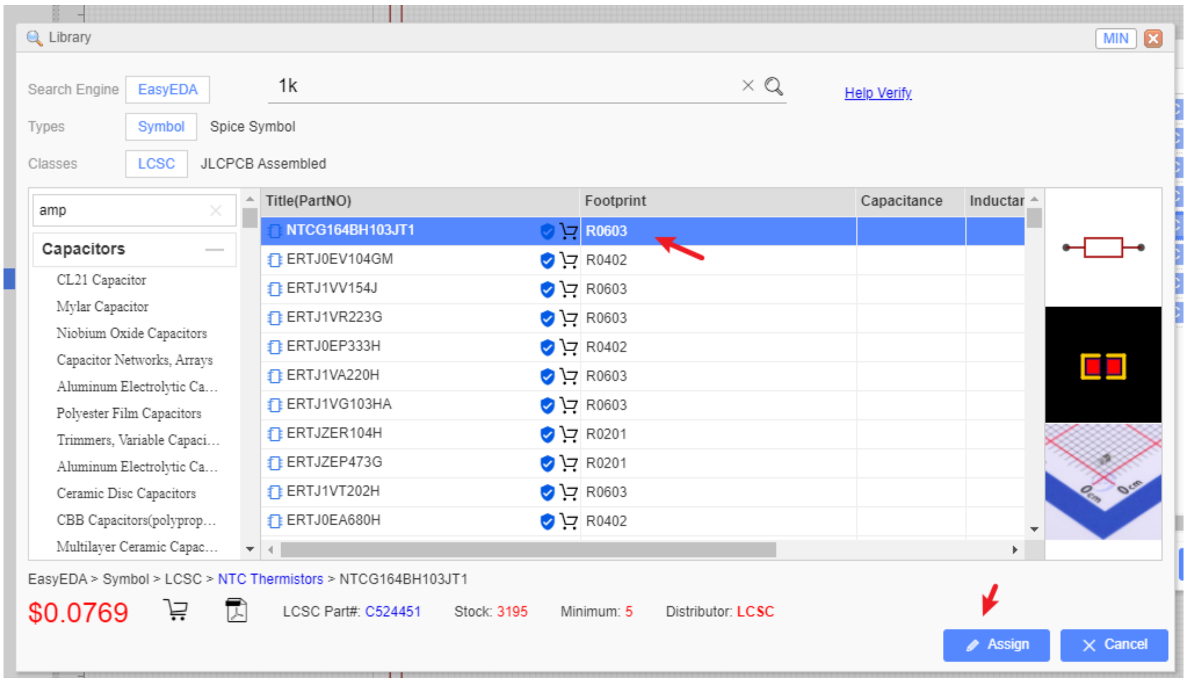
After clicking the BOM export option, the dialog below will open.

In this dialog, you can click the buttom to assign LCSC part's order code for your components.


Export BOM

ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part	Price
1	47k	R1	0805-RESISTOR	1	?			Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?			Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?			Assign LCSC Part#	
4	10u	C1	0805	1	?			Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593 Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471 Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772 Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772 Assign LCSC Part#	$0.0308

After clicking on the assign icon， the components and footprints search dialog will pop up, and you can choose which component you want to assign.


Library

Search Engine: EasyEDA
1k
Help Verify
Types: Symbol | Spice Symbol
Classes: LCSC | JLCPCB Assembled

amp

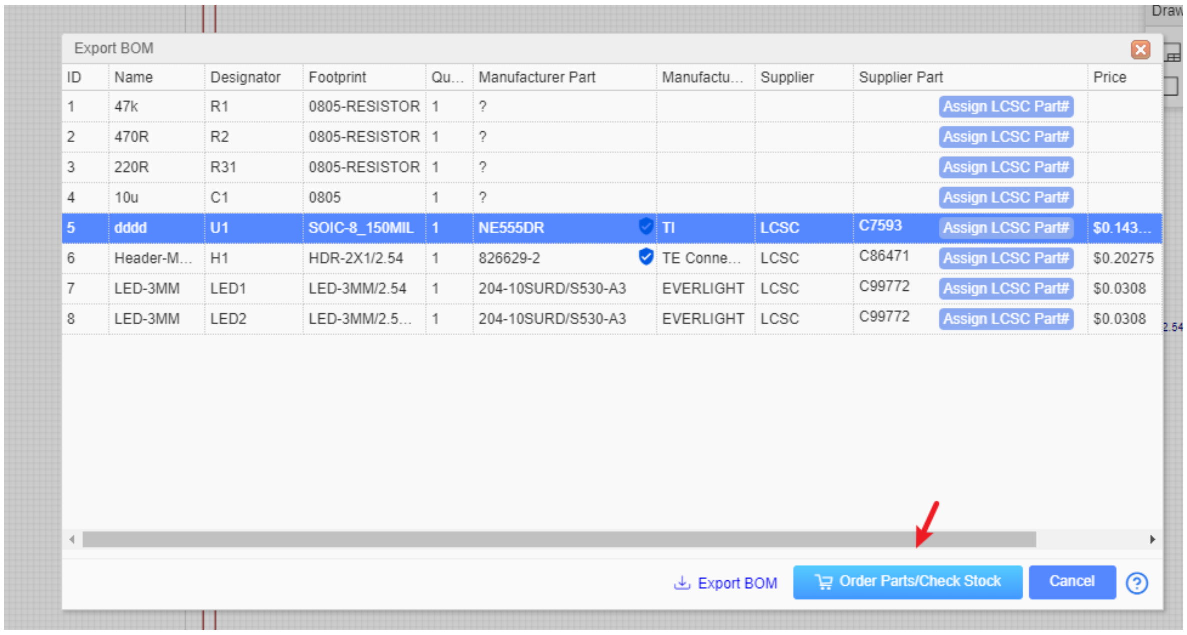
Capacitors
- CL21 Capacitor
- Mylar Capacitor
- Niobium Oxide Capacitors
- Capacitor Networks, Arrays
- Aluminum Electrolytic Ca...
- Polyester Film Capacitors
- Trimmers, Variable Capaci...
- Aluminum Electrolytic Ca...
- Ceramic Disc Capacitors
- CBB Capacitors(polyprop...
- Multilayer Ceramic Capac...

Title(PartNO)	Footprint	Capacitance	Inductar
NTCG164BH103JT1	R0603		
ERTJ0EV104GM	R0402		
ERTJ1VV154J	R0603		
ERTJ1VR223G	R0603		
ERTJ0EP333H	R0402		
ERTJ1VA220H	R0603		
ERTJ1VG103HA	R0603		
ERTJZER104H	R0201		
ERTJZEP473G	R0201		
ERTJ1VT202H	R0603		
ERTJ0EA680H	R0402		

EasyEDA > Symbol > LCSC > NTC Thermistors > NTCG164BH103JT1

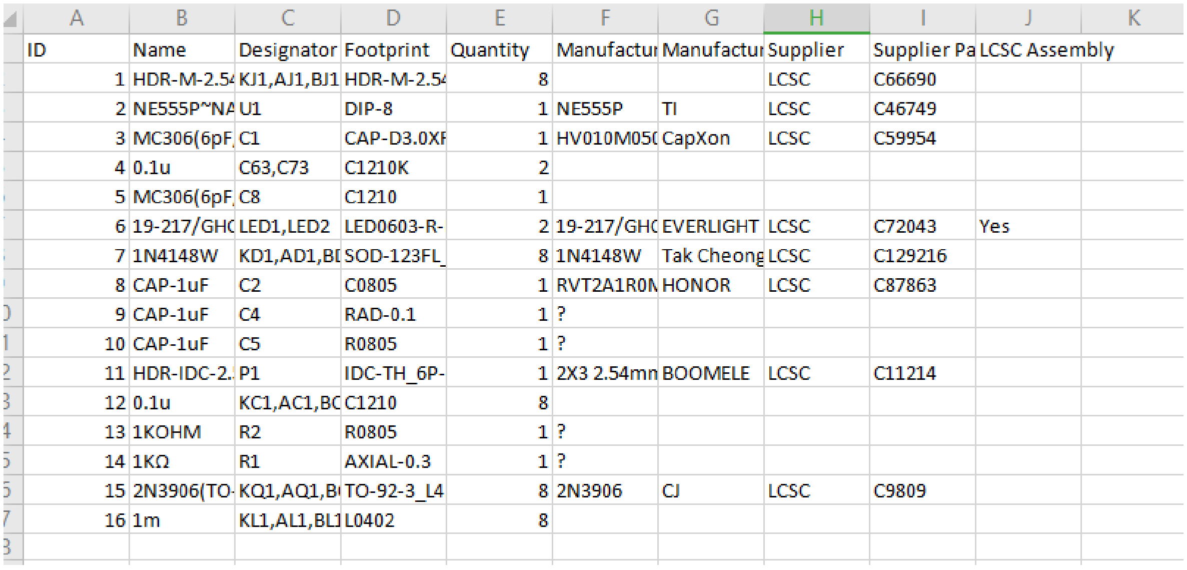
$0.0769 LCSC Part#: C524451 Stock: 3195 Minimum: 5 Distributor: LCSC

When you click the "Order Parts/Check Stock" button, we will help you to list all the components of your BOM at LCSC.com(If you haven't login LCSC, you have to login first). If you want to buy the components form LCSC, and you just need to put them to the cart and check out.


Export BOM

ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part		Price
1	47k	R1	0805-RESISTOR	1	?				Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?				Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?				Assign LCSC Part#	
4	10u	C1	0805	1	?				Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593	Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471	Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308

You can open the BOM in any text editor or spreadsheet.



ID	Name	Designator	Footprint	Quantity	Manufactu	Manufactu	Supplier	Supplier Pa	LCSC Assembly	
1	HDR-M-2.5	KJ1,AJ1,BJ1	HDR-M-2.5	8			LCSC	C66690		
2	NE555P~NA	U1	DIP-8	1	NE555P	TI	LCSC	C46749		
3	MC306(6pF,	C1	CAP-D3.0XI	1	HV010M05(	CapXon	LCSC	C59954		
4	0.1u	C63,C73	C1210K	2						
5	MC306(6pF,	C8	C1210	1						
6	19-217/GH(	LED1,LED2	LED0603-R-	2	19-217/GH(	EVERLIGHT	LCSC	C72043	Yes	
7	1N4148W	KD1,AD1,BI	SOD-123FL_	8	1N4148W	Tak Cheong	LCSC	C129216		
8	CAP-1uF	C2	C0805	1	RVT2A1R0N	HONOR	LCSC	C87863		
9	CAP-1uF	C4	RAD-0.1	1	?					
10	CAP-1uF	C5	R0805	1	?					
11	HDR-IDC-2.	P1	IDC-TH_6P-	1	2X3 2.54mn	BOOMELE	LCSC	C11214		
12	0.1u	KC1,AC1,BC	C1210	8						
13	1KOHM	R2	R0805	1	?					
14	1KΩ	R1	AXIAL-0.3	1	?					
15	2N3906(TO-	KQ1,AQ1,B	TO-92-3_L4	8	2N3906	CJ	LCSC	C9809		
16	1m	KL1,AL1,BL1	L0402	8						

Export BOM supports to export LCSC part price, it is the same as LCSC website.

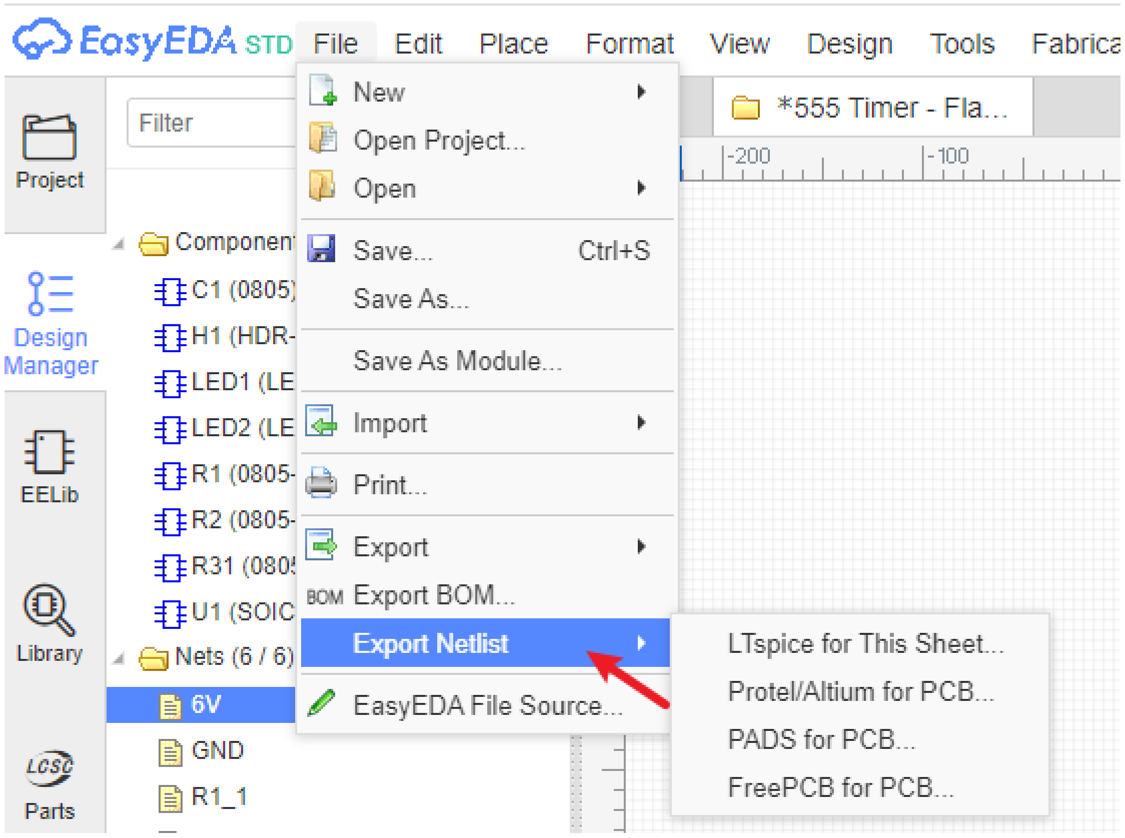
Notice:

- Before v6.4.17, If your project has schematic and PCB, the BOM data will come from schematic; if the project only has PCB, the BOM data will come from PCB.
- Since v6.4.17, the schematic BOM and PCB BOM are separated. If you assign the LCSC part at the PCB, it will not modify the schematic.
- In order to support multiple languages, BOM and coordinate files (CSV file) are UNICODE encoded and tab-based. If the CSV file cannot be read by your components vendor or PCB manufacturer, please convert the encoding and change the delimiter.
- Recommended solution: Save as a new CSV file in Excel or WPS. For example, open a CSV file in Excel, click or select: Save As - Other Formats - CSV (Comma Separated) (*. csv). You can also open the CSV file with any text editor (such as Windows Notepad) and save as ANSI or UTF-8 encoding. If necessary, replace all tabs with commas.

Export NetList

EasyEDA can export the netlist for the whole active project:

File > Export NetList > Spice...



EasyEDA can export a netlist in a variety of formats:

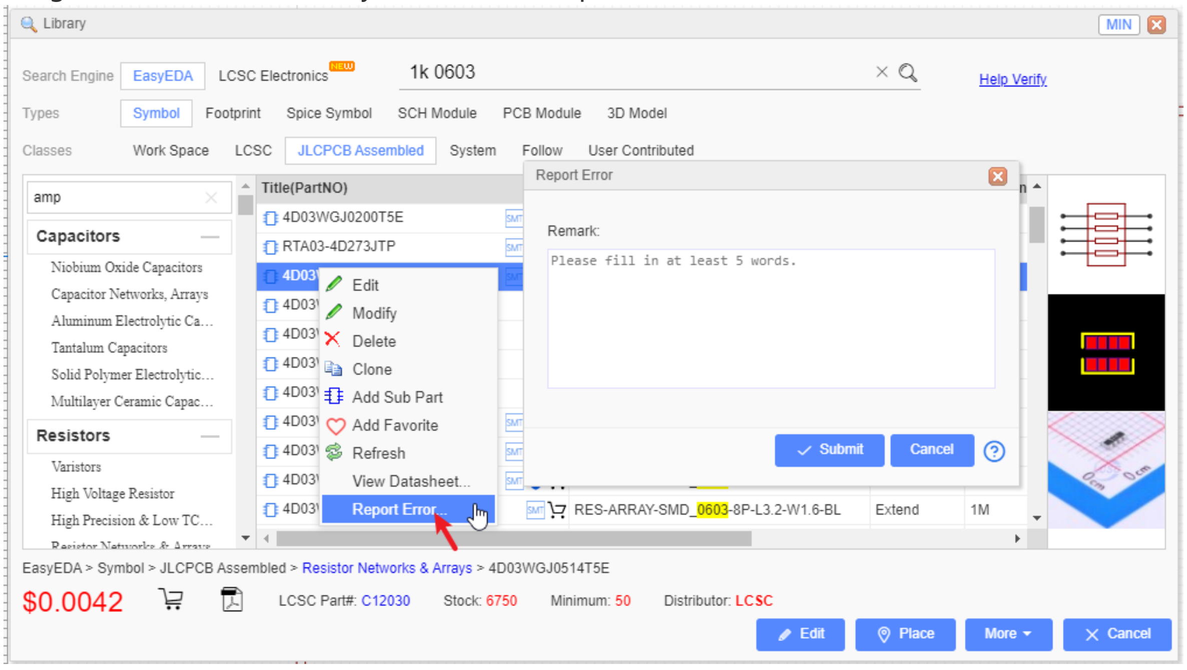
- **LTSpice for this Sheet**: this is a Spice compatible netlist generated by the simulation engine of EasyEDA, It is not normally used as the basis for as a PCB layout.
- **Protel/Altium for PCB**: a PCB netlist in a format that can be imported straight into Altium Designer and it's predecessor, Protel.
- **PADS for PCB**: a PCB netlist in a format that can be imported straight into Pads PCB layout tools.
- **FreePCB for PCB**: a PCB netlist in a format that can be imported straight into FreePCB, a free, open source PCB editor for Windows.

Report Error

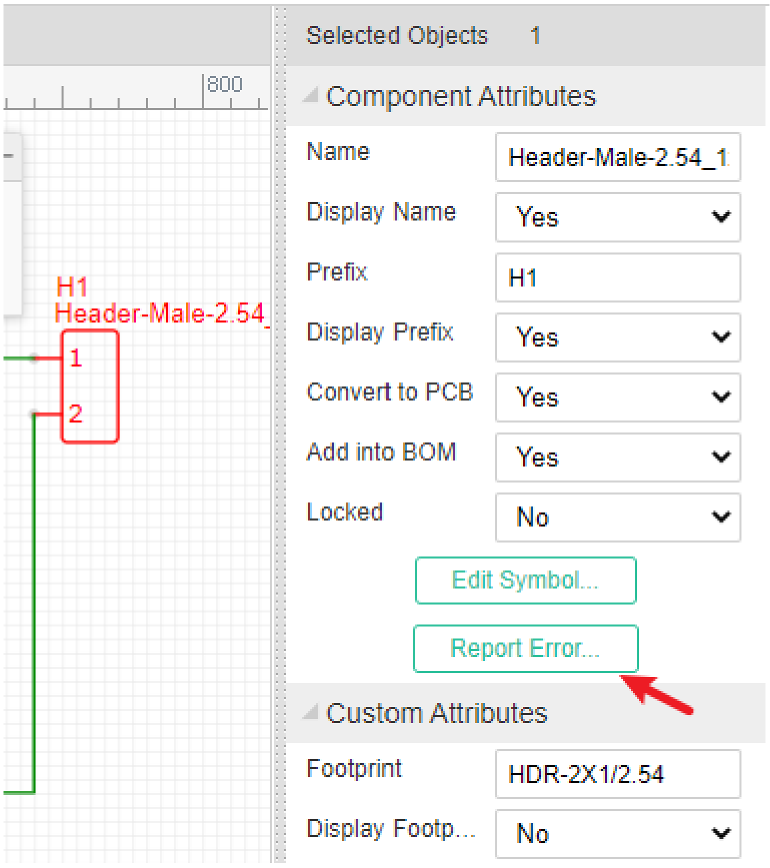
For EasyEDA official libraries, we have staffs to draw and maintain(LCSC & JLCPCB Assembled part) and we will try to keep them correctly as we can, but EasyEDA(System part) included a lot of open source of the libraries and the official drawing of the libraries, that can not avoid the wrong situation 100%, so when you meet a incorrect library, Please inform us in time, we will fix it as soon as possible.

There are 3 ways to report error:

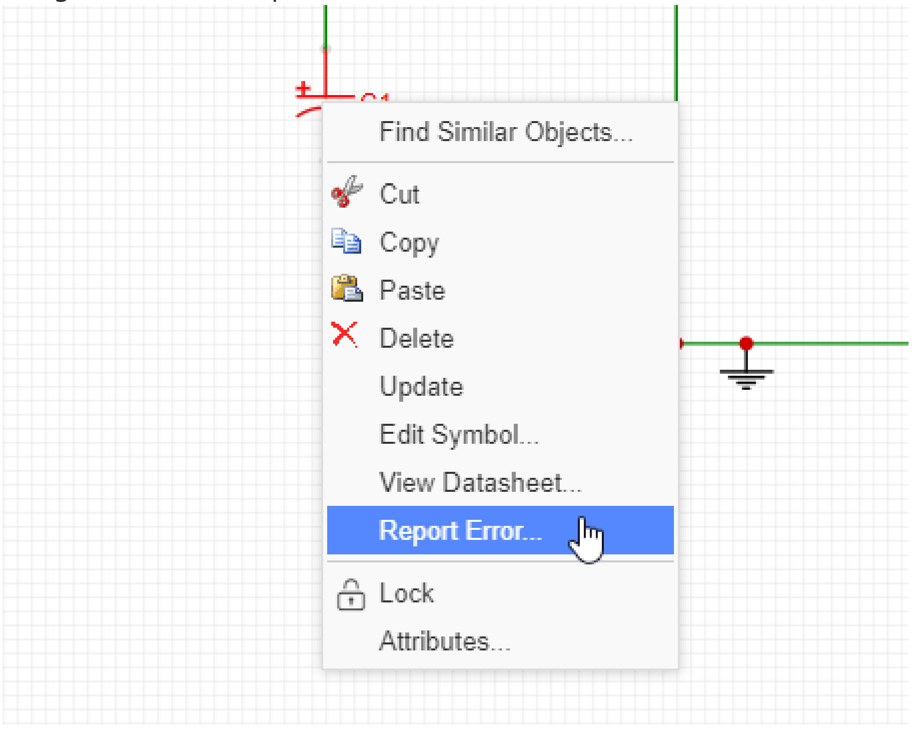
1.Right-click the offical library and use the "Report Error" function on the "Libraries".



2.Select the offical library on the canvas of the schematic/schematic module, click the "Report Error" button at the right-hand panel.



or right-click the component:



3.Send Email to us or post a topic at [Bug report]

support@easyeda.com

Create the Schematic Symbol

Create the Schematic Symbol

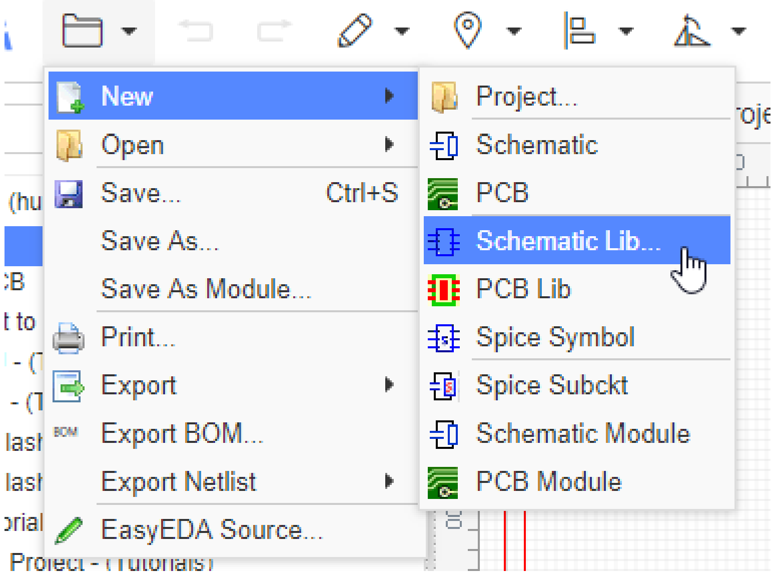
Using **Schematic Symbol Wizard** and **Group/Ungroup...** is a quick way to create schematic symbols but they are placed directly into the schematic that they are built in.

It is possible to reuse them by copying them (`CTRL+C` hotkeys) from the schematic they were created in and then cross-document-pasting them (`CTRL+SHIFT+V` hotkeys) into a different schematic but this quickly gets messy if you need to copy symbols that were created in several different schematics.

OK, you could keep copying new symbols into a dedicated "symbol library" schematic sheet to save searching for them but EasyEDA offers you an easier way to create and manage your symbols in a library.

Start a new Schematic Lib as shown below or by doing:

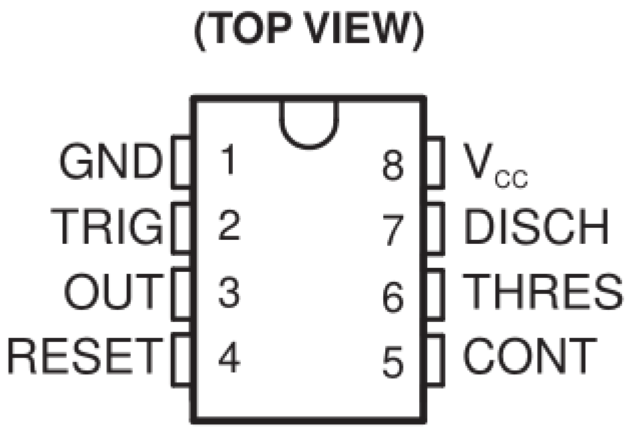
1. File > New > Symbol



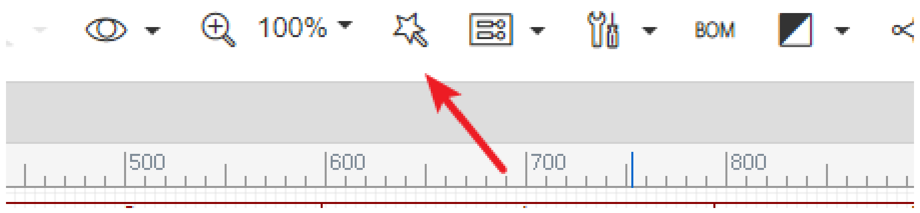
This opens the New SchematicLib symbol editor.

2. Create the symbol

- **Get the Datasheet**
  For example, using the NE555DR, the datasheet you can refer LCSC: NE555DR.
  And then create the symbol and place the pins for the library base on the datasheet.
  This component have 8 pins and names.



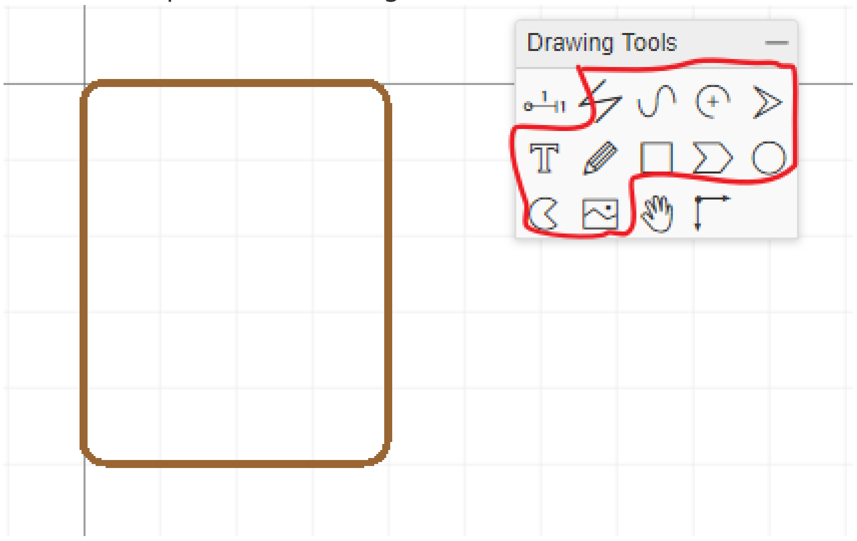
- **Create via Schematic Symbol Wizard**



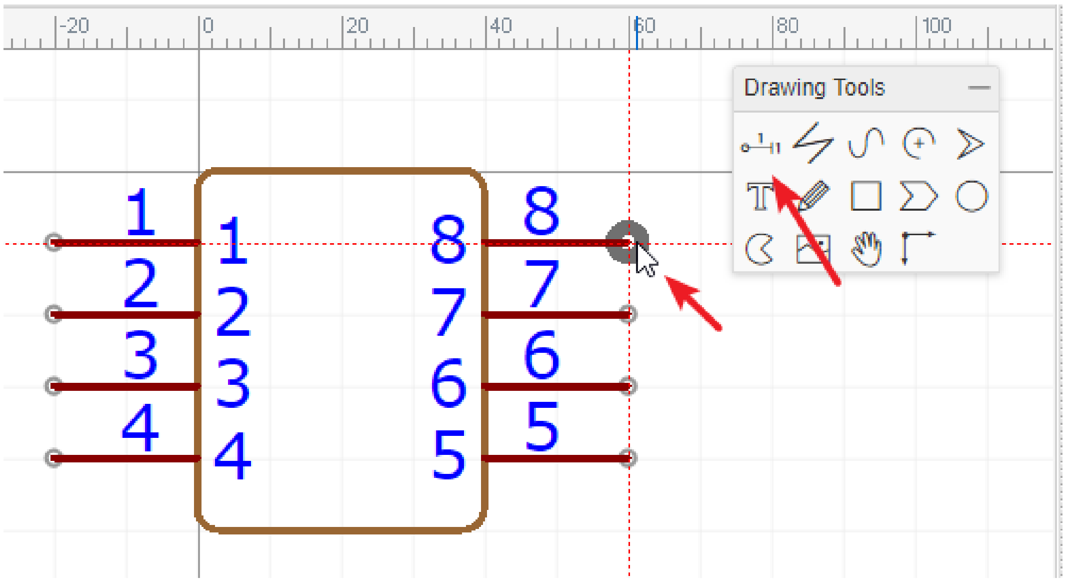
  The more information of **Schematic Symbol Wizard** please refer next section.

- **Create by Manually**

- Draw the shape via the Drawing Tools

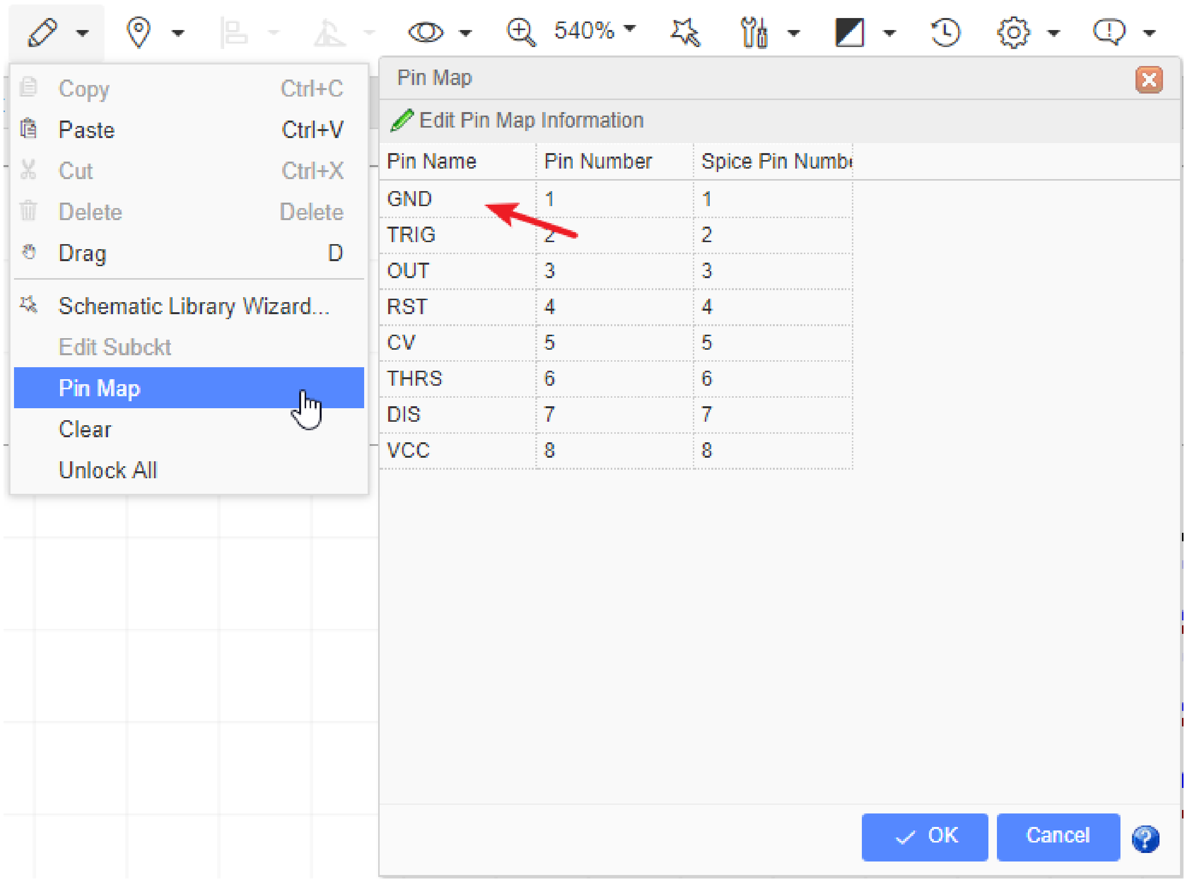


- Place the Pins



The Pin dot must keep out side as the image indicated, it is connecting with the wires. The more information please refer **SchematicLib Attributes - Pins** Section.

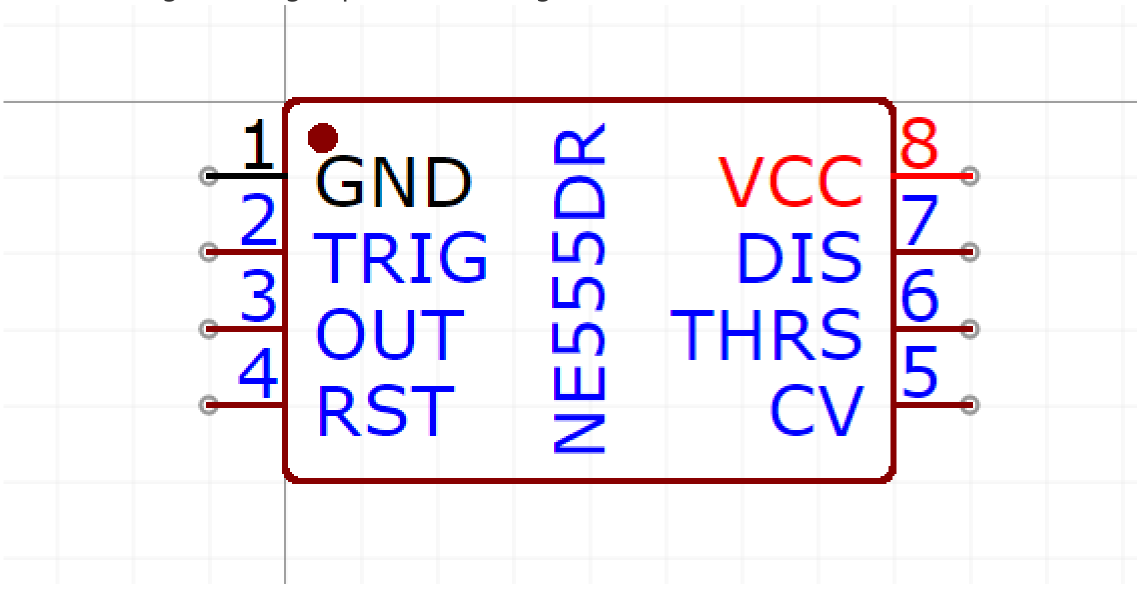
3. Edit the pin map
Via **Edit** > **Pin Map...**, change Pin names and Pin numbers. For some complicated IC, will use the alphabet for the pin number.



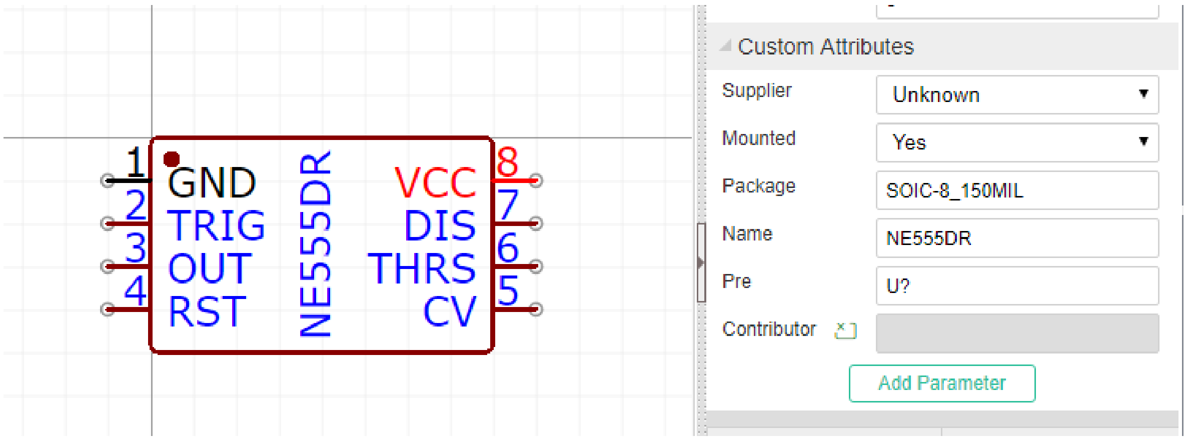
Pin Name	Pin Number	Spice Pin Numb
GND	1	1
TRIG	2	2
OUT	3	3
RST	4	4
CV	5	5
THRS	6	6
DIS	7	7
VCC	8	8

Note: if the pin is hidden, a network will be generated according to the pin name for connection. If it is not necessary, it is recommended not to hide.

4. Modify the Detail
such as change Pin length, place text, change Pin color, Pin attributes etc.



5. Set Costom Attributes
You can set the supplier, footprint(Suggested, you must assign the footprint via "Footprint Manager"), Name(Required), Prefix(Required) for it, the more detail of attributes please refer below section: **Custom Attributes**



Custom Attributes

Supplier	Unknown
Mounted	Yes
Package	SOIC-8_150MIL
Name	NE555DR
Pre	U?
Contributor	

If the schematiclib need to assign the packahe, the Pin number should match the footprint's Pad number. The detail of the footprint assign please refer the **Footprint Manager** section at previous.

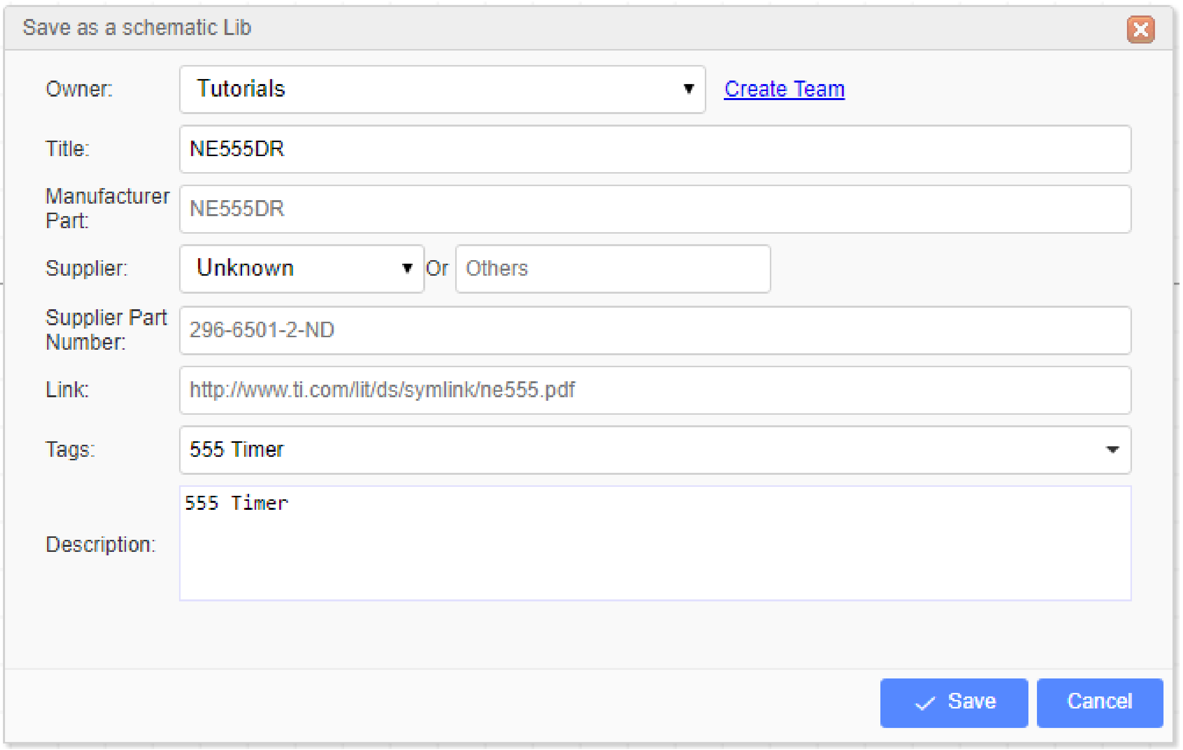
- *If the part's property "Convert to PCB" is set as "No", it will not appear at footprint manager.*

6. Set the Origin

You can via: "Top Menu - Place - Set Canvas Origin - By Center Grid of Symbols" to set the origin.

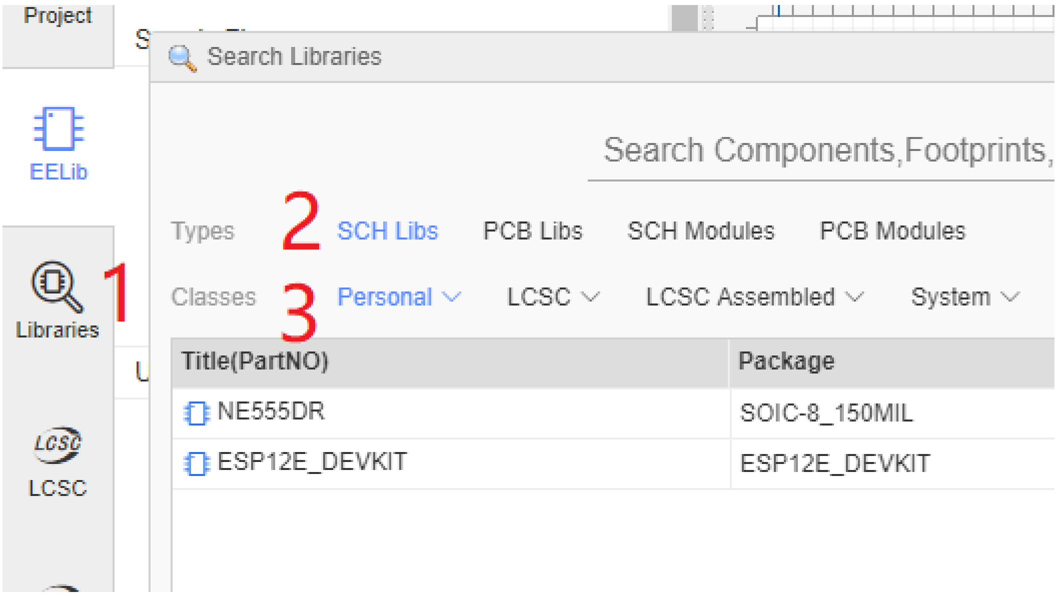
7. Save your SchameticLib

You can set this library's owner, datasheet link and tags etc.


Save as a schematic Lib

Owner:	Tutorials
Title:	NE555DR
Manufacturer Part:	NE555DR
Supplier:	Unknown Or Others
Supplier Part Number:	296-6501-2-ND
Link:	http://www.ti.com/lit/ds/symlink/ne555.pdf
Tags:	555 Timer
Description:	555 Timer

Then a Schematic Symbol is created finish. And the you can find it at "Libraries - SchematicLib - Personl" on the left-hand.



Title(PartNO)	Package
NE555DR	SOIC-8_150MIL
ESP12E_DEVKIT	ESP12E_DEVKIT

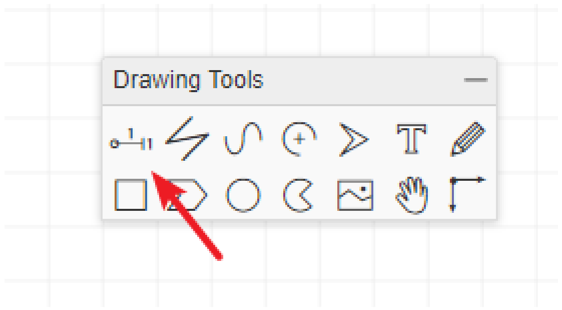
Notice:

- **Note the Origin Point.** To simplify rotating your symbols when they are placed into the canvas, make sure all of your symbols are created as near as possible centered around that point. Suggesting the first Pin/Pad or its center to be the origin point.
- *Please make sure all pins dot are placed on the grid, otherwise, when place the library on the schematic will causing the wiring difficult.*

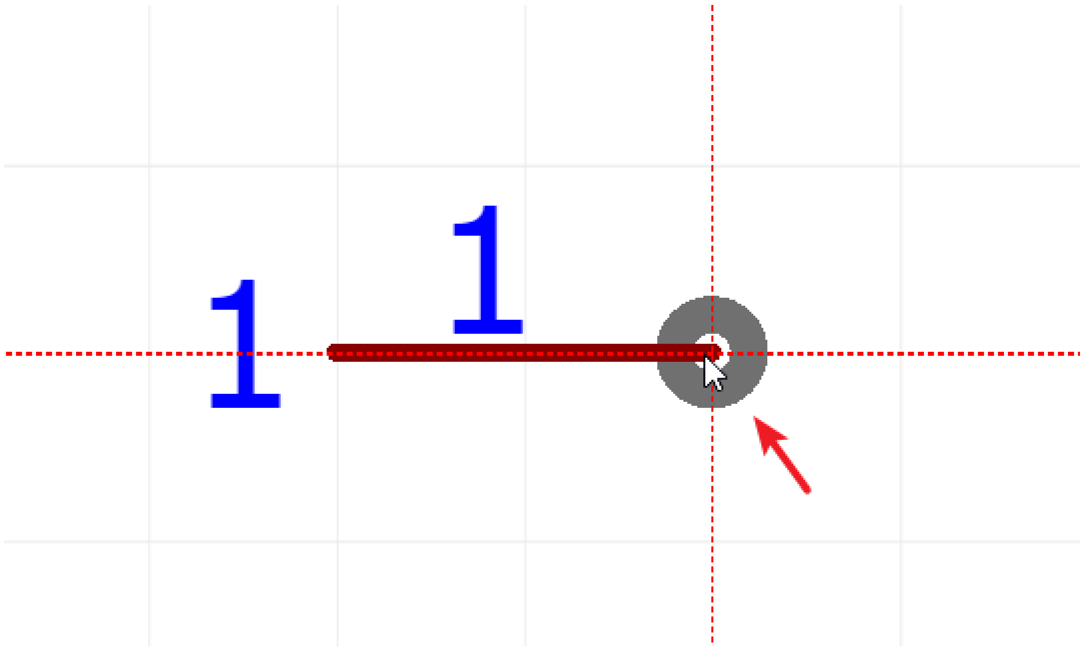
Pin Attributes

Symbols pins are the most important part of any Schematic Lib symbol. They are the things that allow wires to be attached to symbols to connect up your circuit.

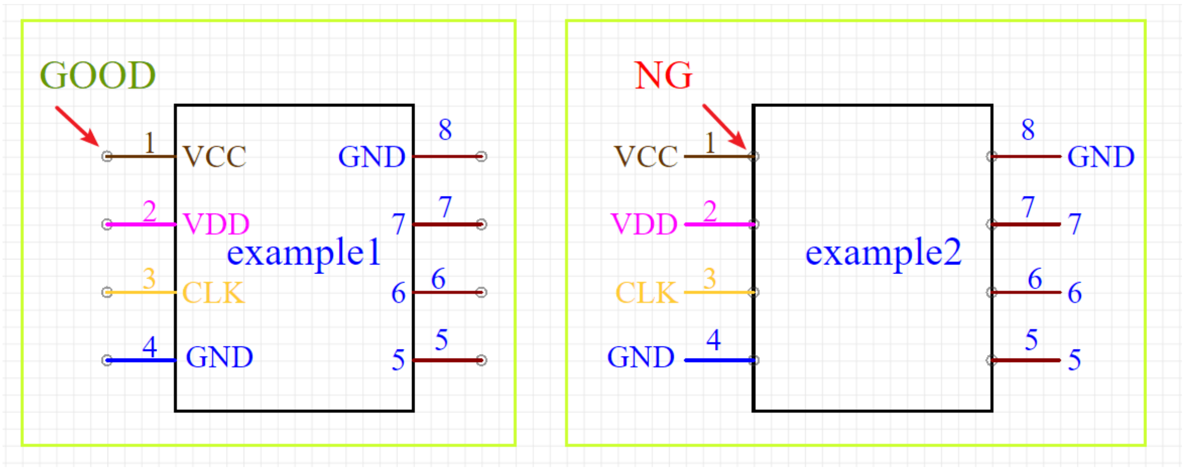
You can use the **P** hotkey to add a Pin or from the Draw Tools pallete:



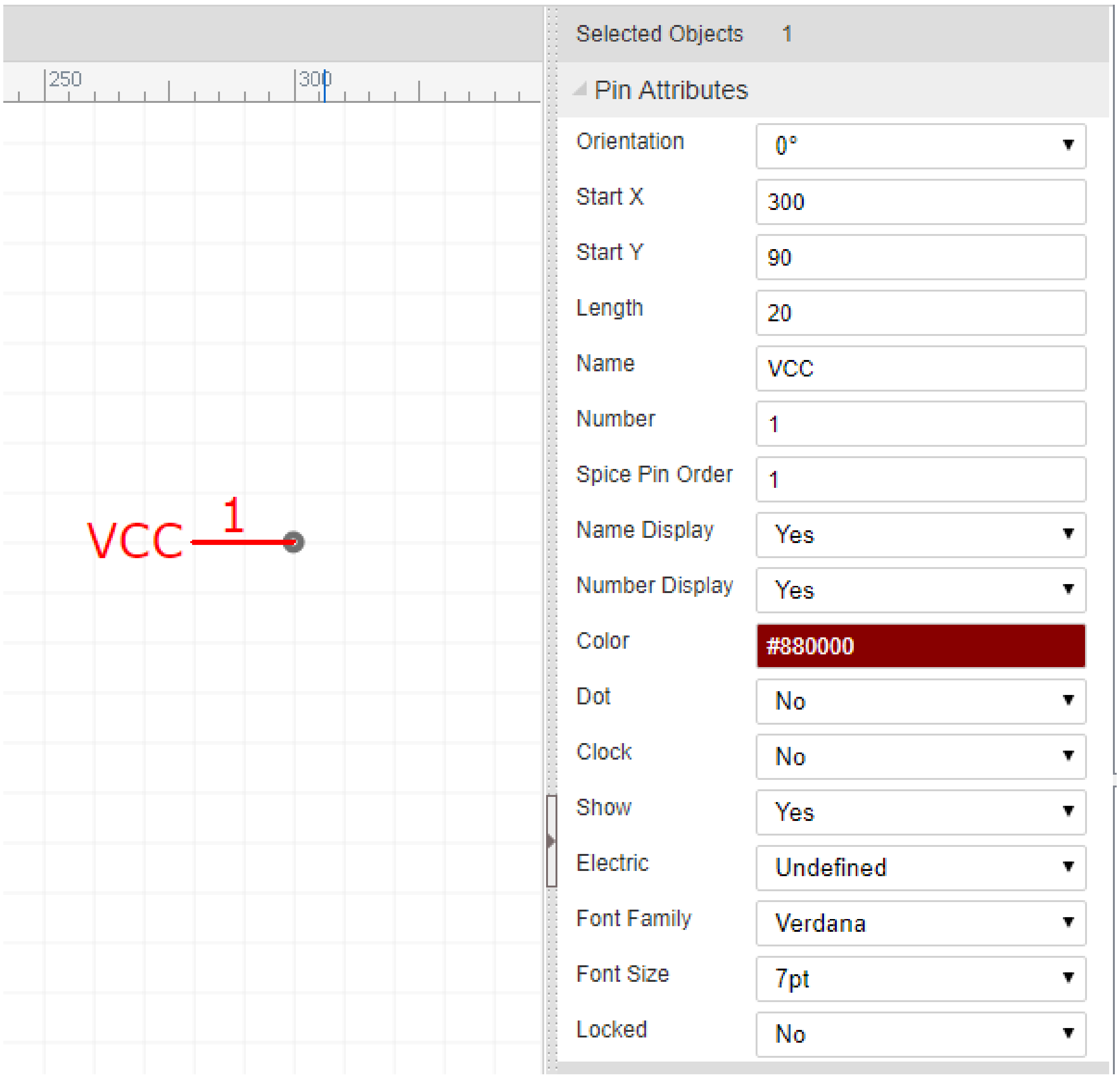
Before placing it on the canvas, you can use the rotation hotkey or rotate and flip from the menu to rotate it to the right orientation. Make sure the **Pin Dot(black dot)** is in the right position. The **Pin Dot** will be used to connect your wires or netlabels. Whenever a PIN is either placed as directly onto the canvas or as part of a symbol, the mouse has to point to the **Pin Dot** position to automatically start the Wire mode or to join a wire to it.



Whenever a Pin is placed as part of a symbol, the **Pin dot** should be **outside** of — and pointing away from — the symbol like in example 1(correct position), inside or pointing towards the symbol as shown in example 2(wrong position).



When you select a single Pin, the **Pin attributes** will be shown in the right hand **Properties** panel:



Selected Objects 1

Pin Attributes

Orientation	0°
Start X	300
Start Y	90
Length	20
Name	VCC
Number	1
Spice Pin Order	1
Name Display	Yes
Number Display	Yes
Color	#880000
Dot	No
Clock	No
Show	Yes
Electric	Undefined
Font Family	Verdana
Font Size	7pt
Locked	No

Orientation: 0°,90°, 180° and 270°。 If you want to create a 45° pin, you need to set it length as 0, and draw a line with 45°。

Start-X and Start-Y: The pindot position. Sometimes it may be difficult to move the pin to the desired position using the mouse, so you can move the pin via Start-X and Start-Y.

Length: Pin length.

Name: In this example, *VCC* is the name of the Pin.

Number: In this example, *1* is the number of the Pin. This number is the pin number of the device in a physical footprint.

Note that you can use alphanumeric identifiers such as; A1, B1, C1, A2, B2 and so on as the Number.

Spice Number: These are the pin numbers used to connect your symbol to the corresponding pins defined by the .model or .subckt used to simulate your device. The pin numbers of the simulation model may be different from the physical footprint pin numbers and - unless the model is specifically created to model multiple devices in a single footprint - do not change for different instances of a device in a multi-device footprint. The Spice Pin order must be **numerals** only.

Name Display: If you don't want to show *VCC*, switch it to NO.

Number Display: If you don't want to show *1*, switch it to NO.

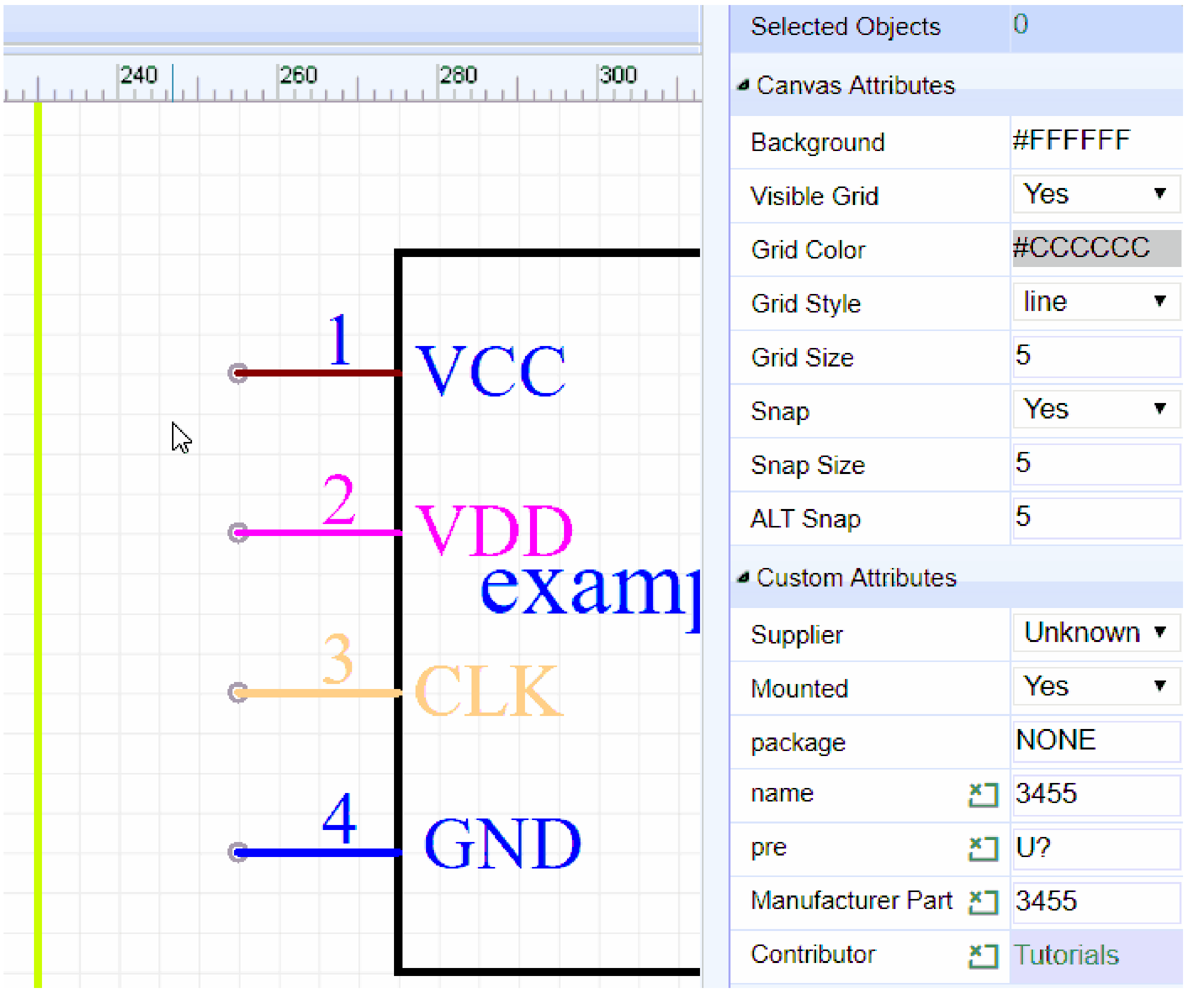
You can adjust the Name or Number position using your mouse but note that rotate and flip applies to the whole pin including the name and pin number; these items cannot be rotated and flipped independently of the pin itself.

Note also that rotate and flip actions do not result in upside down or mirrored pin number or names.

Color: You can set the Pin to different colours, such as *PIN3:CLK* as orange and *PIN4:GND* as blue. In this example, the PIN1 is set as color `#880000`, but it shows as red, because it is selected. After deselecting it, the pin will appear color `#880000`.

Dot: adds a circle to the inside end of the pin to indicate logical (or analogue) inversion.

Clk: adds a `>` to the inside end of the pin to indicate that the pin is logical clock input.



Selected Objects	0
Canvas Attributes	
Background	#FFFFFF
Visible Grid	Yes
Grid Color	#CCCCCC
Grid Style	line
Grid Size	5
Snap	Yes
Snap Size	5
ALT Snap	5
Custom Attributes	
Supplier	Unknown
Mounted	Yes
package	NONE
name	3455
pre	U?
Manufacturer Part	3455
Contributor	Tutorials

Show: YES/NO. Allows you to hide the pin. When set it to NO, this Pin will be hidden when the symbol is placed on the schematic editor canvas, and then create a net which name same as this pin name.

Note that the pin is not hidden here in the Schematic Lib symbol editor canvas because if it was, it would disappear from view and so how would you find it to make it visible again? For the same reason this option has no effect in symbols made using Group/Ungroup...

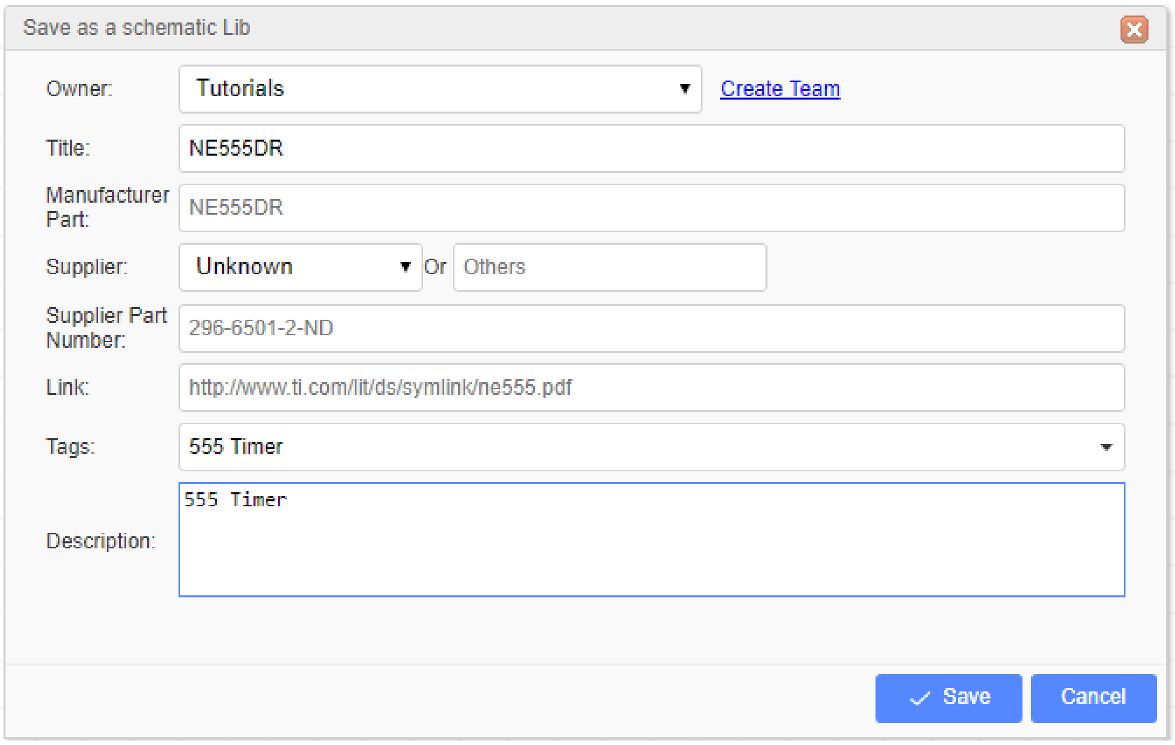
We may not have thought of everything in EasyEDA but we do try. :)

Electric: [Undefined, Input, Output, I/O, Power]

EasyEDA provides Electrical Rules Checking (ERC) right now, But you still need to set electric of your Schematic libs.

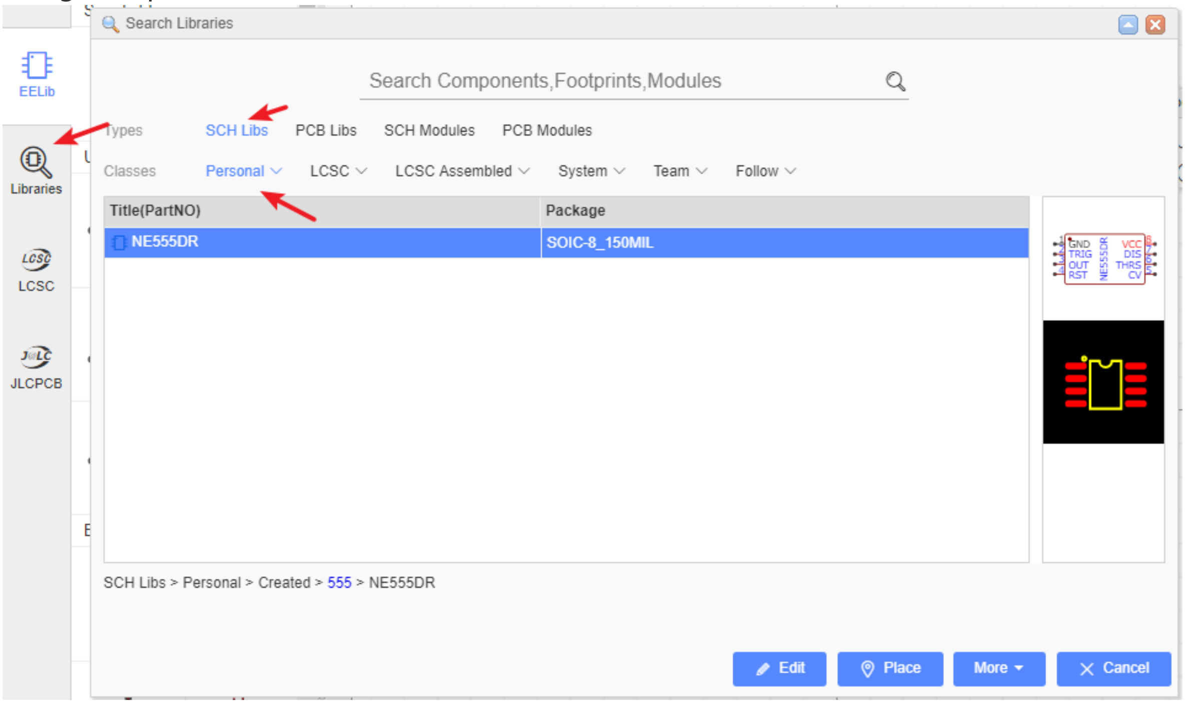
If you set the PIN as Power and set the pin to be hidden, then the Pin will be connected by Name which is the NetLabel. If the Name is VCC, it will be connected to the net in your circuit with the NetLabel or NetFlag VCC. This is helps to keep the schematic clear and uncluttered when using Multi-part Components.

After created the Lib, use `CTRL+S` will open the save dialog:


Save as a schematic Lib

Owner:	Tutorials
Title:	NE555DR
Manufacturer Part:	NE555DR
Supplier:	Unknown Or Others
Supplier Part Number:	296-6501-2-ND
Link:	http://www.ti.com/lit/ds/symlink/ne555.pdf
Tags:	555 Timer
Description:	555 Timer

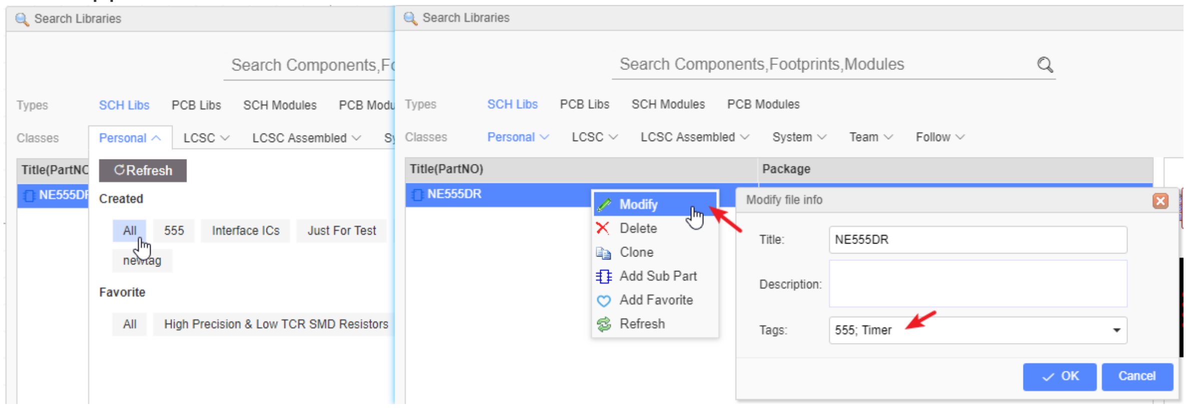
After clicking **Save**, you will see it appears in **Libraries** > **Symbols** > **Personal** of the left hand Navigation panel.



Title(PartNO)	Package
NE555DR	SOIC-8_150MIL

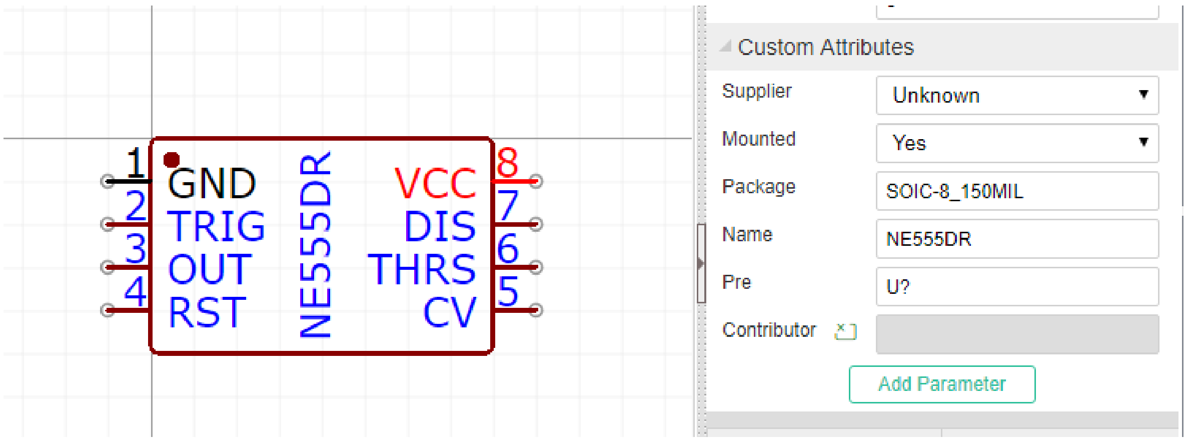
SCH Libs > Personal > Created > 555 > NE555DR

If you want to modify the tag for your new symbol: **Libraries** > **Symbols** > **Personal** > **Select New Lib** > **More** > **Modify**, or **right-click new Lib** > **Modify**, if your Lib doesn't have the tags it will appears on **All**.


Search Libraries
Search Components,Footprints,Modules
Types SCH Libs PCB Libs SCH Modules PCB Modules
Classes Personal LCSC LCSC Assembled System Team Follow
Title(PartNO) Package
NE555DR
Refresh
Created
All 555 Interface ICs Just For Test newtag
Favorite
All High Precision & Low TCR SMD Resistors
Modify
Delete
Clone
Add Sub Part
Add Favorite
Refresh
Modify file info
Title: NE555DR
Description:
Tags: 555; Timer
OK Cancel

Custom Attributes

In the Schematic Lib editor's canvas Properties panel, you will find a **Custom Attributes** section:



- **footprint**

How to change Schematic Symbol's footprint? If you would like to built a PCB, you need to assign a footprint for your Schematic symbol. Although there are other ways to do this in EasyEDA, here is the right place to do it. When you set a footprint , **the footprint's pad numbers must match the schematic Lib's pin number**, otherwise, when you convert the schematic to PCB , there will miss several nets.

Click in the **footprint** input box, and the **Footprint Manager** dialog will open as used to do this task in the Schematic Editor.

The more information please refer to **Schematic - Footprint Manager** section.

Notie:
You have to assign the footprint via the Footprint Manager, otherwise, the Schematic lib will not get the footprint correctly. The footprint is linked with SchematicLib by global unique ID not the title.

- **Prefix**

The default Schematic symbol Prefix is **U?** If you create a resistor, you can set the Prefix to **R?**. It is filled required.

- **Name**

You can change the schematic lib's name here, it is can be different from the part's file name.

- **Contributor**

This is your registered user name. When Other EasyEDA's users use your libraries, they will remember your contributions!

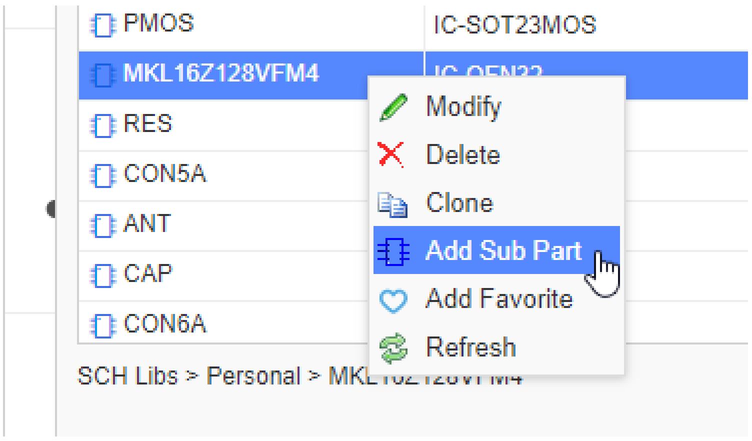
Symbol Subparts

We have already touched on how EasyEDA can support **Multi-part/Subpart Components** , but how do you create **multi-part components**?

EasyEDA provides a sub parts facility to do this.

After creating a part, you can right-click the part in the **Library** > **Symbols** > **Work Space** > **Created** section to pop up the content menu.

Suppose you have created your own symbol for a 74HCT04 hex inverter.



Right Click **Add sub part** and that will add 74HCT04.*1*,

Click again to add 74HCT04.*2* , up to 74HCT04.*6*.

Then double click on each sub part in turn to modify the Pin Name and Number attributes.

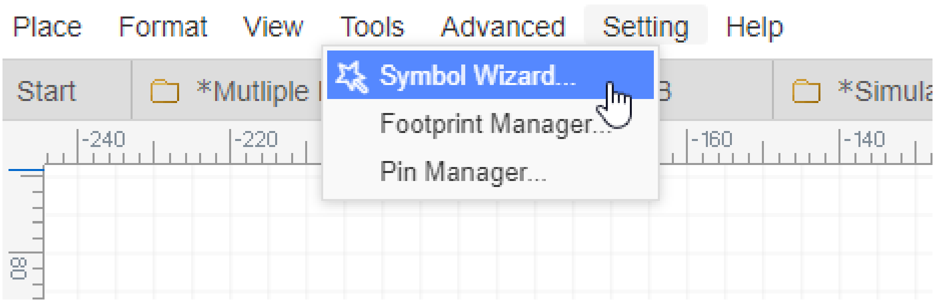
Easy or what?

Schematic Symbol Wizard

How many times have you hit a schematic capture roadblock because you couldn't find a component symbol?

Well, in EasyEDA that would be never because the **Schematic Symbol Wizard** provides a quick and easy way to create a general Schematic Symbol symbol.

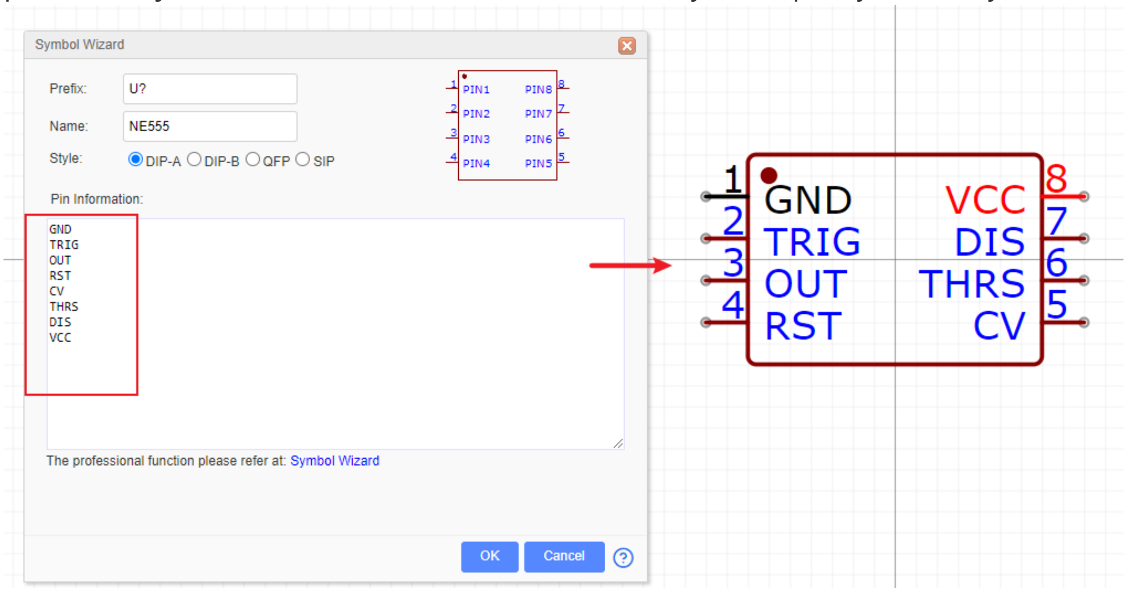
Via: Top Menu - Tools - Symbol Wizard



Basic Function

Input the Pins' name Only

1. Using the **NE555** timer as an example: this device is available in a **DIP8** package so select **DIP**. Then enter the NE555 pin names into the **Pin Names** text box separated by new line or space, Then press OK. Abracadabra! As if by magic, you will find a perfectly formed dual in line 8 pin symbol for the NE555 attached to your mouse cursor, ready to be placed! You just need a few seconds to build a NE555 symbol, quickly and easily.


Symbol Wizard

Prefix: U?

Name: NE555

Style: [✓ DIP-A] DIP-B QFP SIP

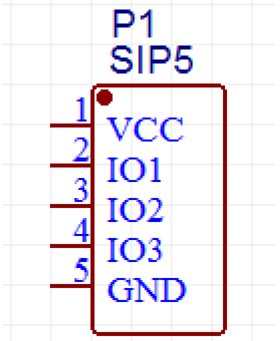
Pin Information:

```
GND
TRIG
OUT
RST
CV
THRS
DIS
VCC
```

The professional function please refer at: Symbol Wizard

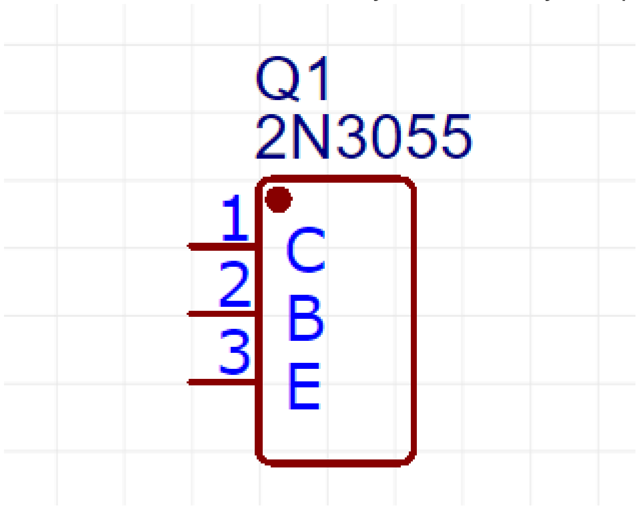
OK Cancel

2. The EasyEDA Schematic Symbol Wizard allows you to create DIP, QPF or SIP styles symbols. If you are designing Arduino Shields then you will need lots of SIP symbol, so you can create a SIP symbol like the one shown below in a few seconds.



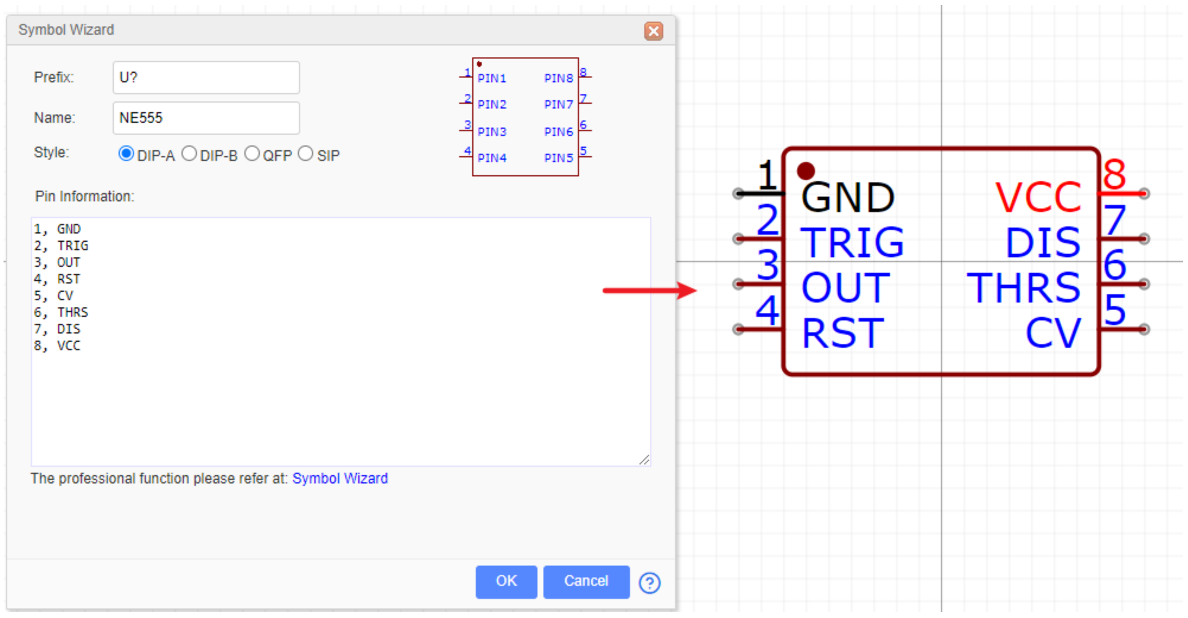
3. If you are not too worried that the symbols may not look quite the way people might expect and that they may not look anything like the Type you select, then of course you

can use the wizard to create symbols for any component:



Input the Pins' number and name

Schematic Symbol wizard support you input the pins' number and name.
As below example, setting every pin's number is easily.


Symbol Wizard

Prefix: U?

Name: NE555

Style: [✓ DIP-A] DIP-B QFP SIP

Pin Information:

```
1, GND
2, TRIG
3, OUT
4, RST
5, CV
6, THRS
7, DIS
8, VCC
```

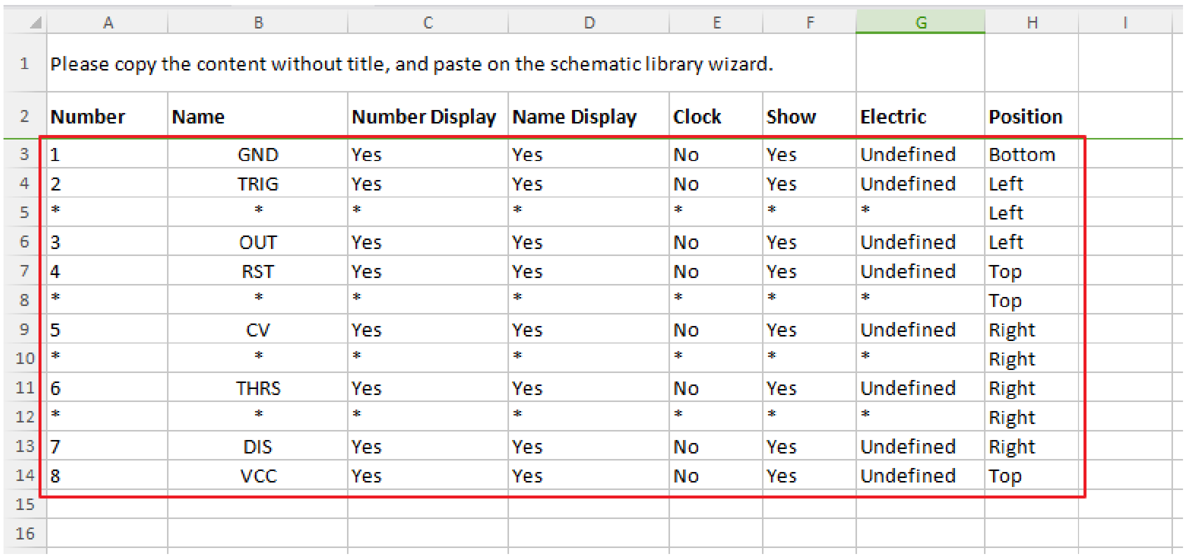
The professional function please refer at: Symbol Wizard

Professional Function

Schematic Symbol Wizard support the professional function, it is easier to create the large and complex and more convenient Schematic Symbol.

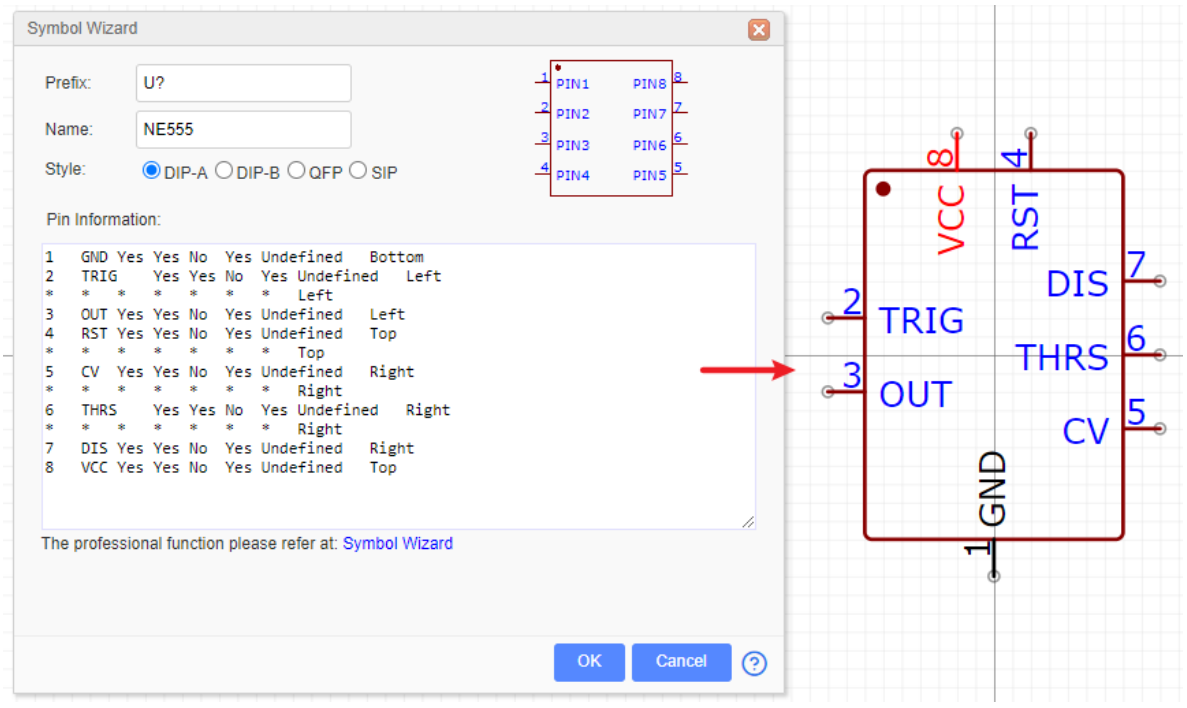
1.Download [Schematic Symbol Wizard Template.xlsx]

2.Open it via Excel or WPS, and edit each Pins attributes and position, and then copy the content and paste in wizard dialog without content title.
Tip: If you want to create the gap between Pin and Pin, you can use the `*` as below image.



	A	B	C	D	E	F	G	H
1	Please copy the content without title, and paste on the schematic library wizard.							
2	Number	Name	Number Display	Name Display	Clock	Show	Electric	Position
3	1	GND	Yes	Yes	No	Yes	Undefined	Bottom
4	2	TRIG	Yes	Yes	No	Yes	Undefined	Left
5	*	*	*	*	*	*	*	Left
6	3	OUT	Yes	Yes	No	Yes	Undefined	Left
7	4	RST	Yes	Yes	No	Yes	Undefined	Top
8	*	*	*	*	*	*	*	Top
9	5	CV	Yes	Yes	No	Yes	Undefined	Right
10	*	*	*	*	*	*	*	Right
11	6	THRS	Yes	Yes	No	Yes	Undefined	Right
12	*	*	*	*	*	*	*	Right
13	7	DIS	Yes	Yes	No	Yes	Undefined	Right
14	8	VCC	Yes	Yes	No	Yes	Undefined	Top
15								
16								

3.The Wizard will create the symbol follow your content. The types you chosen will be ignored.



Prefix: U?

Name: NE555

Style: [✓ DIP-A] DIP-B QFP SIP

Pin Information:

```
1   GND Yes Yes No  Yes Undefined   Bottom
2   TRIG    Yes Yes No  Yes Undefined   Left
*   *   *   *   *   *   *   Left
3   OUT Yes Yes No  Yes Undefined   Left
4   RST Yes Yes No  Yes Undefined   Top
*   *   *   *   *   *   *   Top
5   CV  Yes Yes No  Yes Undefined   Right
*   *   *   *   *   *   *   Right
6   THRS    Yes Yes No  Yes Undefined   Right
*   *   *   *   *   *   *   Right
7   DIS Yes Yes No  Yes Undefined   Right
8   VCC Yes Yes No  Yes Undefined   Top
```

The professional function please refer at: Symbol Wizard

Notice:

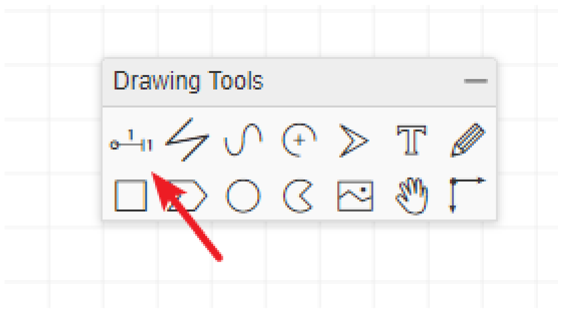
- *If the content you input wasn't one, two or eight columns, it will shown incorrect format.*
- *You can use the Key Space to separate the column data.*

Schematic Symbol Attributes

Pin Attributes

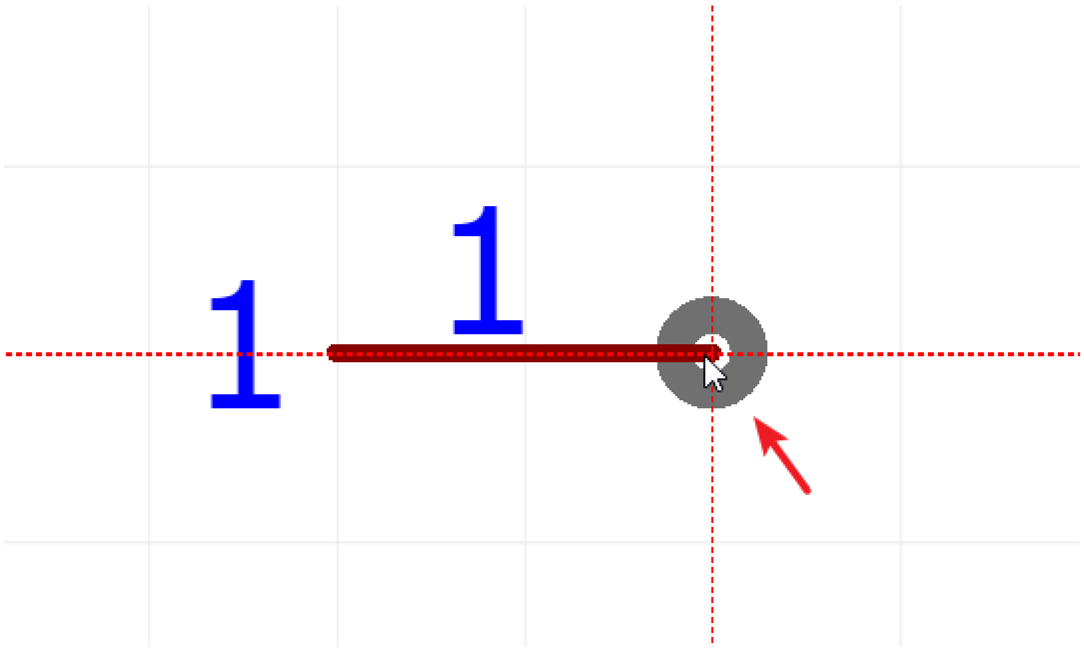
Symbols pins are the most important part of any Schematic Lib symbol. They are the things that allow wires to be attached to symbols to connect up your circuit.

You can use the **P** hotkey to add a Pin or from the Draw Tools pallete:

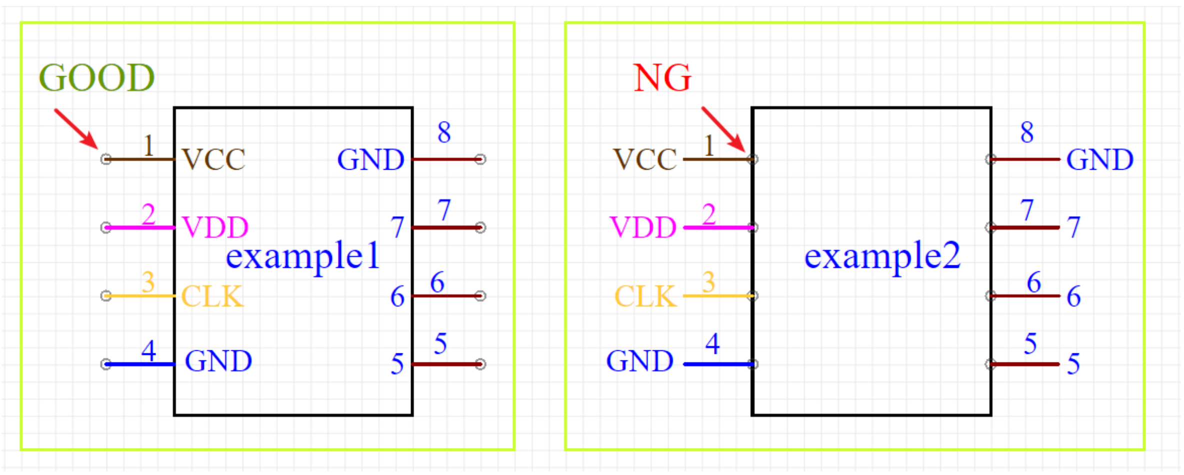


Pin Orientation

Before placing it on the canvas, you can use the rotation hotkey or rotate and flip from the menu to rotate it to the right orientation. Make sure the **Pin Dot(black dot)** is in the right position. The **Pin Dot** will be used to connect your wires or netlabels. Whenever a PIN is either placed as directly onto the canvas or as part of a symbol, the mouse has to point to the **Pin Dot** position to automatically start the Wire mode or to join a wire to it.

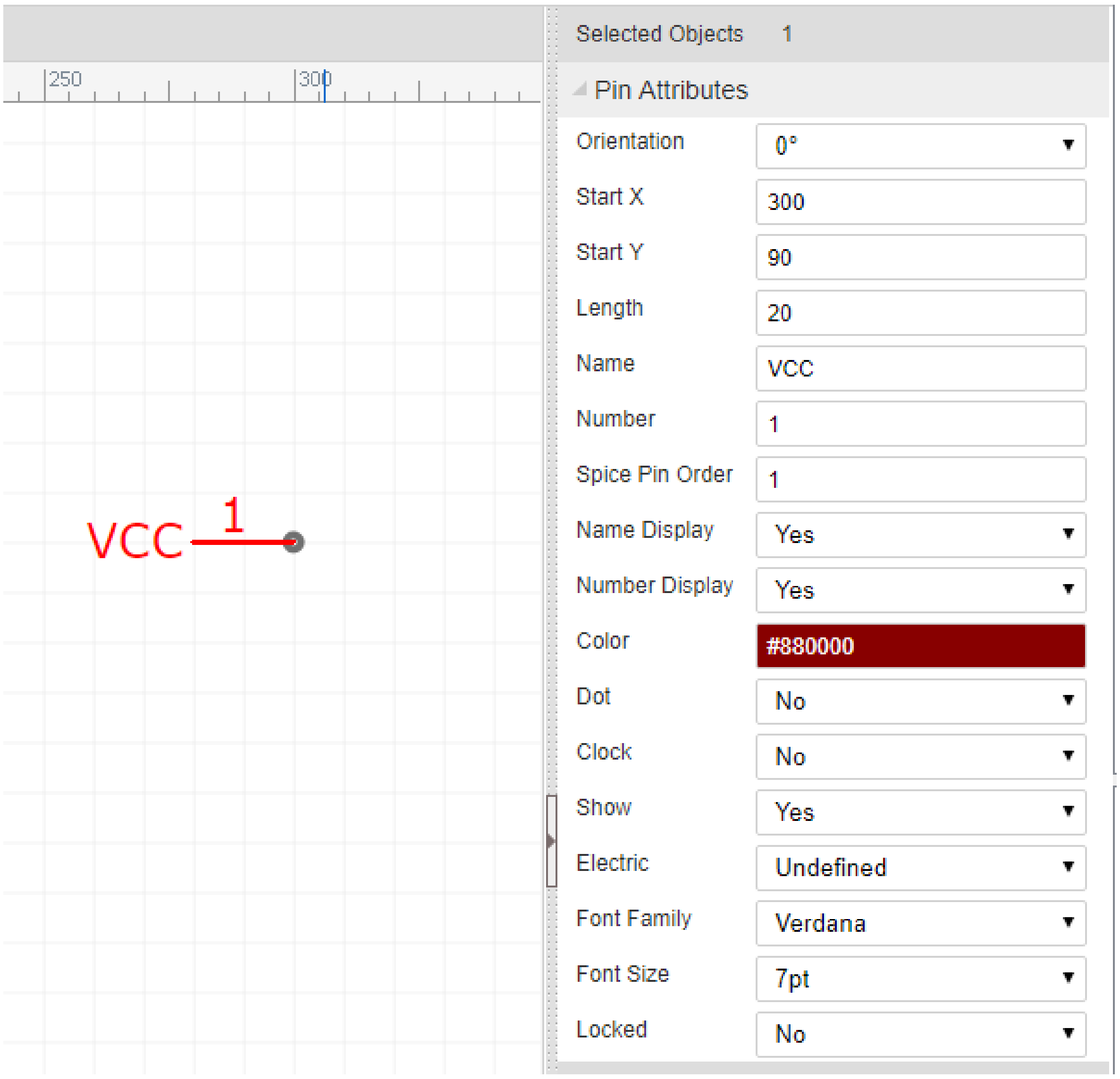


Whenever a Pin is placed as part of a symbol, the **Pin dot** should be **outside** of — and pointing away from — the symbol like in example 1(correct position), inside or pointing towards the symbol as shown in example 2(wrong position).



Pin Attributes

When you select a single Pin, the **Pin attributes** will be shown in the right hand **Properties** panel:



Selected Objects	1
Pin Attributes	
Orientation	0°
Start X	300
Start Y	90
Length	20
Name	VCC
Number	1
Spice Pin Order	1
Name Display	Yes
Number Display	Yes
Color	#880000
Dot	No
Clock	No
Show	Yes
Electric	Undefined
Font Family	Verdana
Font Size	7pt
Locked	No

Orientation: 0°,90°, 180° and 270°。 If you want to create a 45° pin, you need to set it length as 0, and draw a line with 45°。

Start-X and Start-Y: The pindot position. Sometimes it may be difficult to move the pin to the desired position using the mouse, so you can move the pin via Start-X and Start-Y.

Length: Pin length.

Name: In this example, *VCC* is the name of the Pin.

Number: In this example, *1* is the number of the Pin. This number is the pin number of the device in a physical footprint

Note that you can use alphanumeric identifiers such as; A1, B1, C1, A2, B2 and so on as the Number.

Spice Number: These are the pin numbers used to connect your symbol to the corresponding pins defined by the .model or .subckt used to simulate your device. The pin numbers of the simulation model may be different from the physical footprint pin numbers and - unless the model is specifically created to model multiple devices in a single footprint - do not change for different instances of a device in a multi-device footprint. The Spice Pin order must be **numerals** only.

Display Name: If you don't want to show *VCC*, switch it to NO.

Display Number: If you don't want to show *1*, switch it to NO.

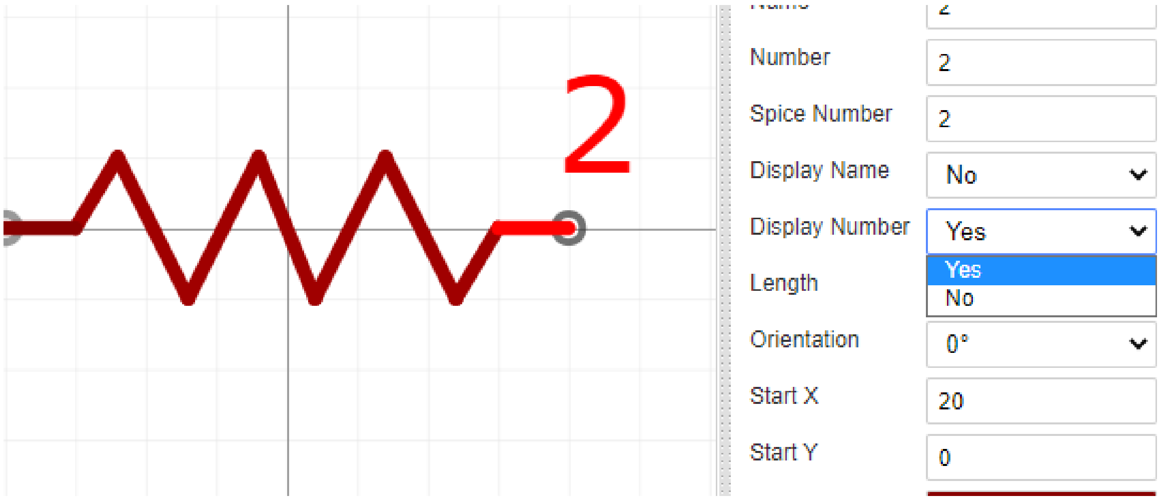
You can adjust the Name or Number position using your mouse but note that rotate and flip applies to the whole pin including the name and pin number; these items cannot be rotated and flipped independently of the pin itself.

Note also that rotate and flip actions do not result in upside down or mirrored pin number or names.

Color: You can set the Pin to different colours, such as *PIN3:CLK* as orange and *PIN4:GND* as blue. In this example, the PIN1 is set as color `#880000`, but it shows as red, because it is selected. After deselecting it, the pin will appear color `#880000`.

Dot: adds a circle to the inside end of the pin to indicate logical (or analogue) inversion.

Clock: adds a `>` to the inside end of the pin to indicate that the pin is logical clock input.



Number	2
Spice Number	2
Display Name	No
Display Number	Yes
Length	
Orientation	0°
Start X	20
Start Y	0

Show: YES/NO. Allows you to hide the pin. When set it to NO, this Pin will be hidden when the symbol is placed on the schematic editor canvas.

Note that the pin is not hidden here in the Schematic Lib symbol editor canvas because if it was, it would disappear from view and so how would you find it to make it visible again? For the same reason this option has no effect in symbols made using Group/Ungroup...

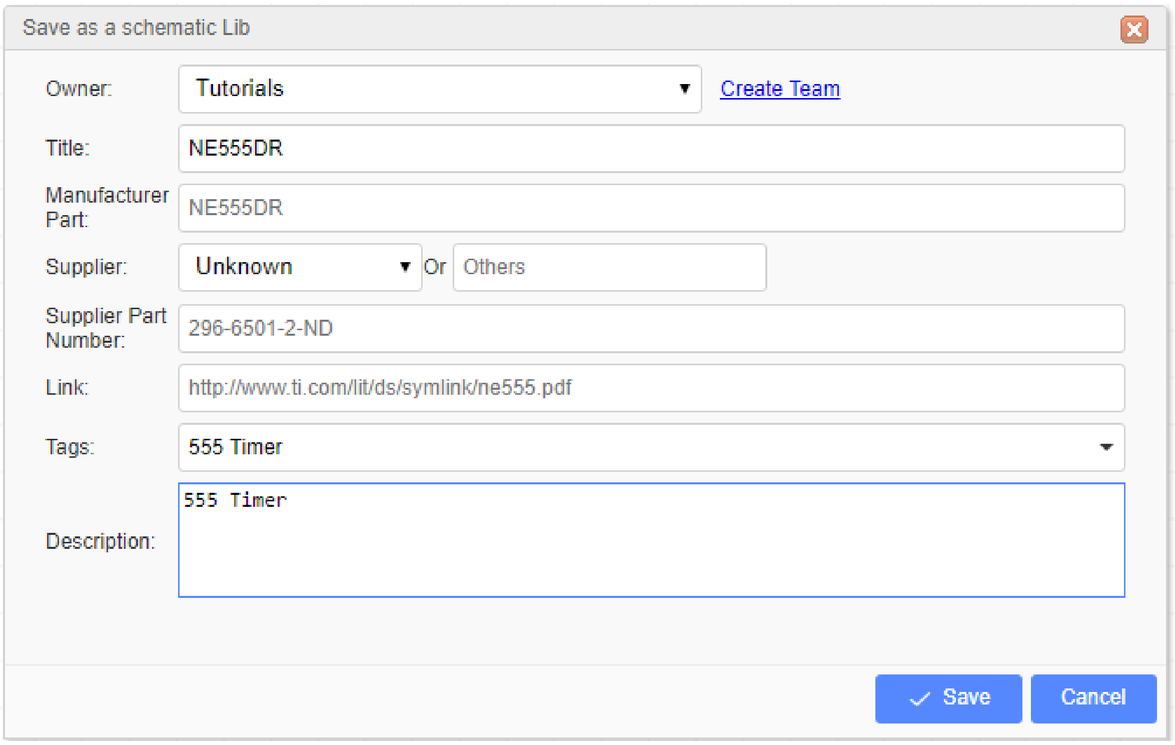
We may not have thought of everything in EasyEDA but we do try. :)

Electric: [Undefined, Input, Output, I/O, Power]

EasyEDA provides Electrical Rules Checking (ERC) right now, But you still need to set electric of your Schematic libs.

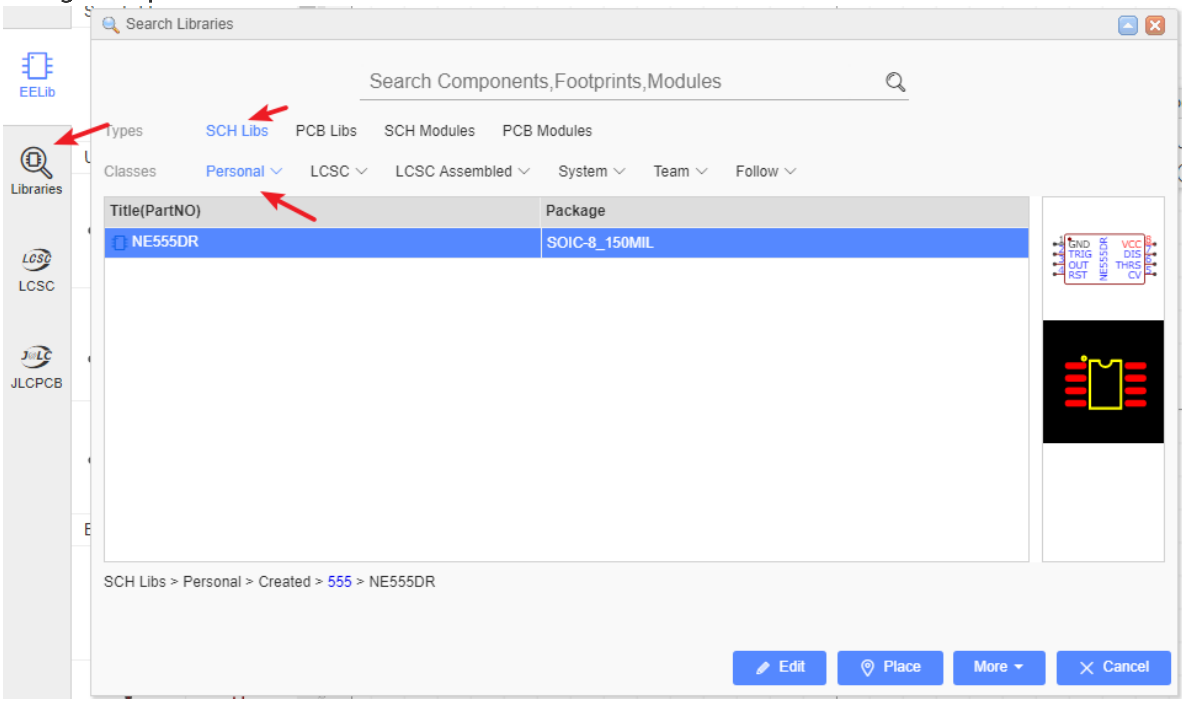
If you set the PIN as Power and set the pin to be hidden, then the Pin will be connected by Name which is the NetLabel. If the Name is VCC, it will be connected to the net in your circuit with the NetLabel or NetFlag VCC. This is helps to keep the schematic clear and uncluttered when using Multi-part Components.

After created the Lib, use `CTRL+S` will open the save dialog:


Save as a schematic Lib

Owner:	Tutorials
Title:	NE555DR
Manufacturer Part:	NE555DR
Supplier:	Unknown Or Others
Supplier Part Number:	296-6501-2-ND
Link:	http://www.ti.com/lit/ds/symlink/ne555.pdf
Tags:	555 Timer
Description:	555 Timer

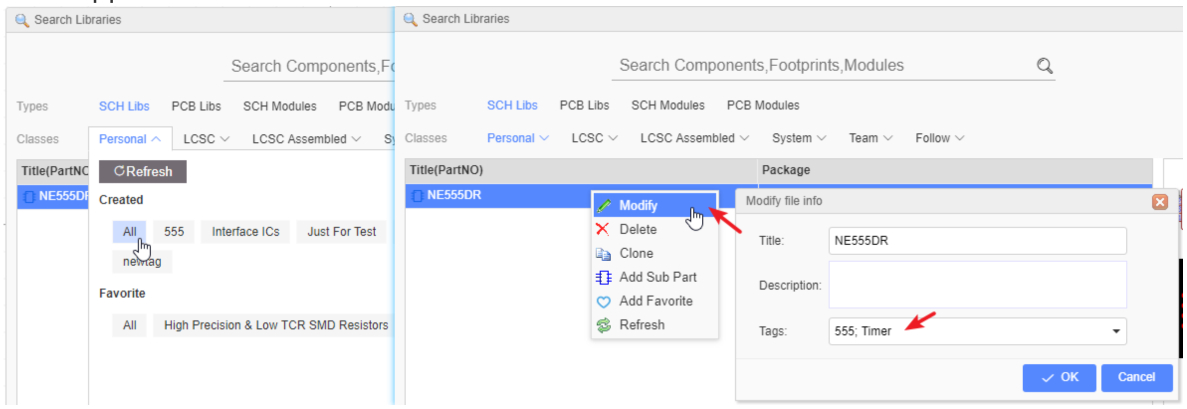
After clicking **Save**, you will see it appears in **Libraries** > **Symbols** > **Personal** of the left hand Navigation panel.


Search Libraries

Title(PartNO)	Package
NE555DR	SOIC-8_150MIL

SCH Libs > Personal > Created > 555 > NE555DR

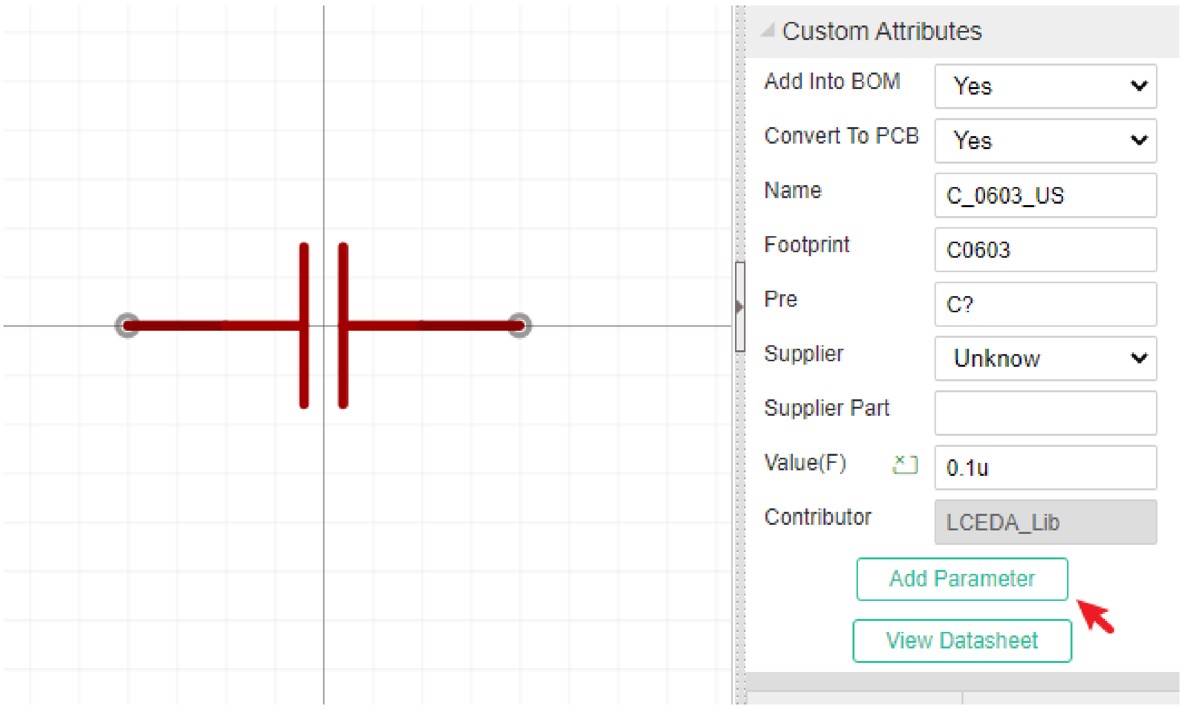
If you want to modify the tag for your new symbol: **Libraries** > **Symbols** > **Personal** > **Select New Lib** > **More** > **Modify**, or **right-click new Lib** > **Modify**, if your Lib doesn't have the tags it will appears on **All**.


Modify file info

Title:	NE555DR
Description:	
Tags:	555; Timer

Symbol Custom Attributes

In the Schematic Lib editor's canvas Properties panel, you will find a **Custom Attributes** section:


Custom Attributes

Add Into BOM	Yes
Convert To PCB	Yes
Name	C_0603_US
Footprint	C0603
Pre	C?
Supplier	Unknow
Supplier Part	
Value(F)	0.1u
Contributor	LCEDA_Lib

- **Add into BOM**
  This part display at BOM or not.
- **Convert to PCB**
  If you set it as No, this part will not display at Footprint Manager and can't not convert to PCB.
- **footprint**
  To assign a footprint for this part. Only assign one footprint.
  The more information please refer to **Schematic - Footprint Manager** section.
  Notie:
  You have to assign the footprint via the Footprint Manager, otherwise, the Schematic Symbol will not corresponding the Footprint correctly. The Footprint is linked with Symbol by global unique ID not the title.
- **Pre**
  The default Schematic symbol Prefix is **U?** If you create a resistor, you can set the Prefix to **R?**. It is filled required.
- **Name**
  You can change the schematic lib's name here, it is can be different from the part's file name.
- **Contributor**
  This is your registered user name. When Other EasyEDA's users use your libraries, they will remember your contributions!

Show symbol value as component name when place component at schematic

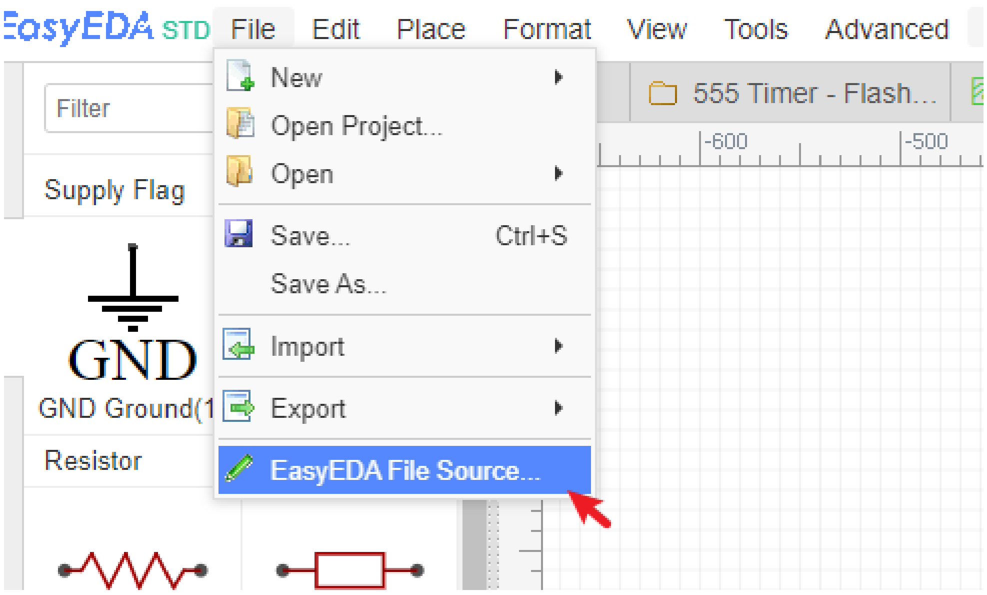
For example, a resistor symbol vaule is 2KΩ, name is ABC, but when place it at schematic, it will not show 2KΩ as component name, the name is ABC. You can change name to 2KΩ, but it not very well.

EasyEDA doesn't support common function to support this feature yet.

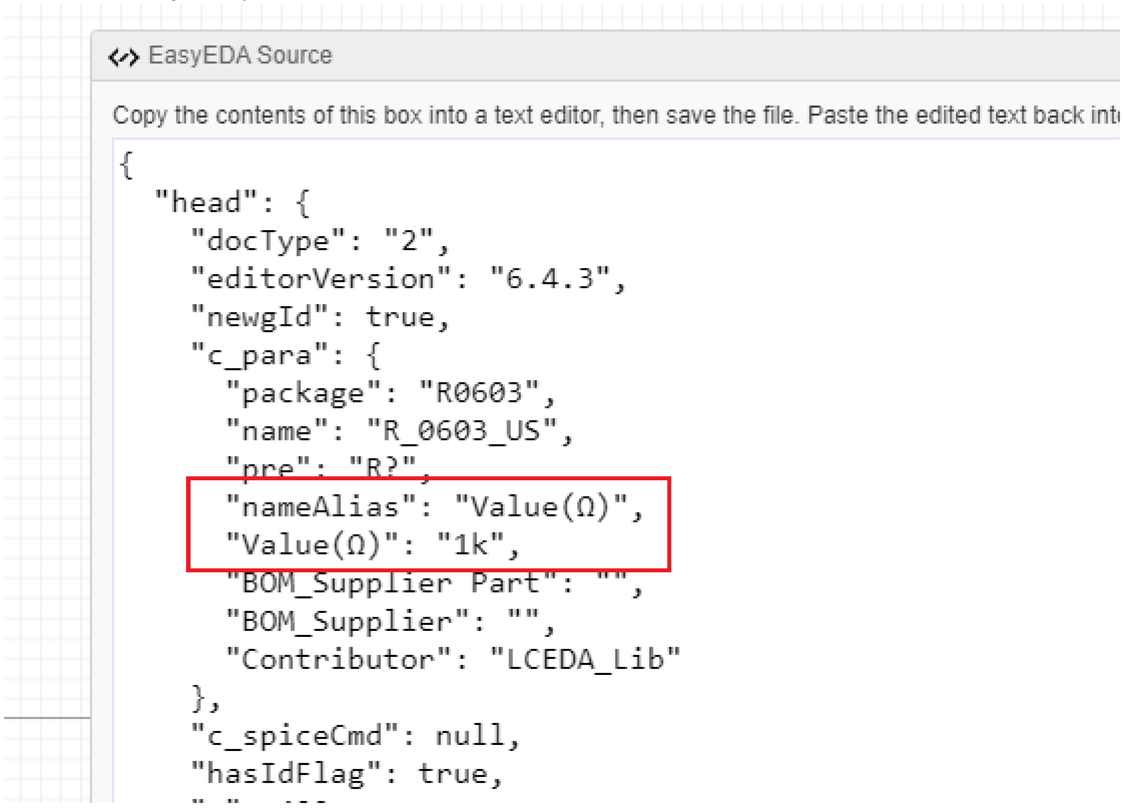
But, we can edit the symbol file soure to implement this feature.

How do it works:

1. Finish symbol and parameter edit.
2. Open file source. via: Top Menu - File - EasyEDA File Source.



3. Add or modify the parameter: nameAlias.



Copy the contents of this box into a text editor, then save the file. Paste the edited text back int

```
{
  "head": {
    "docType": "2",
    "editorVersion": "6.4.3",
    "newgId": true,
    "c_para": {
      "package": "R0603",
      "name": "R_0603_US",
      "pre": "R?",
      "nameAlias": "Value(Ω)",
      "Value(Ω)": "1k",
      "BOM_Supplier Part": "",
      "BOM_Supplier": "",
      "Contributor": "LCEDA_Lib"
    },
    "c_spiceCmd": null,
    "hasIdFlag": true,
```

This symbol will show 1k as component name after placing at schematic.

4. Apply after modified, and save.

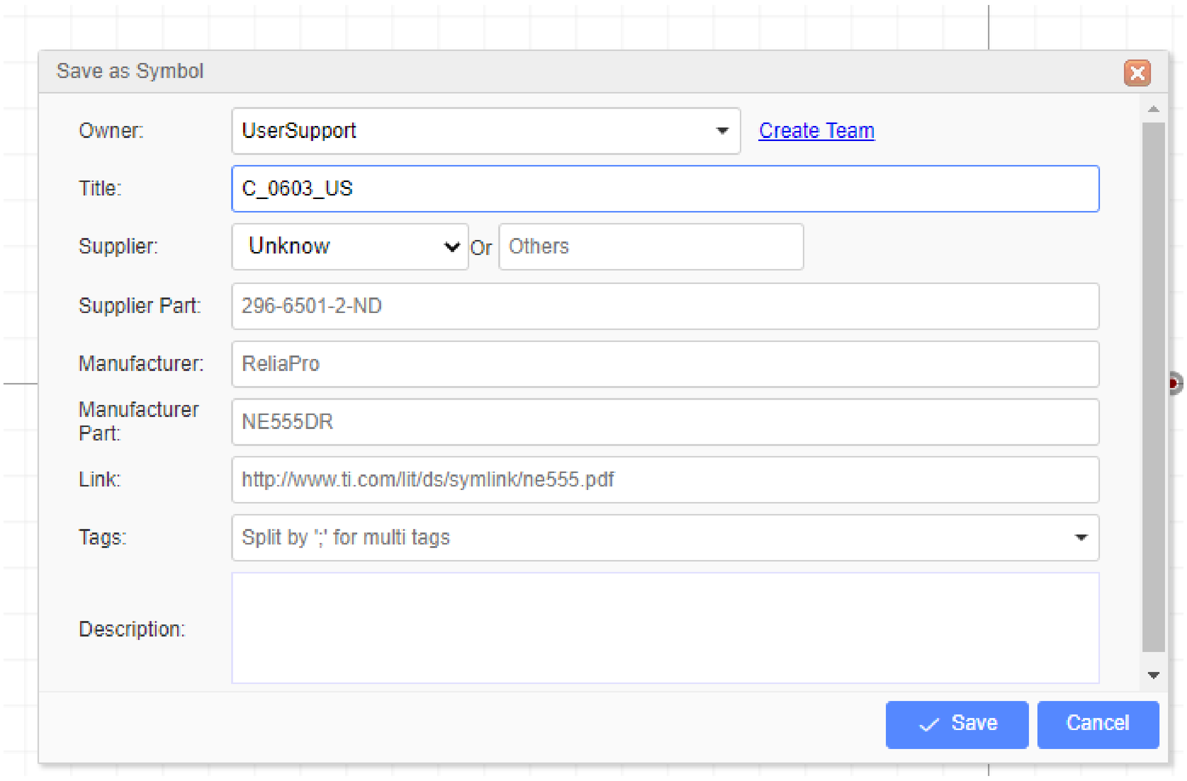
You can double click the EElib resistor symbol to get an example.

EasyEDA will provide it as a feature in the future.

Edit Exited Schematic Symbol

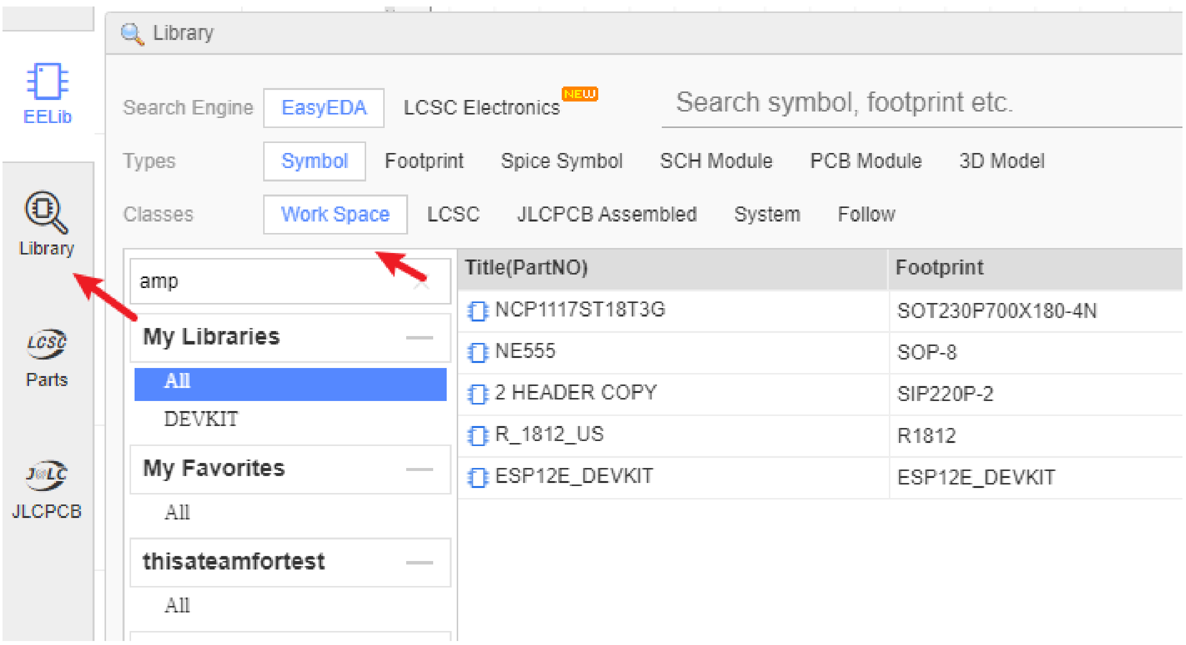
Personal Libraries

When you `CTRL+S` to save the Schematic Symbol, will pop up a dialog, you can choose this library's owner:


Save as Symbol

Owner:	UserSupport
Title:	C_0603_US
Supplier:	Unknow Or Others
Supplier Part:	296-6501-2-ND
Manufacturer:	ReliaPro
Manufacturer Part:	NE555DR
Link:	http://www.ti.com/lit/ds/symlink/ne555.pdf
Tags:	Split by ';' for multi tags
Description:	

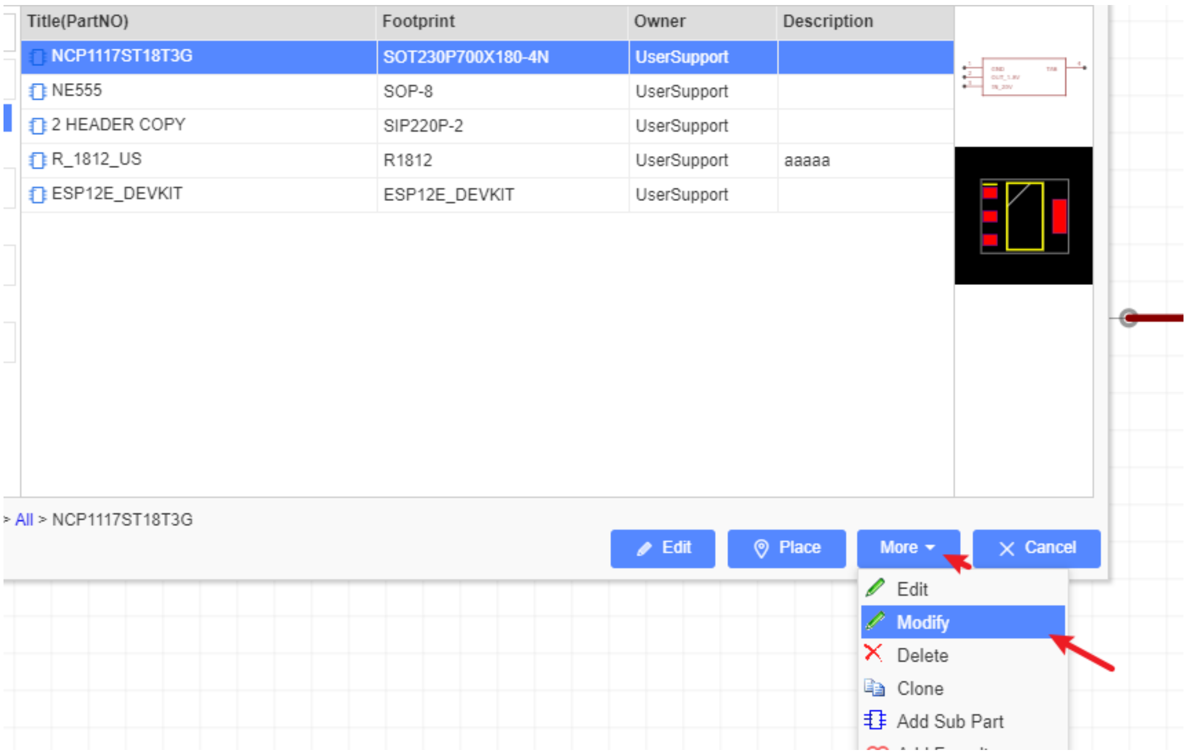
After finish, you can find your library at the left panel: **Library > Symbols > Work Space > All**



Title(PartNO)	Footprint
NCP1117ST18T3G	SOT230P700X180-4N
NE555	SOP-8
2 HEADER COPY	SIP220P-2
R_1812_US	R1812
ESP12E_DEVKIT	ESP12E_DEVKIT

Tag

When you select it , right-click it and select the menu "modify", you can add a tag for it.

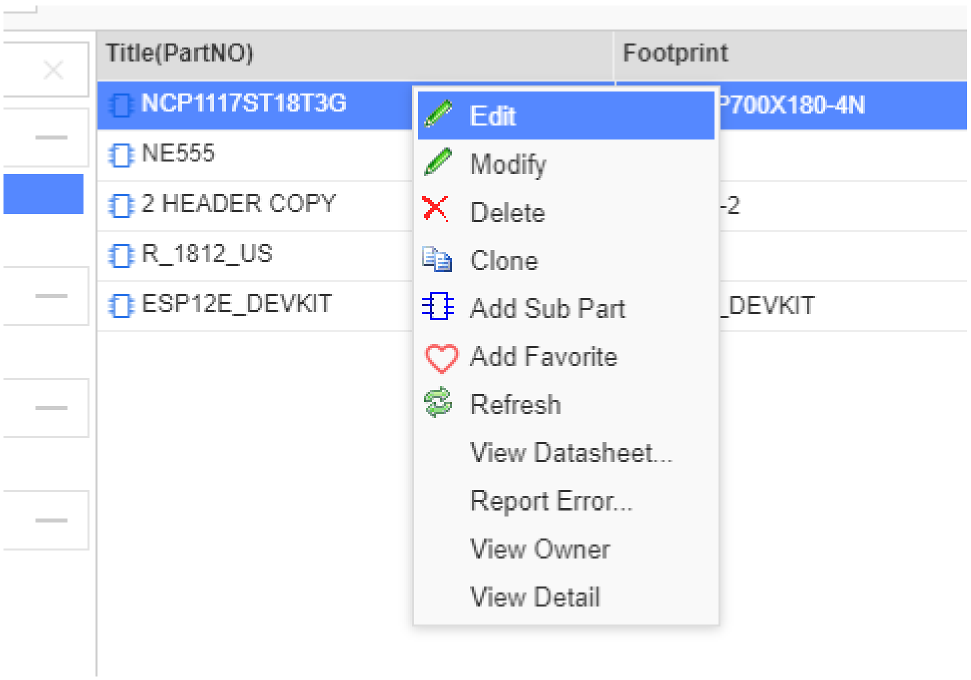


Title(PartNO)	Footprint	Owner	Description
NCP1117ST18T3G	SOT230P700X180-4N	UserSupport	
NE555	SOP-8	UserSupport	
2 HEADER COPY	SIP220P-2	UserSupport	
R_1812_US	R1812	UserSupport	aaaaa
ESP12E_DEVKIT	ESP12E_DEVKIT	UserSupport	

> All > NCP1117ST18T3G

Favorite

When you favorite a library, you can find it at **Library** > **Symbols** > **Work Spacel** > **Favorite** , If this library has a tag, the tag will show up too, but you can't edit that.
But you can via "Clone" or "Edit and save" to create a new library to personal libraries.



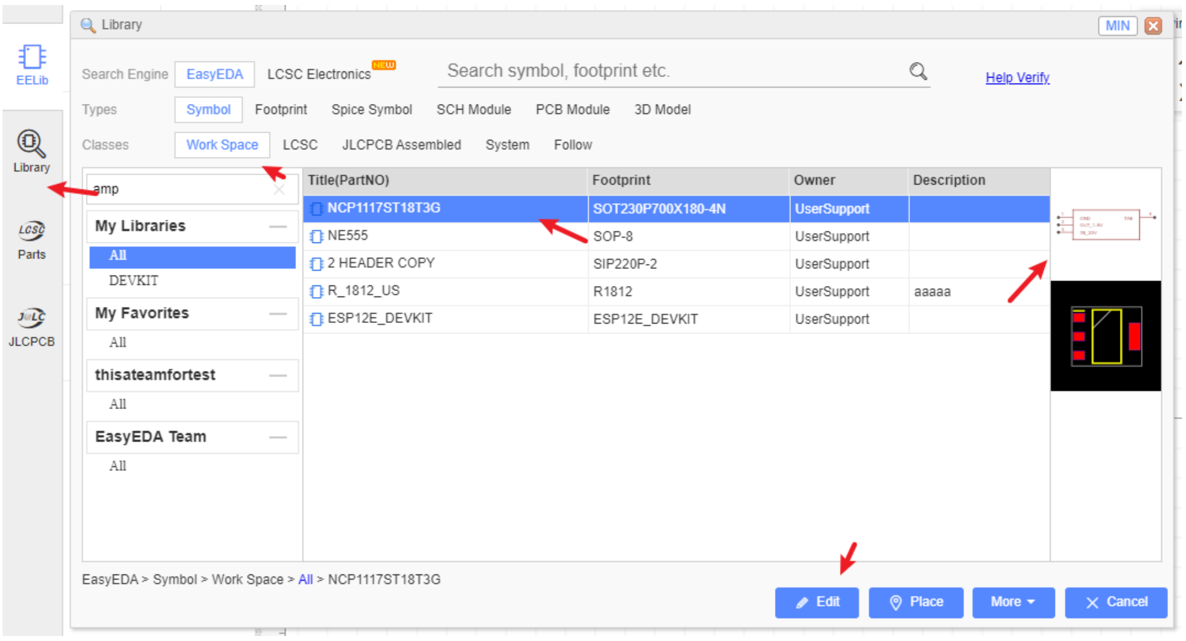
Title(PartNO)	Footprint
NCP1117ST18T3G	P700X180-4N
NE555	
2 HEADER COPY	-2
R_1812_US	
ESP12E_DEVKIT	_DEVKIT

Edit Symbol in the Library

When you feel the Schematic Libs can not be satisfied for you, you can edit it.

Via **"Library"** > **"Search Part/Work Space/LCSC/System"** > **Select Symbol** > **Edit**

or you can click the preview image



Library

Search Engine EasyEDA LCSC Electronics

Search symbol, footprint etc.

Types Symbol Footprint Spice Symbol SCH Module PCB Module 3D Model

Classes Work Space LCSC JLCPCB Assembled System Follow

amp

My Libraries
All
DEVKIT

My Favorites
All

thisateamfortest
All

EasyEDA Team
All

Title(PartNO)	Footprint	Owner	Description
NCP1117ST18T3G	SOT230P700X180-4N	UserSupport	
NE555	SOP-8	UserSupport	
2 HEADER COPY	SIP220P-2	UserSupport	
R_1812_US	R1812	UserSupport	aaaaa
ESP12E_DEVKIT	ESP12E_DEVKIT	UserSupport	

EasyEDA > Symbol > Work Space > All > NCP1117ST18T3G

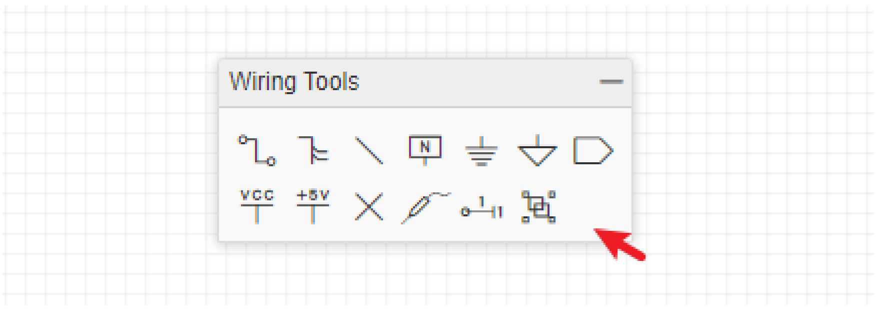
Edit Place More Cancel

when you finish and save, it will be saved to your personal libraries **Work Space** and become your personal libraries.

Edit Symbol in the Schematic

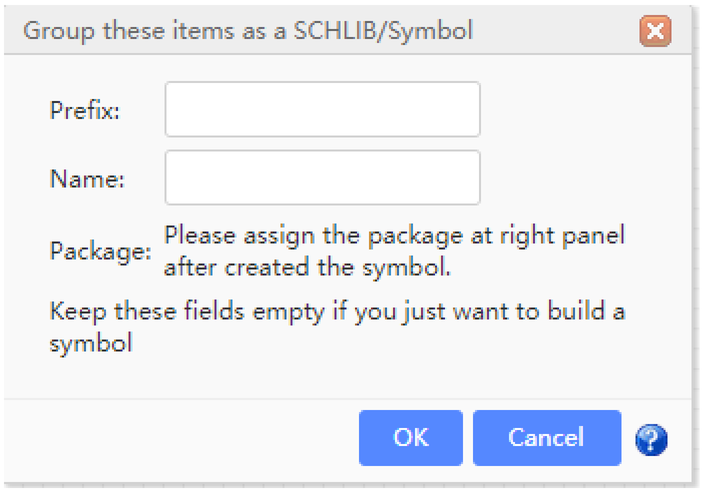
If you want to edit a symbol in the schematic, you can use the Ungroup/Group function.

On the **Wiring Tools** palette there is the **Group/Ungroup Symbol...** button.



This tool is for you to quickly create or edit schematic library symbols.

1. Select a symbol
2. Click the **Group/Ungroup Symbol...** button
   Up to this point you have a collection of separate pins, a drawn rectangle and some text that are all separate items with no particular association with each other.
3. Edit the shape or pin what you want to change
4. Select all of the items and click the **Group/Ungroup Symbol...** button.
   A dialog will be opened:


Group these items as a SCHLIB/Symbol

Prefix:

Name:

Package: Please assign the package at right panel after created the symbol.

Keep these fields empty if you just want to build a symbol

After you click OK, all those separate elements will be grouped together to form your new symbol directly in the schematic.

Using the group function, you can create/edit any symbol in the schematic, easily and quickly.

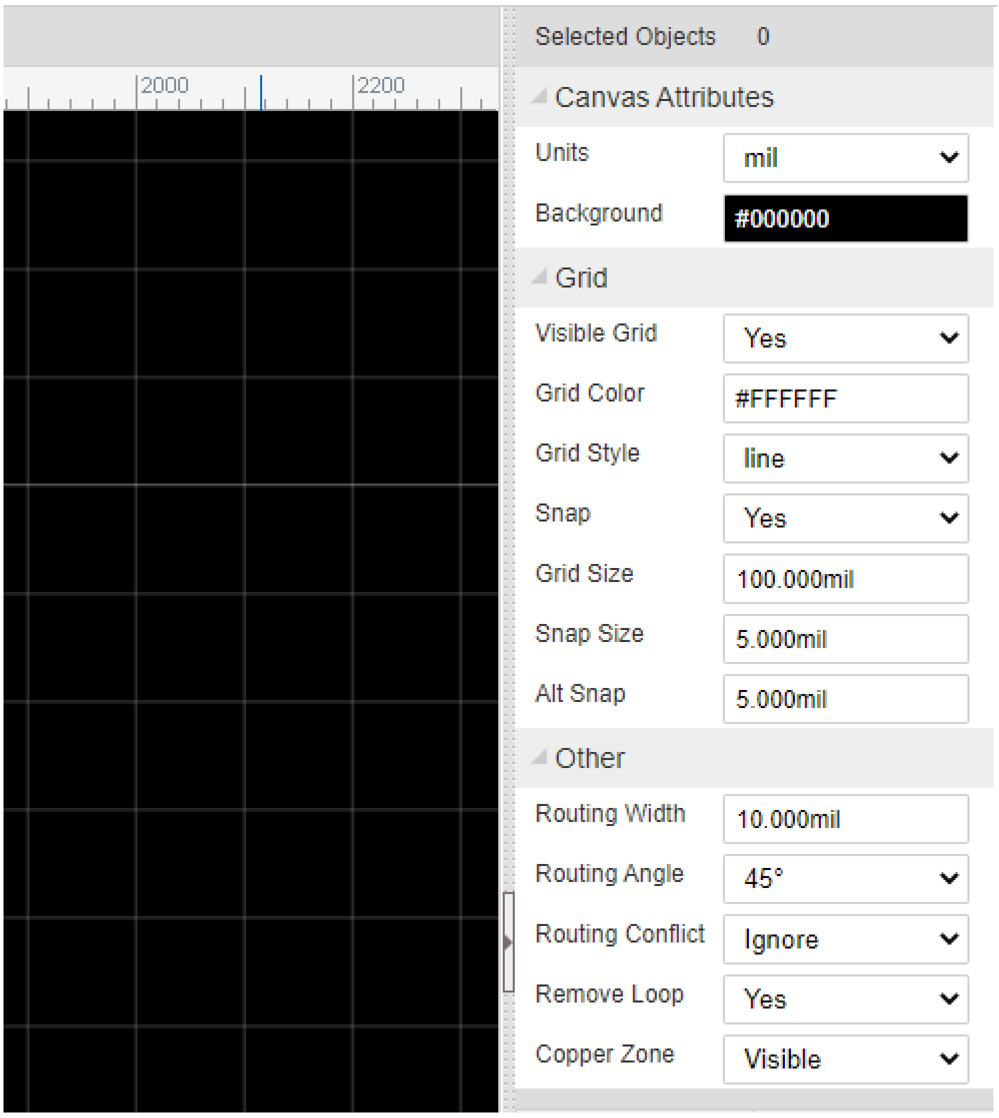
PCB Layout

Canvas Setting

After the initial conversion of a schematic to PCB, it is time to learn how to manage EasyEDA's PCB Design Editor.

Canvas Attributes

Lots of PCB canvas attributes are the same as Schematic canvas attributes. The key is that you can set **units** in PCB canvas attributes.



Selected Objects 0

Canvas Attributes

Units	mil
Background	#000000

Grid

Visible Grid	Yes
Grid Color	#FFFFFF
Grid Style	line
Snap	Yes
Grid Size	100.000mil
Snap Size	5.000mil
Alt Snap	5.000mil

Other

Routing Width	10.000mil
Routing Angle	45°
Routing Conflict	Ignore
Remove Loop	Yes
Copper Zone	Visible

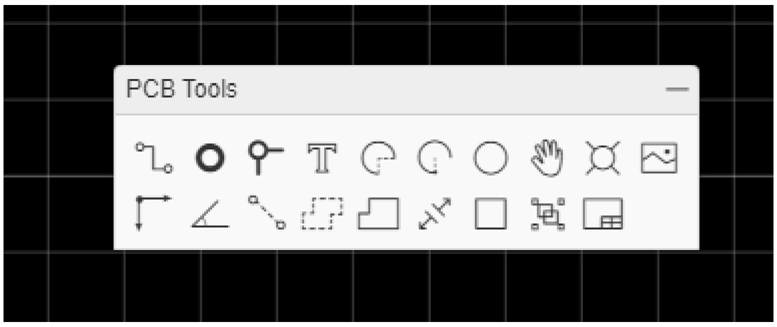
When you select a object at the canvas, you can modify its attributes at the right panel.

Snap Size: The cursor snapping size.
Alt Snap: When press hotkey ALT the cursor snapping size.
Other

- **Routing Width**: Setting the default routing width.
- **Routing Angle**: Setting the routing angle.
- **Routing Conflict**: When routing the track, what to do when impact the difference net objects.
  - **Ignore**: The track go through the objects.
  - **Block**: The track will stop when meet the difference net objects.
  - **RoundTrack**: The track will go aroud the difference net objects.
- **Remove Loop**: Remove the track loop.
- **Copper Zone**: Setting the copper zone visible or invisible.

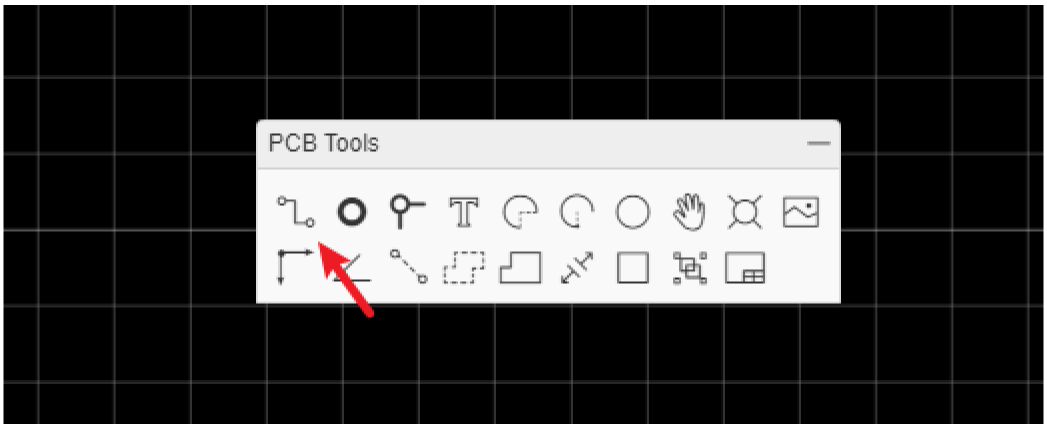
PCB Tools

PCB tools provide many function to fulfill your PCB design requirement.
Such as: Track, Pad, Via, Text, Arc, Circle, Move, Hole, Image, Canvas Origin, Connect Pad to Pad, Copper Area, Solid Region, Measure/Dimension, Rect, Group/Ungroup. etc.

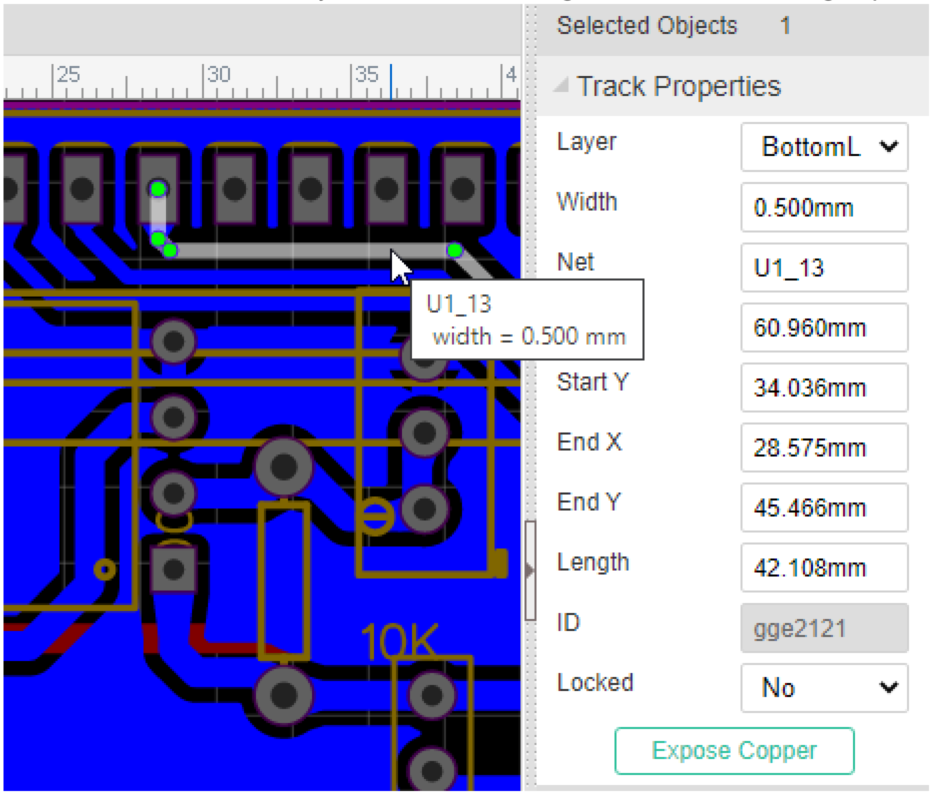


Track

In the schematic editor, we use Wire or the `W` Hotkey to connect Pins, in a similar way in the PCB editor, we use Track to connect Pads. Track allows you to draw PCB tracks and can be found on the PCB Tools palette or using the `W` Hotkey (not T: see above!).



When a track is selected, you can find its Length attribute in the right panel.



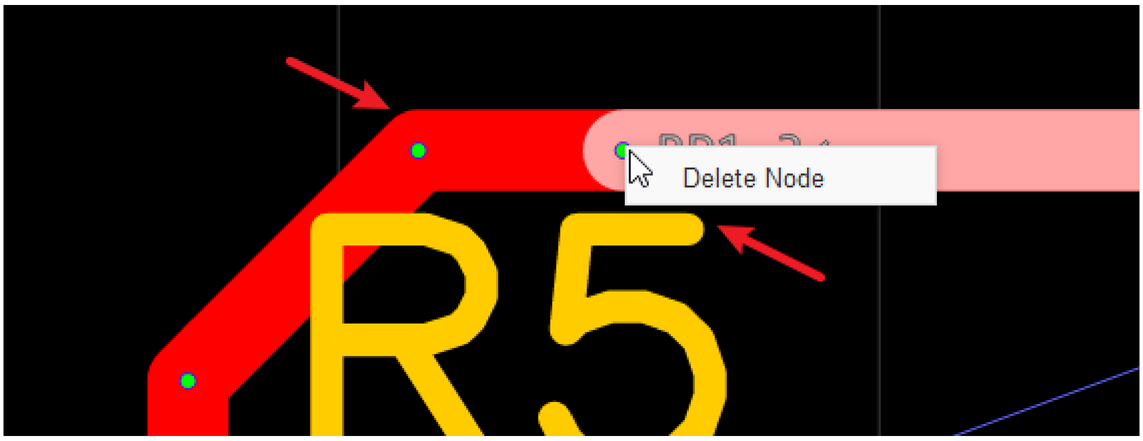
Selected Objects 1

Track Properties

Layer	BottomL
Width	0.500mm
Net	U1_13
	60.960mm
Start Y	34.036mm
End X	28.575mm
End Y	45.466mm
Length	42.108mm
ID	gge2121
Locked	No

If you want to create solder mask for the track, you can click the "Expose Copper" button at the right-hand property panel.

When click the track, you will see the nodes, you can drag it or right-click delete it.

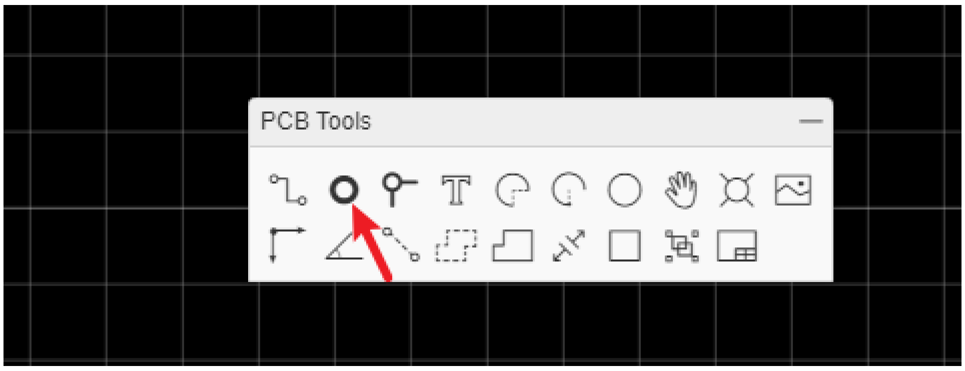


when select the point to point separated tracks, you can convert them as Solid Region or continus track at right-click menu.

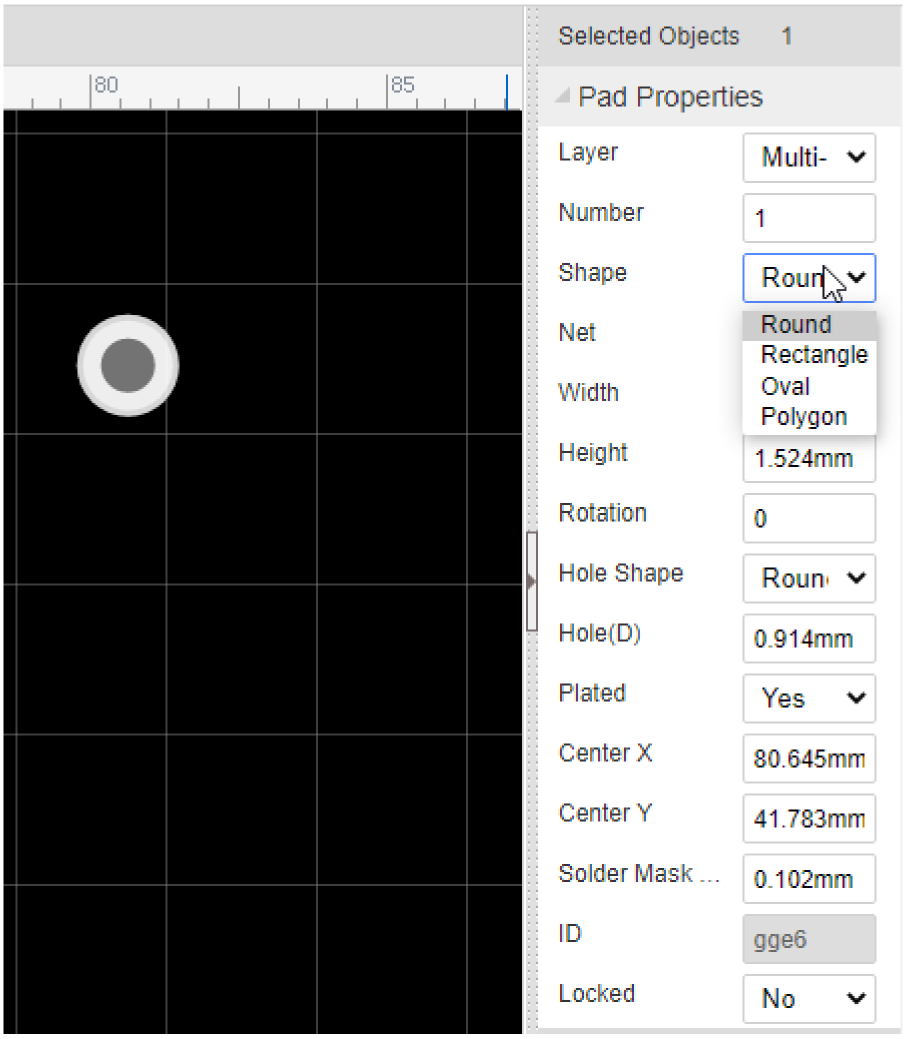
The more information of routing, please refer at PCB: Route Tracks

Pad

You can add pads using the Pads button from the Footprint Tools palette or using the `P` hotkey.



After selecting one of the pads, you can view and adjust its attributes in the right hand Properties panel.



Selected Objects 1

Pad Properties

Layer	Multi-
Number	1
Shape	Roun
Net	
Width	
Height	1.524mm
Rotation	0
Hole Shape	Roun
Hole(D)	0.914mm
Plated	Yes
Center X	80.645mm
Center Y	41.783mm
Solder Mask ...	0.102mm
ID	gge6
Locked	No

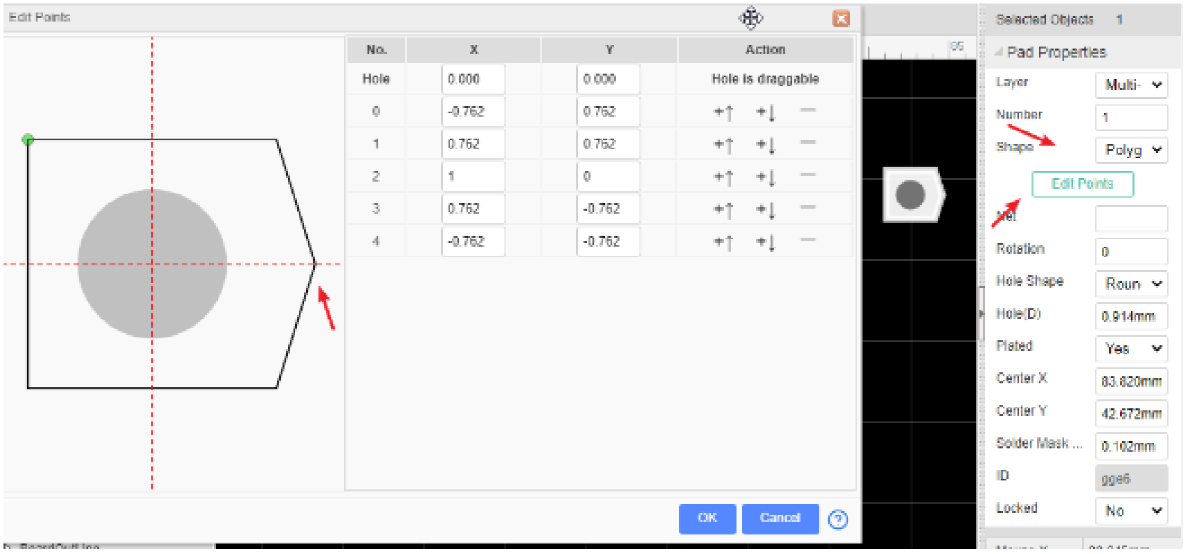
Number: Remembering the pin numbers you set in the schematic symbol in your Schematic Lib: to connect those schematic symbol pins to the pads in your PCB footprint, the pad numbers you set here in the Footprint footprint must be the same.

Shape: Round , Rectangular , Oval and Polygon.

EasyEDA supports four shapes: `Round` , `Rectangular` , `OVAL` and `POLYGON`.

- `OVAL` PAD will give your more space.
- `POLYGON` PAD will let you to create some strange pad.

Like in the image below, you can edit the PADs points when you select a `POLYGON` PAD



Edit Points

No.	X	Y	Action
Hole	0.000	0.000	Hole is draggable
0	-0.762	0.762	+↑ +↓ —
1	0.762	0.762	+↑ +↓ —
2	1	0	+↑ +↓ —
3	0.762	-0.762	+↑ +↓ —
4	-0.762	-0.762	+↑ +↓ —

Selected Objects 1

Pad Properties

Layer	Multi-
Number	1
Shape	Polyg
Net	
Rotation	0
Hole Shape	Roun
Hole(D)	0.914mm
Plated	Yes
Center X	83.820mm
Center Y	42.672mm
Solder Mask ...	0.102mm
ID	gge6
Locked	No

Layer: If the pads are part of a **SMD** footprint, you can set it to **Top layer** or **Bottom layer**. For through hole components you should set it to **Multi-Layer**. If it setting as mult-layer, it will connect with all copper layers.

Net: You don't need to enter anything here because at present this footprint is not connected to anything in a circuit.

Width and Height: When the shape is set to Round, Width will equal Height.

Rotation: Here you can set the Pad's rotation as you want.

Hole(D): This is the drill hole **diameter** for a through hole pad. For a SMD Pad, set its layer to **TopLayer or BottomLayer**.

Hole Shape: Round and Slot. When it is set as a slot, the Gerber is generated through the stitching of multiple drill holes in the corresponding position. If your hole is round, please do not set it as a slot, so as to avoid the overlapping error of holes during the production of DFM detection.

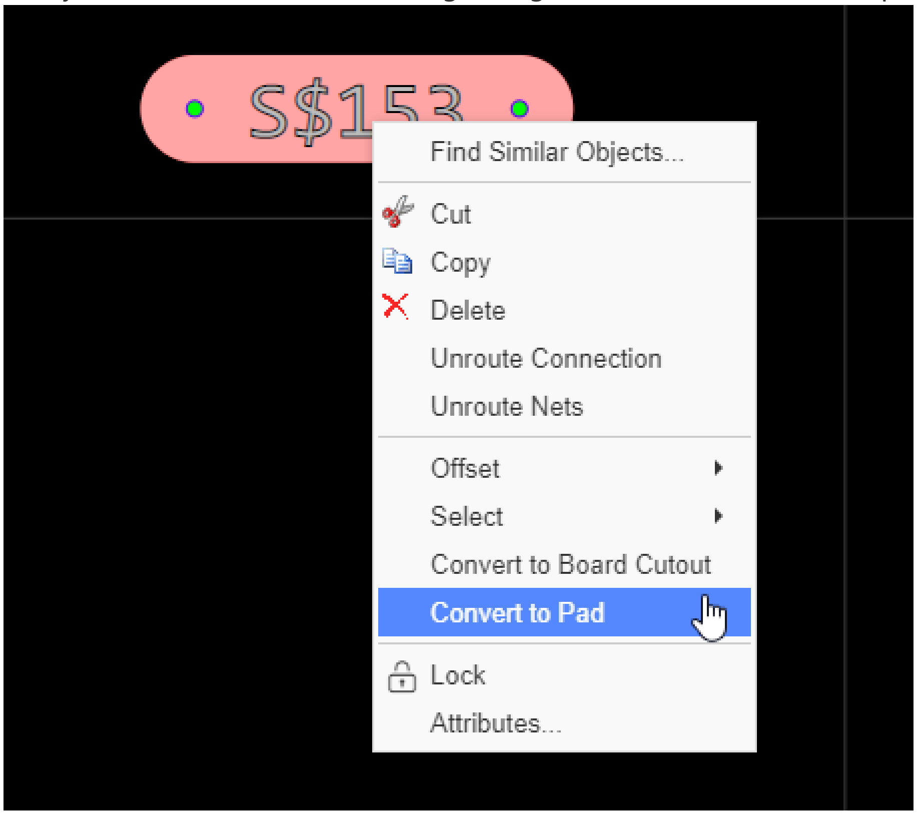
Center-X and Center-Y: using these two attributes, you can set the pad's position with more precision, compared to using the mouse.

Plated: Yes or No. When you set it as No, this pad Inner wall do not metallization.

Paste Mask Expansion: For single layer pad. This property affects the size of the tin area on the plate of the steel mesh. If you want to set a pad that is not open in the steel mesh, you can set the value to be negative, which is usually larger than the diagonal of the pad.

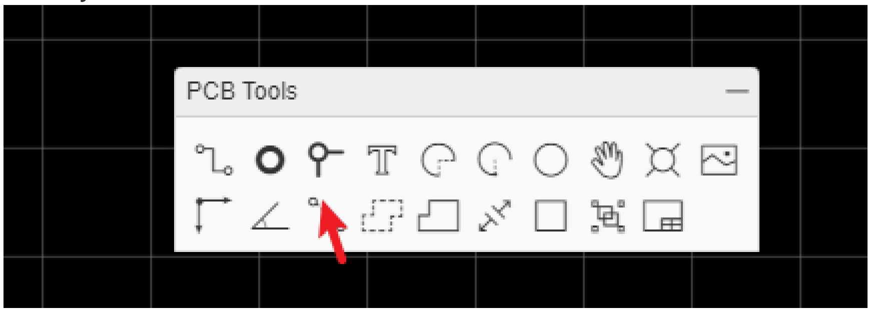
Solder Mask Expansion: This property affects the size of the green oil area cover on the pad. If you want to set a pad not open covered with green oil, you can set the value to be negative, the value is usually set larger than the diagonal of the pad.

And you can select a track/Solid Region, right-click it and convert to a pad.



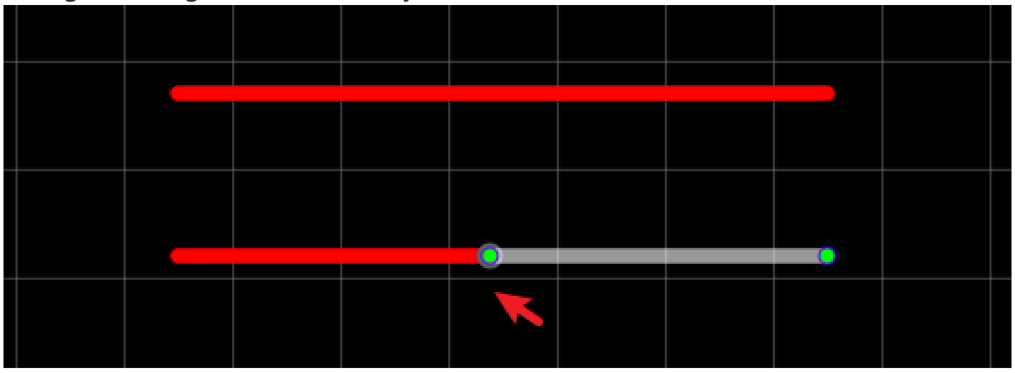
Via

When you want to lay a multilayer PCB, you need to add Vias for nets getting through layer and layer.



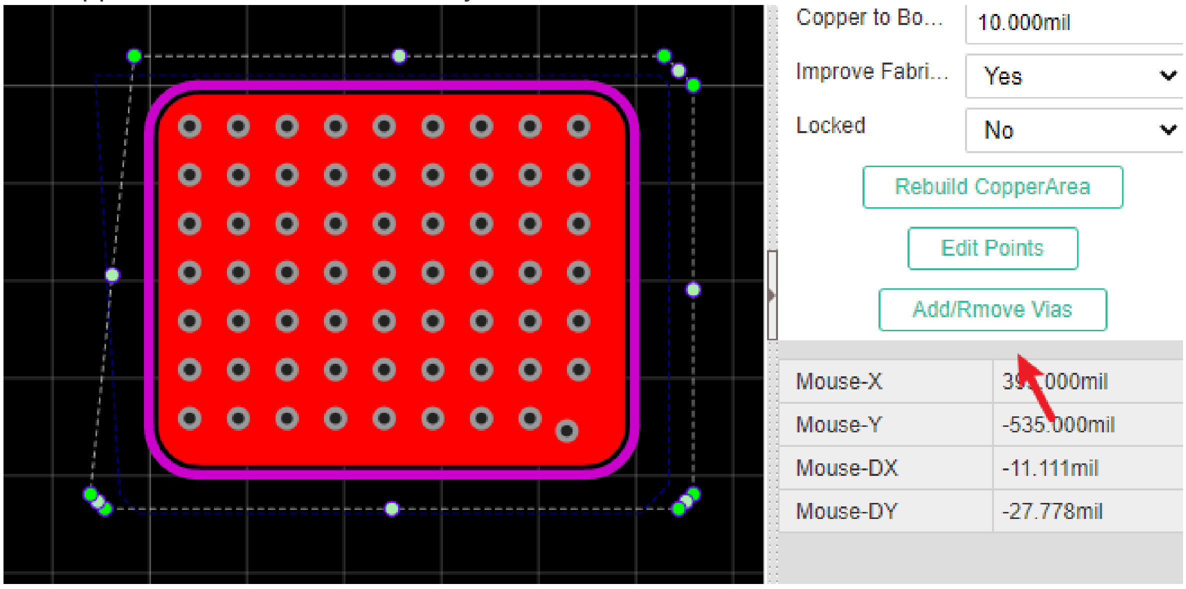
Place a Via on a Track

When placing a `via` on a track, the track will be cut to two segments, and the via net will follow track's net. Placing two vias on a tracks, you will get three segments, then you can change one segment to other layer id, or remove one of them.



Place Multiple Vias

Click the copper area outline, click the "Add/Remove Vias" button. this feature needs the same net copper areas on two and more layers in the same time, the cross area will add the vias.



Copper to Bo...	10.000mil
Improve Fabri...	Yes
Locked	No

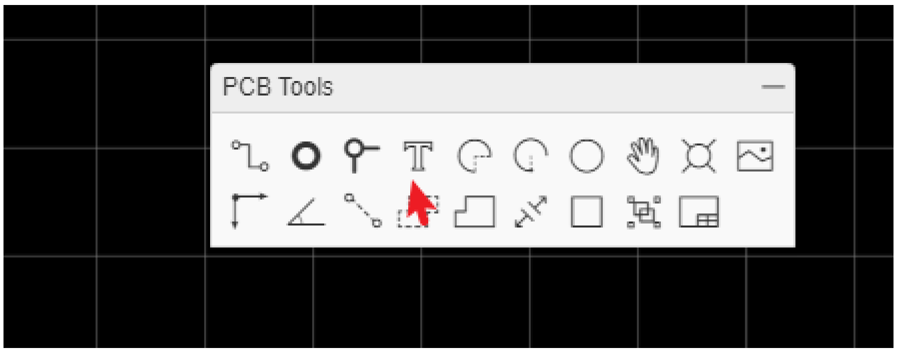
Mouse-X	39[illegible]000mil
Mouse-Y	-535.000mil
Mouse-DX	-11.111mil
Mouse-DY	-27.778mil

Notice:

- EasyEDA only support the through via for all layers, doesn't support the buried/blind via.

Text

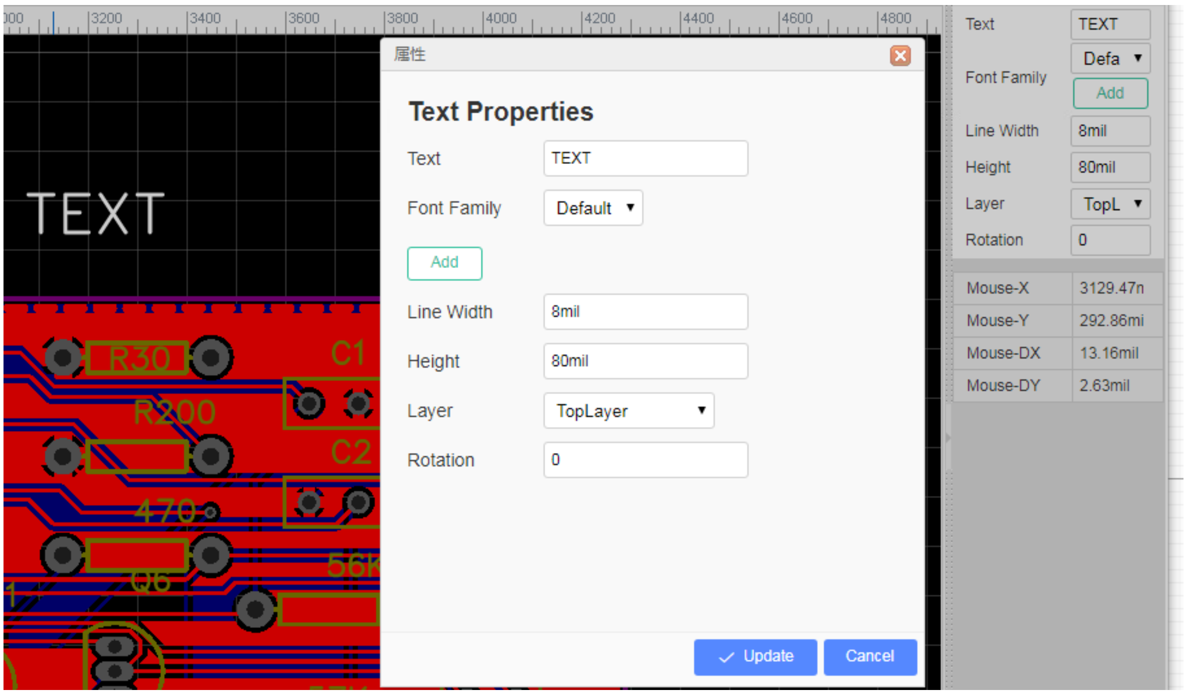
You can add more fonts from your computer or download [free fonts: http://www.fontspace.com/](http://www.fontspace.com/).



if you need Japanese or Korean you can use [Google Noto fonts]

The editor including fonts are:
(NotoSansCJKsc-DemiLight)[https://github.com/googlefonts/noto-cjk/blob/main/Sans/OTF/SimplifiedChinese/NotoSansCJKsc-DemiLight.otf]
(NotoSerifCJKsc-Medium)[https://github.com/googlefonts/noto-cjk/blob/main/Serif/OTF/SimplifiedChinese/NotoSerifCJKsc-Medium.otf]

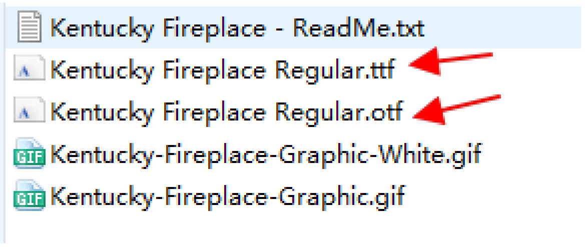
Select the text, then you can find a Font-family attribute on the right panel like in the image below.



Text	TEXT
Font Family	Default
Line Width	8mil
Height	80mil
Layer	TopLayer
Rotation	0

Mouse-X	3129.47n
Mouse-Y	292.86mi
Mouse-DX	13.16mil
Mouse-DY	2.63mil

Click the add button, then choose the font, the font file must be `ttf` or `otf`.



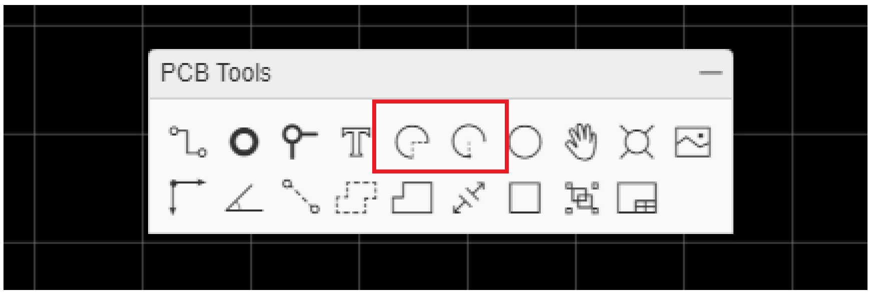
Kentucky Fireplace - ReadMe.txt
Kentucky Fireplace Regular.ttf
Kentucky Fireplace Regular.otf
Kentucky-Fireplace-Graphic-White.gif
Kentucky-Fireplace-Graphic.gif

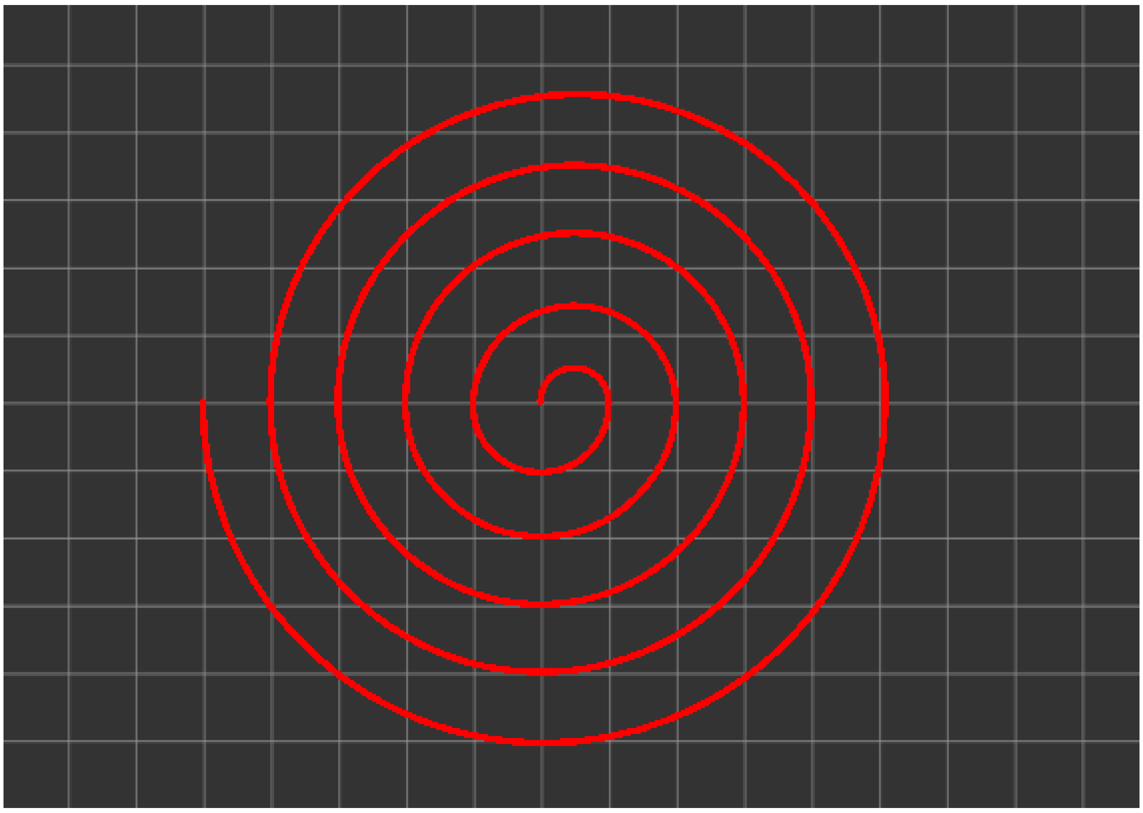
So you can add any fonts by yourself. EasyEDA doesn't cache the font on our server, so if you close the editor, you need to add the font again by yourself.

Note: *If you use the other font, the* `LineWidth` *attribute is useless, because it will be automatically set by changing the* `Height`.

Arc

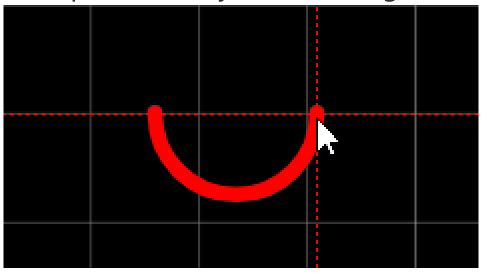
You can draw many Arcs with different sizes, it's easy to create a pretty cool PCB as you like.



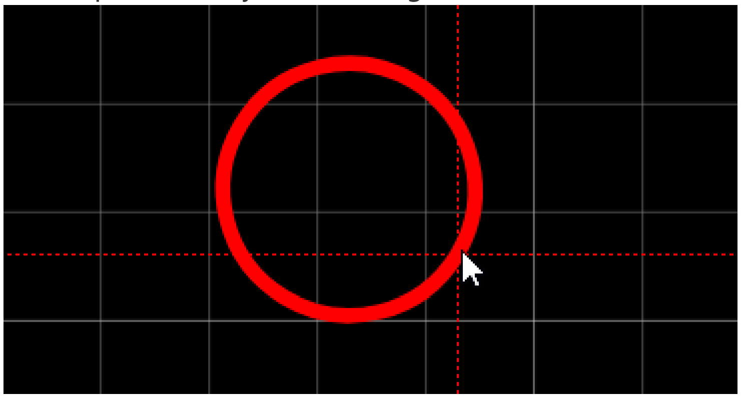


EasyEDA provides two Arc tools:

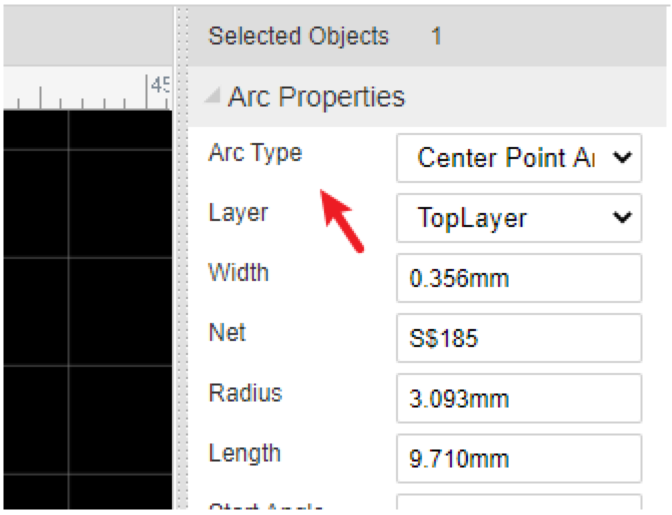
- Start point fixed, you can change the end point position and radius.



- Center point fixed, you can change the radius.



Select the arc, you can change the arc type at property panel, different arc type has different drag behavior.

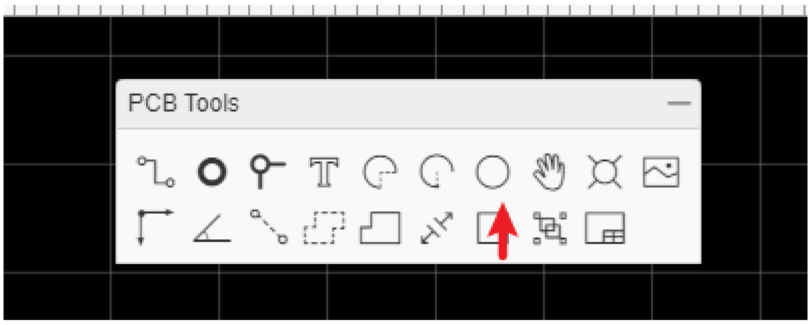

Selected Objects 1

Arc Properties

Arc Type	Center Point Ar
Layer	TopLayer
Width	0.356mm
Net	S$185
Radius	3.093mm
Length	9.710mm

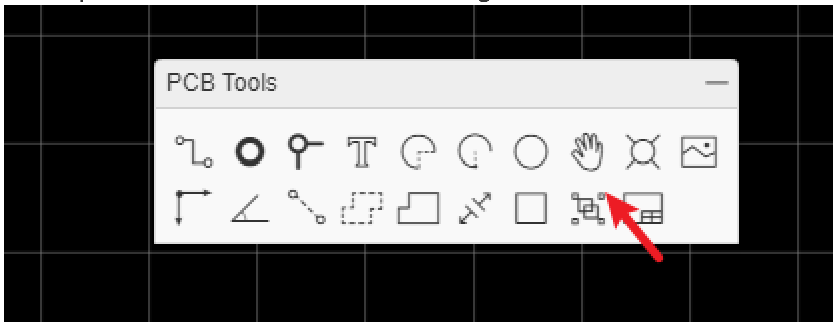
Circle

You can draw a circle in PCB. If you want to draw a circle at TopLayer or BottomLayer, please use Arc.



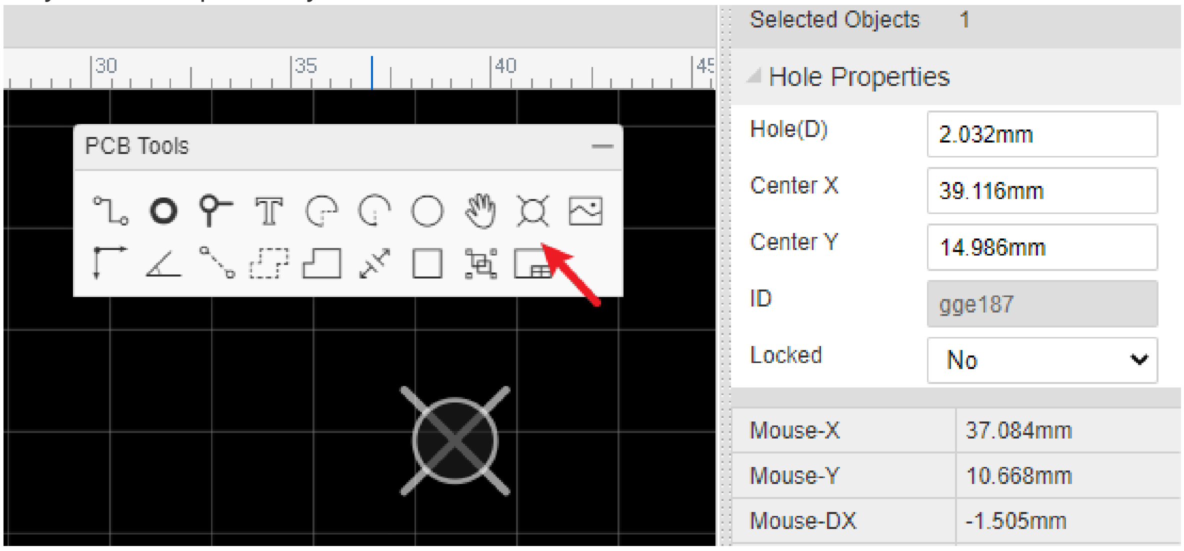
Move

This option is same as schematic's drag.



Hole

There were lots of users that didn't know how to use PAD or VIA as a HOLE, they asked EasyEDA for help, so EasyEDA added a HOLE TOOL in the PCB toolbar.



Selected Objects 1

Hole Properties

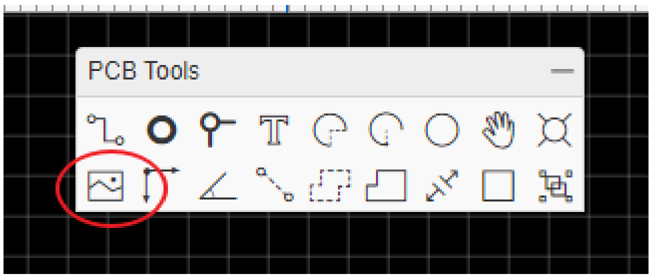
Hole(D)	2.032mm
Center X	39.116mm
Center Y	14.986mm
ID	gge187
Locked	No

Mouse-X	37.084mm
Mouse-Y	10.668mm
Mouse-DX	-1.505mm

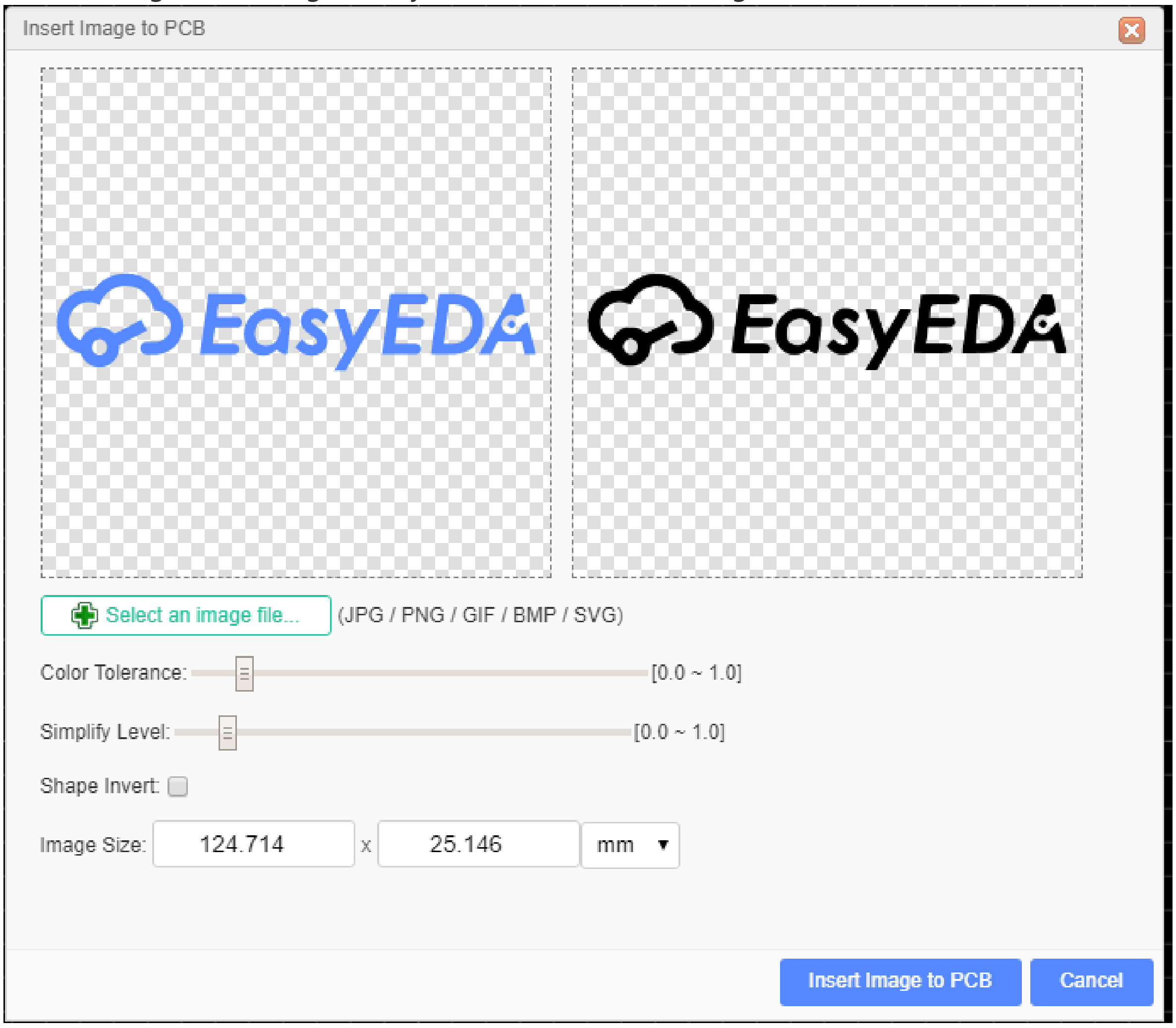
And if you want to create the slot hole, you can use solid region(Type: NPTH), or route a track, and then right-click the "Convert to NPTH" menu.

Image

On PCB and Footprint editor, there is a nice feature on the PCB Tools bar.



After clicking on the image icon, you will see the Insert Image window as below.



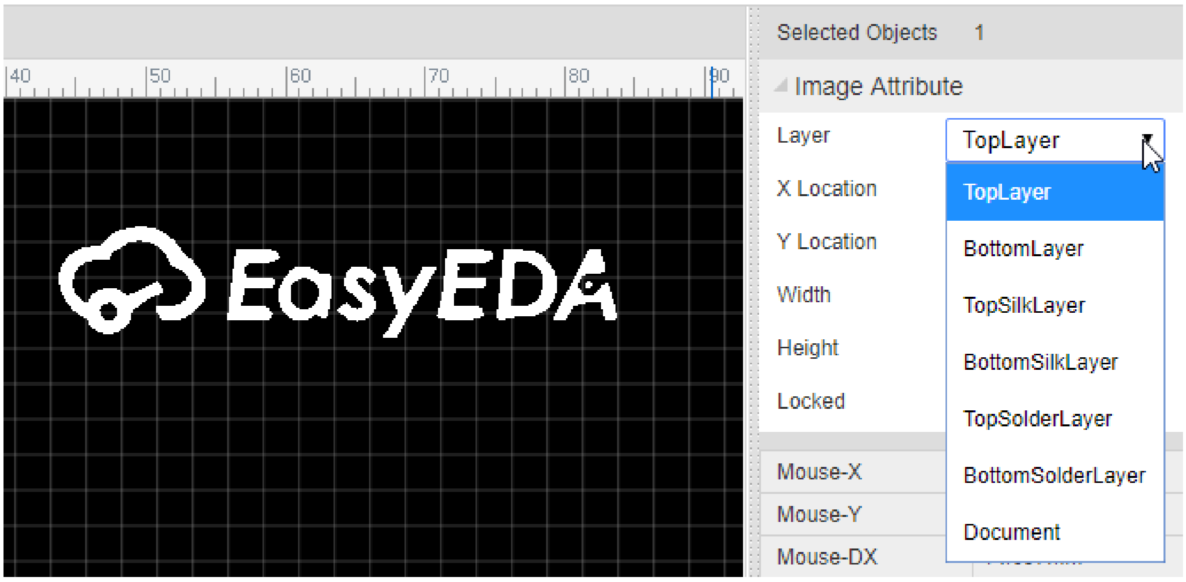
Color Tolerance: [0.0 ~ 1.0]
Simplify Level: [0.0 ~ 1.0]
Shape Invert: ☐
Image Size: 124.714 x 25.146 mm

In this dialog, you can choose your favorite image, EasyEDA support `JPG`, `BMP`, `PNG`, `GIF`, and `SVG`. Unlike some other EDA tools which only support a Monochrome Bitmap image, EasyEDA supports full color, but Monochrome Bitmap is welcome.

You can adjust the color tolerance, simplify level and reset the image size there.

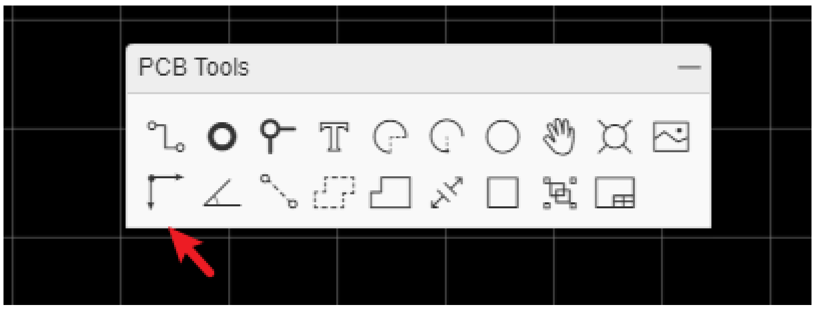
And you can select shape invert.

The image will be inserted to the active layer, if it is not right, you can change the attribute. Such as TopSilkLayer.



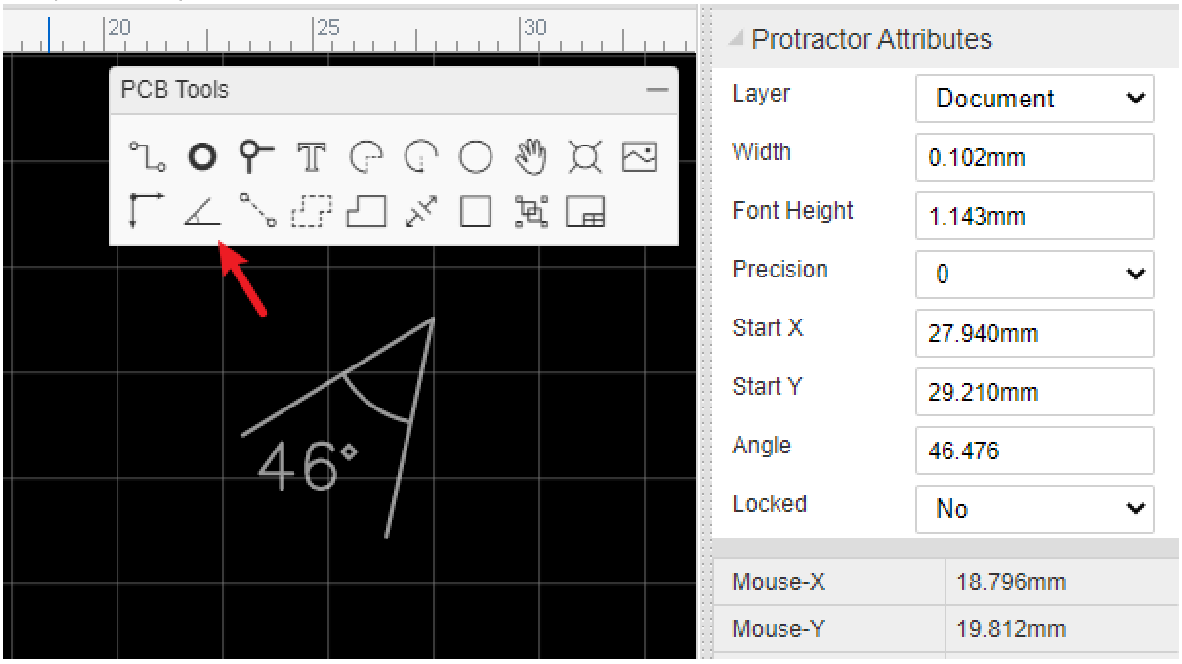
Canvas Origin

This option is the same as schematic's Canvas Origin.



Protractor

We provide a protractor for PCB tools.

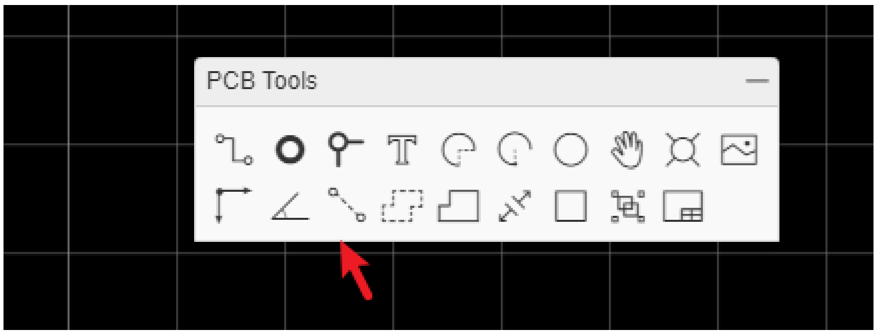


Protractor Attributes

Layer	Document
Width	0.102mm
Font Height	1.143mm
Precision	0
Start X	27.940mm
Start Y	29.210mm
Angle	46.476
Locked	No
Mouse-X	18.796mm
Mouse-Y	19.812mm

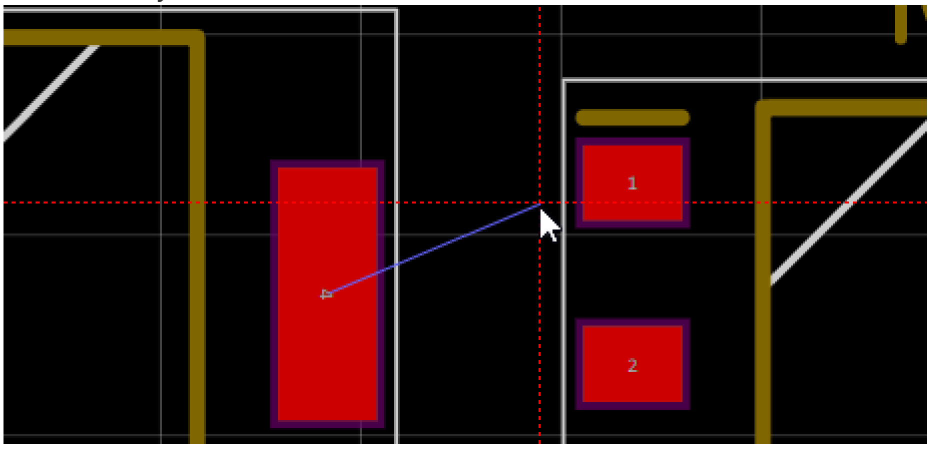
Connect Pad to Pad

When creating a PCB without a Schematic, none of the pads on the Footprints have nets connecting them so there will be no ratlines.



Rather than try to track the pads from scratch, it is a good idea to connect them up by hand first using `Connect Pad to Pad` from the PCB Tools palette. This will help you to remember to track the pads correctly with fewer mistakes.

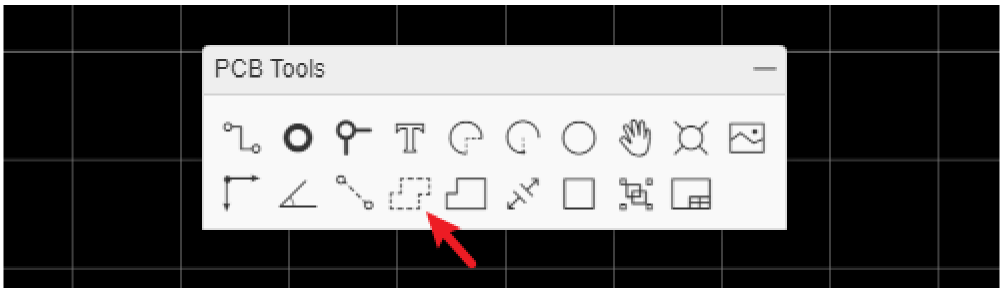
You could also do this by setting net names for all the pads: if the two pads are given the same net name then EasyEDA will understand that they are connected together and will automatically create a ratline between them.



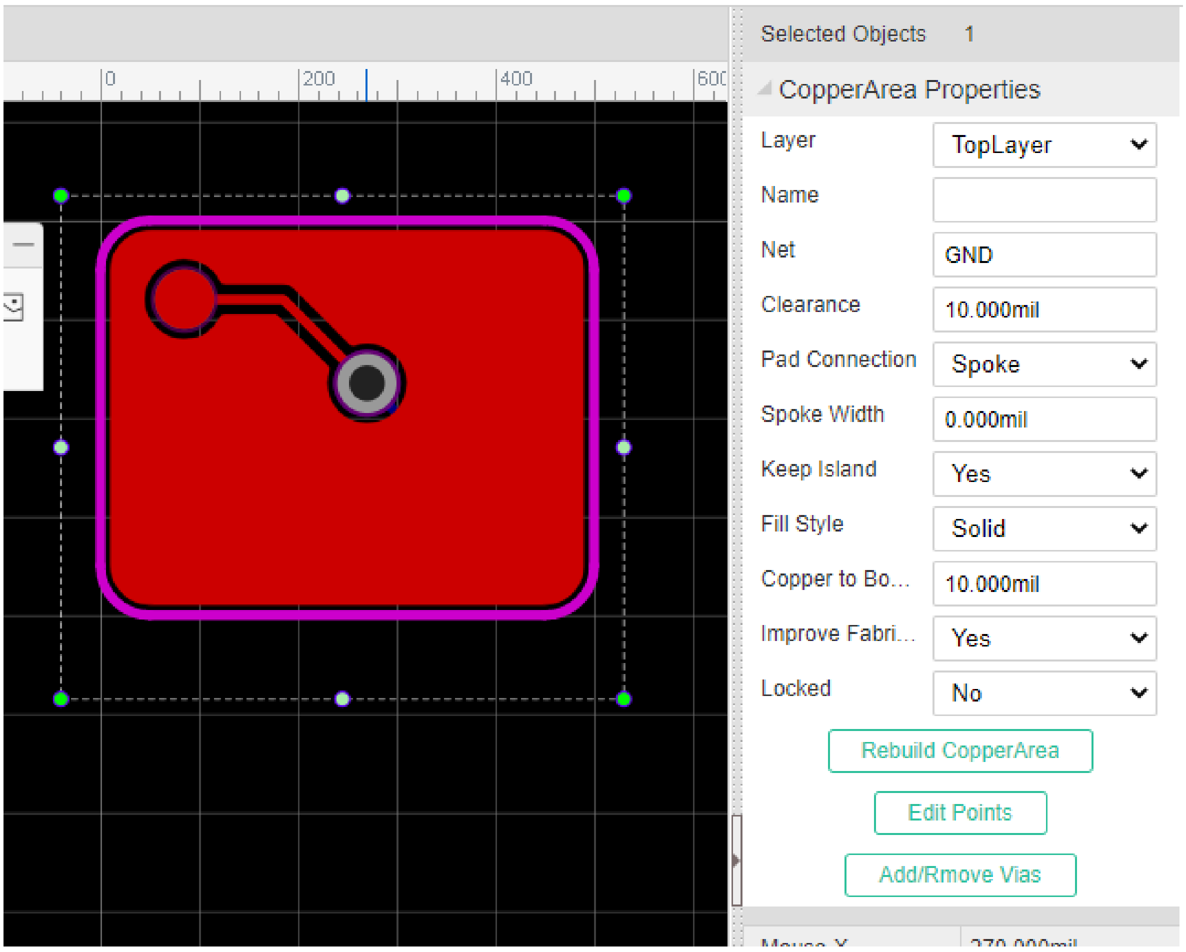
Or you can set these two pads with the same net name at the right panel Pad Properties after you click the pad.

Copper Area

Sometimes you will want to fill in or flood an area with copper. Usually this copper area will be connected to a net such as **GND** or a supply rail. You can draw the outline of a flood using the **Copper Area** button from the PCB Tools palette.



When selecting a copper area, you can find its attributes from the right hand **Properties** panels.


Selected Objects 1

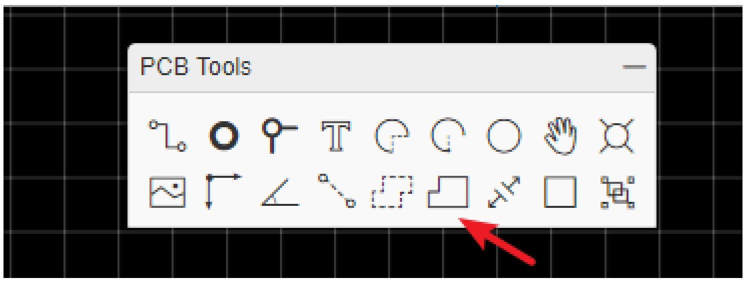
CopperArea Properties

Layer	TopLayer
Name	
Net	GND
Clearance	10.000mil
Pad Connection	Spoke
Spoke Width	0.000mil
Keep Island	Yes
Fill Style	Solid
Copper to Bo...	10.000mil
Improve Fabri...	Yes
Locked	No

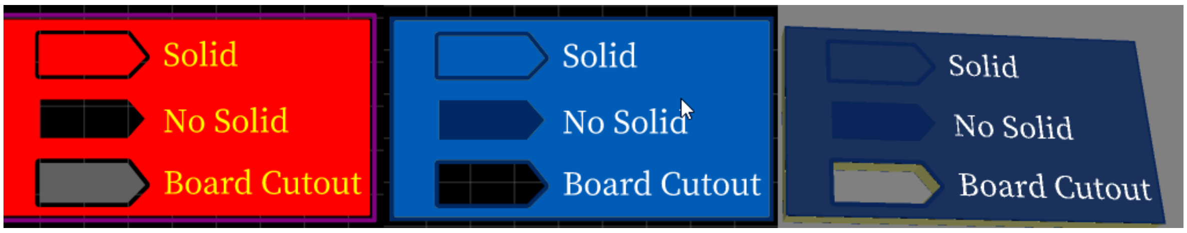
The more information please refer at [PCB: Copper Pour]()

Solid Region

EasyEDA has added a new tool Solid Region for PCB design



This is a very useful, quick way to connect Pads. You can draw a Solid Region to include all of these pads with same net name, then set the region to the same net name as the pads. It is like Copper Area but easier to use for small areas. To use Solid Region like this, set the Type attribute (in the right hand Properties panel) to Solid.

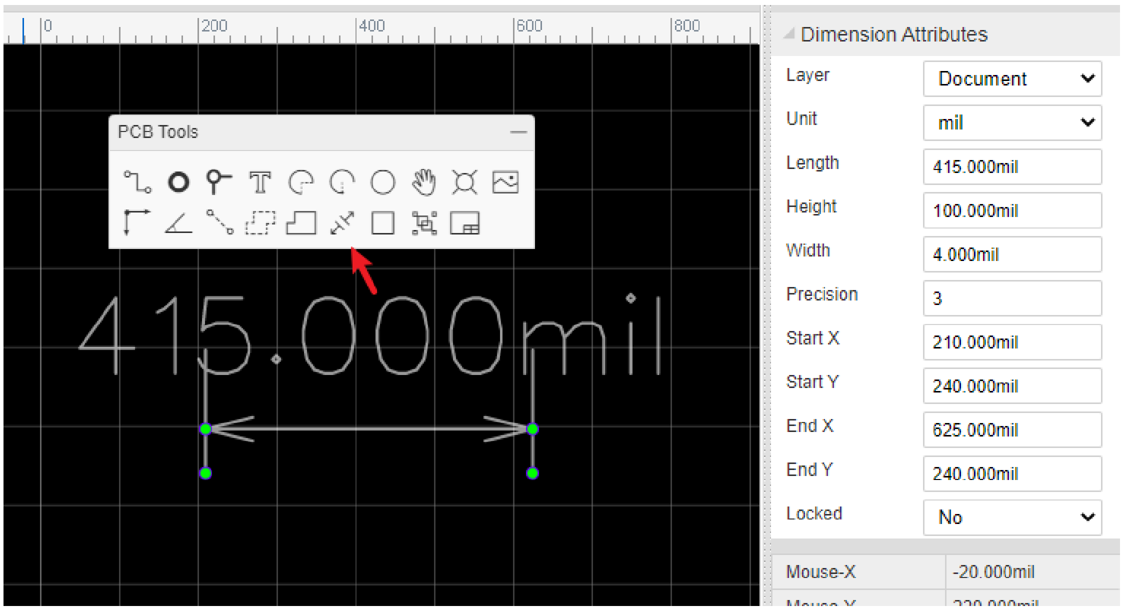


The more information please refer at [PCB: Solid Region]()

Measure/Dimension

Making and adding measurements is useful in PCB design. EasyEDA provides two methods to do this.

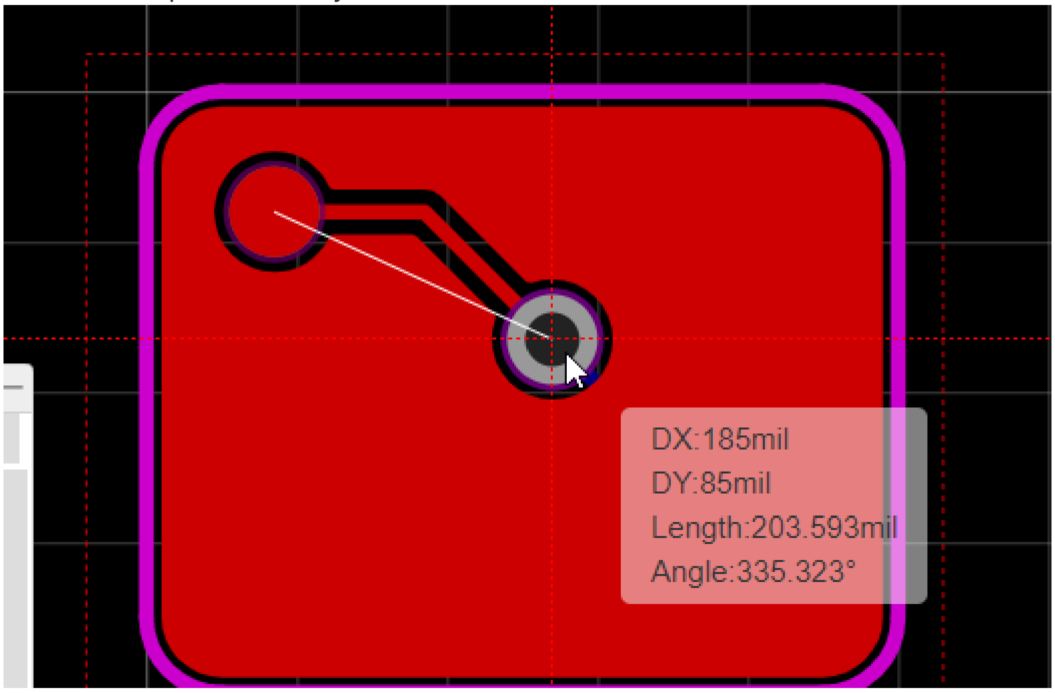
1. Dimension tool in the PCB Tools palette:
   This tool can show three units on the canvas, milliliter, inch and millimeter.



Layer	Document
Unit	mil
Length	415.000mil
Height	100.000mil
Width	4.000mil
Precision	3
Start X	210.000mil
Start Y	240.000mil
End X	625.000mil
End Y	240.000mil
Locked	No
Mouse-X	-20.000mil

   When you click one side of the dimension on the PCB, you can drag it for any directions or change its length.

2. Measure a distance using Hotkey **M**, Or Via: **Top Menu** > **Edit** > **Measure Distance**, then click the two points which you would like to measure.



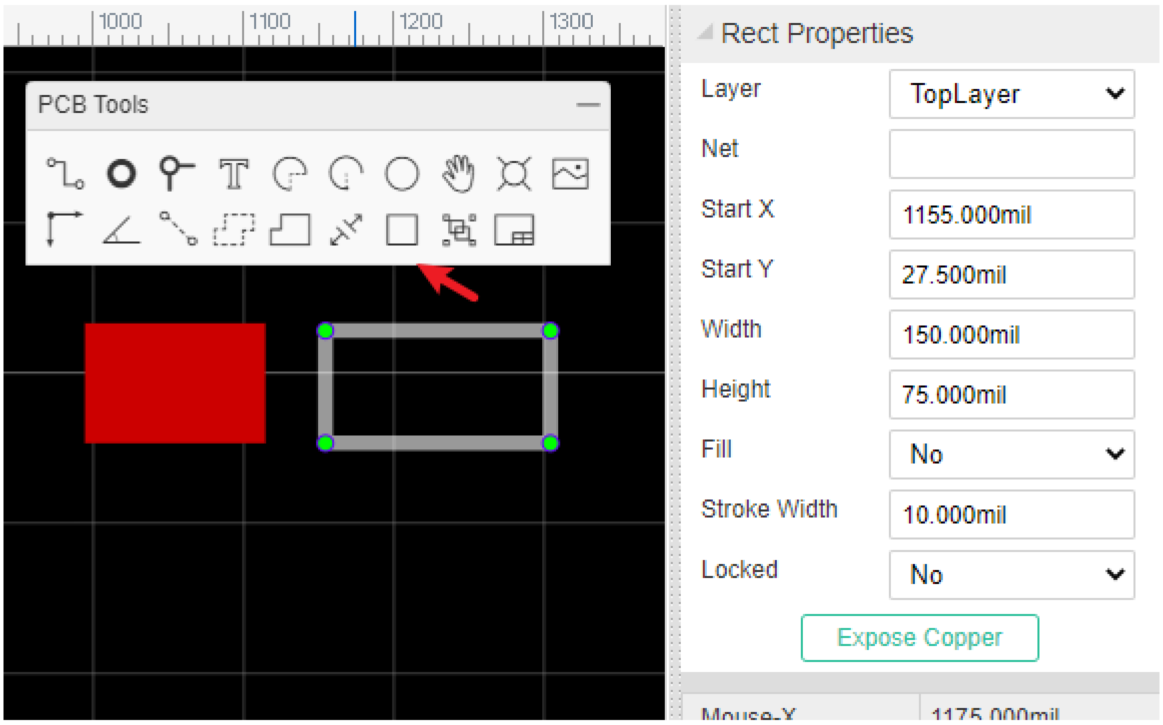
DX:185mil
DY:85mil
Length:203.593mil
Angle:335.323°

Tips:

- *It's unit follows canvas's units.*
- *You can disable the snap option to measure at the canvas property panel.*

Rect

It looks like a Solid Region, but it can't be set Nets and you can't set the Layer as NTPH.



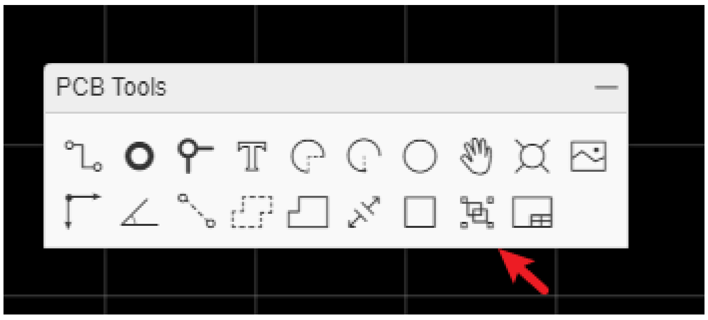
Rect Properties

Layer	TopLayer
Net	
Start X	1155.000mil
Start Y	27.500mil
Width	150.000mil
Height	75.000mil
Fill	No
Stroke Width	10.000mil
Locked	No

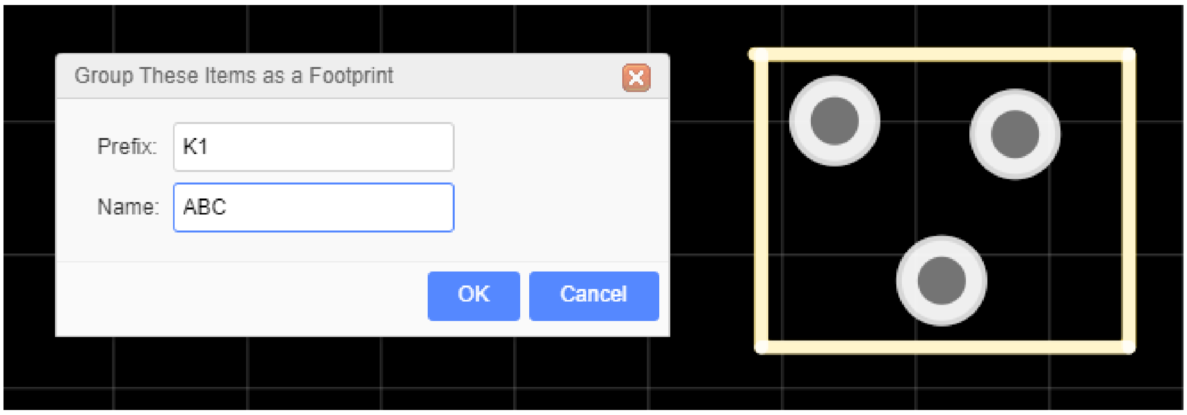
The rect doesn't rotate, you can change its width and height.

Group/Ungroup

Just like Group/Ungroup in the Schematic Editor can be used to create a schematic lib symbol, you can use Group/Ungroup from the PCB Tools palette to create a Footprint footprint in the PCB editor.



For example, place Tracks and Pads on the canvas, then select all of them and click **Group/Ungroup** to group them like as a footprint in the image below:


Group These Items as a Footprint

Prefix: K1

Name: ABC

Notice：

- Before ungroup the footprint, please change it's layer to top layer first, because of the footprint after grouping will at top layer.

- The grouped footprint doesn't support Import Changes, it will be removed if you Import Changes.

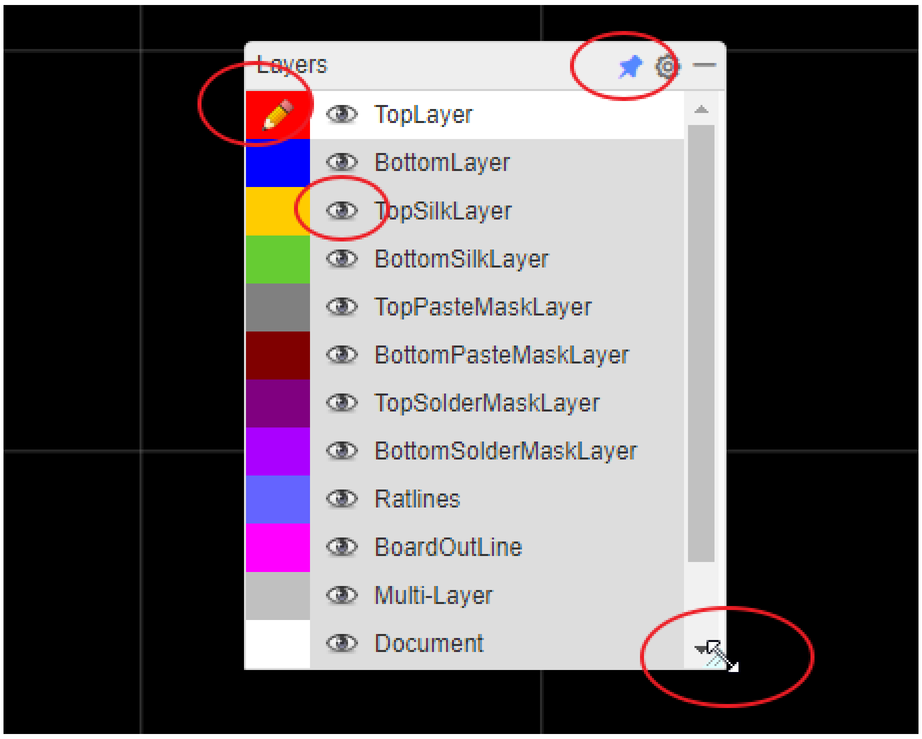
Layers Tool and Objects

Layers Tool

Active Layer: The colours of the layers in the **Layers Tool** are defined in the Layer Options Settings. To work on a layer then you must make it the Active layer.

To do this,

- Click on the eye icons to show/hide layers.
- The pencil icon in the coloured rectangle indicates that this is the active layer.
- Click the pin icon to fix the layering tool without automatically closing it.
- The height and width of the layer tool can be adjusted when dragging the lower right corner of the Layers Tool.



HotKeys for layer activation:

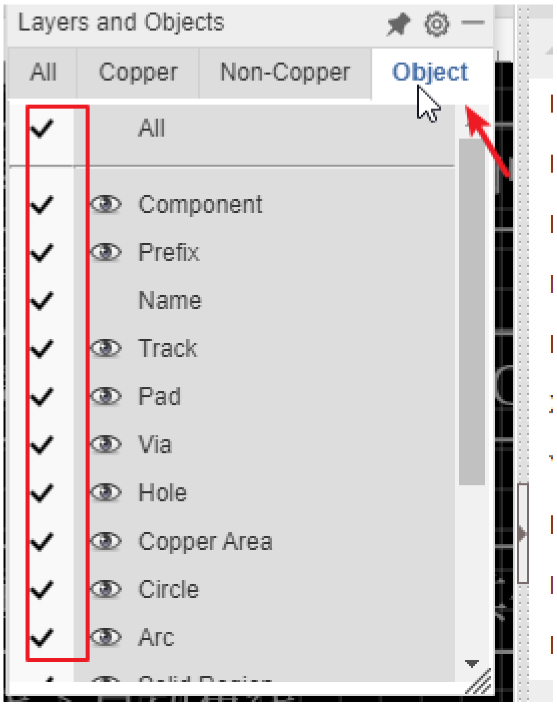
- **T**: Top Layer is active
- **B**: Bottom Layer
- **1**: Inner1 Layer
- **2**: Inner2 Layer
- **3**: Inner3 Layer
- **4**: Inner4 Layer

The more information for the PCB layers please refer at PCB Layout - Layer Manager

note: the hidden PCB layer is only visually hidden. The corresponding layer will still be exported during photo preview, 3D preview and Gerber export.

Objects Filter Tool

Click "Object" to switch to object filtering.



Select： When the tick in front of the object is checked, the corresponding object in the canvas can be manipulated with the mouse. Uncheck will not allow mouse operation. Including click selection, box selection, drag and other operations.

Eye： Click eyes to modify the display and hiding of corresponding objects in batches.

- Component：Displays or hides the entire components, excluding the component's name and prefix
- Prefix：Displays or hides the entire components' prefix
- Name：Displays or hides the entire components' name
- Track：Displays or hides the entire tracks, for all layers
- Pad：Displays or hides the entire free pads, excluding the pads in the component
- Copper Area：Displays or hides the entire copper areas' fill area, excluding copper outline
- Text：Displays or hides the entire normal texts, excluding the text of the component

Note:

- The layer and object invisible and visible will not go into Undo and Redo.

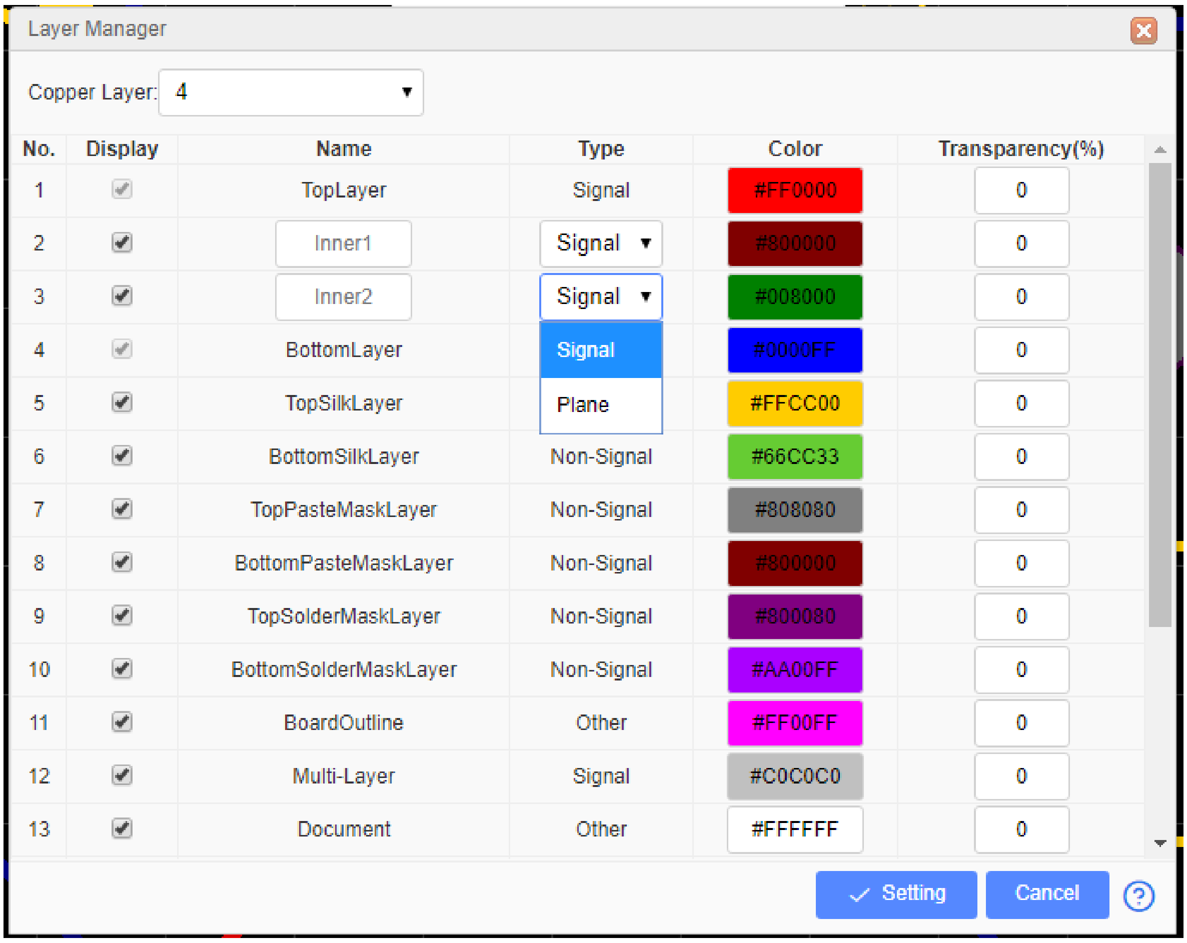
Layer Manager

Layer Manager

You can set the PCB layer's parameters at the Layer Manager.

Via **Top Menu> Tools > Layer Manager...**, Or Click **Layers Tool** gear icon. Or right-click the canvas - Layer Manager menu.

The Layer Manager dialog:


Layer Manager

Copper Layer: 4

No.	Display	Name	Type	Color	Transparency(%)
1	✓	TopLayer	Signal	#FF0000	0
2	✓	Inner1	Signal	#800000	0
3	✓	Inner2	Signal	#008000	0
4	✓	BottomLayer	Signal	#0000FF	0
5	✓	TopSilkLayer	Plane	#FFCC00	0
6	✓	BottomSilkLayer	Non-Signal	#66CC33	0
7	✓	TopPasteMaskLayer	Non-Signal	#808080	0
8	✓	BottomPasteMaskLayer	Non-Signal	#800000	0
9	✓	TopSolderMaskLayer	Non-Signal	#800080	0
10	✓	BottomSolderMaskLayer	Non-Signal	#AA00FF	0
11	✓	BoardOutline	Other	#FF00FF	0
12	✓	Multi-Layer	Signal	#C0C0C0	0
13	✓	Document	Other	#FFFFFF	0

The Layer Manager setting only works for the current editing PCB.

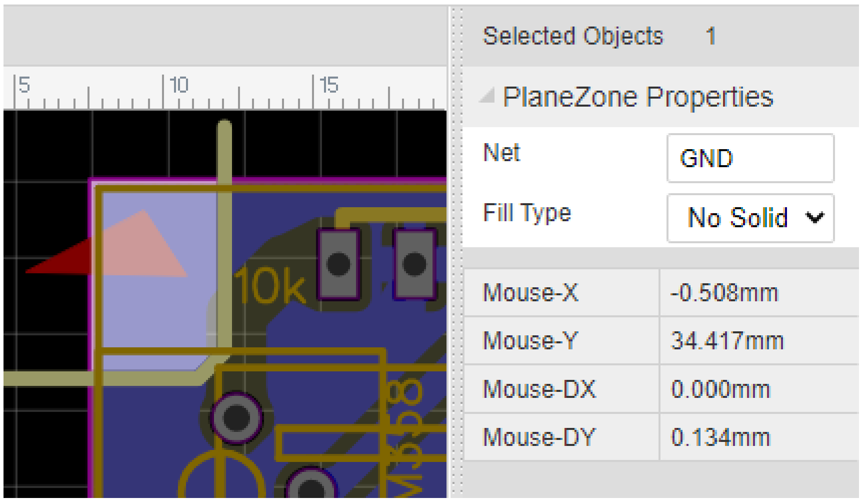
Copper Layer: The copper layer of your PCB. EasyEDA support 34 copper layers. The more copper layers the PCB will be more expensive. The TopLayer and BottomLayer is default layer, can not be disable. If you want change the copper layers from 4 to 2, you must delete the inner layers objects first.

Display: If you don't want a layer dosen't display at "Layers Tool", you can disable the checkbox. Notice: This option only hide the layer name on the "Layers Tool", the objects of the hidden layer still exist, when you generating the Gerber, they will appear.

Name: Layers name. For the inner layer, you can define the name.

Type:

- **Signal**: Which is working for the signal. Such as Top and bottom layer.
- **Plane**: When the inner layer type is "Plane", this layer will be copper pourred, if you want to separate the copper area you can draw the Track or Arc. You can treat this layer is a only has the copper area, but its easy than draw the copper area. The track you routed will generate the clearance when generating the Gerber. The "Plane" usually is using for the Power or Ground copper pour on the inner layer. You can set the net for the plane zone.


Selected Objects 1

PlaneZone Properties

Net	GND
Fill Type	No Solid
Mouse-X	-0.508mm
Mouse-Y	34.417mm
Mouse-DX	0.000mm
Mouse-DY	0.134mm

Notice:
When draw the track to separate the plane zone, the track start ponit and end point must over the middle line of the board oultine track. Otherwise, the plane zone will not be separated; When using the plane layer, the PCB can not exist two closed borad outline, only one closed board outline will generate the plane zone.

- **Non-Signal**: Such as silk screen, mechanical layer, document layer etc.

Color: You can define the color for each layer.

Transparency: You can change the layer transparency.

Layer Definination:

- **TopLayer/BottomLayer**: The top side and bottom side of the PCB board, copper layer.
- **InnerLayer**: Copper layer, routing track and copper pour.
- **TopSilkLayer/BottomSilkLayer**: Board silkscreen.
- **TopPasteMaskLayer/BottomPasteMaskLayer**: This layer is the layer used to make the stencil for the SMT pads, helping to solder. This layer has no effect on production if the board is not required to make the stencil.
- **TopSolderMaskLayer/BottomSolderMaskLayer**: The top and bottom cover layers of the board are typically green oil, which acts to prevent unwanted welding. This layer belongs to the negative film drawing mode. When you have wires or areas that do not need to cover green oil, draw them at the corresponding positions. After the PCB is generated, these areas will not be covered with green oil, which is convenient for operations such as tinning.
- **BoardOutline**: The board shape definition layer. To define the actual size of the board, the board factory will produce the board according to this shape.
- **TopAssemblyLayer/BottomAssemblyLayer**: Simplified outline of components for product assembly and repair. Used to export document printing, without affecting PCB production.
- **MechanicalLayer**: Record the information on the mechanical layer in the PCB design, and only use it for information recording. By default, the shape of the layer is not manufactured at the time of production. Some board manufacturers use the mechanical layer to make the frame when using Altium file to production. When using Gerber file, it is only used for text identification in JLCPCB. For example: process parameters; V cut path etc. In EasyEDA, this layer does not affect the shape of the border of the board. If the mechanical layer has closed wires, JLCPCB will give priority to using the mechanical layer as the shape of the board when producing the board. If there is no outer frame of the mechanical layer, GKO will be used as the frame (historical influence of Altium file). It is necessary to pay attention to the use of the mechanical layer in the design.

- **DocumentLayer**: Similar to the mechanical layer. But this layer is only visible in the editor and will not be generated in the Gerber file.
- **RatlineLayer**: PCB network ratline display, this layer is not in the physical sense, in order to facilitate the use and set color, it is placed in the layer manager for configuration.
- **HoleLayer**: Similar to the RatlineLayer. For Hole(Non-Plated Hole) display.
- **Multi-Layer**: Similar to the RatlineLayer. For multi-layer hole(Plated hole) display. If the PAD setting layer property as mult-layer, it will connect with all copper layers.
- **DRCErrorLayer** Similar to the RatlineLayer. For DRC(Design Rule Error) marking dispaly.

Layout Single Layer PCB

The PCB copper layers of EasyEDA are double, EasyEDA doesn't support layout a signle layer directly. if you want to layout a single layer PCB(such as only layout on the bottom layer),

There are two methods:

Method 1:

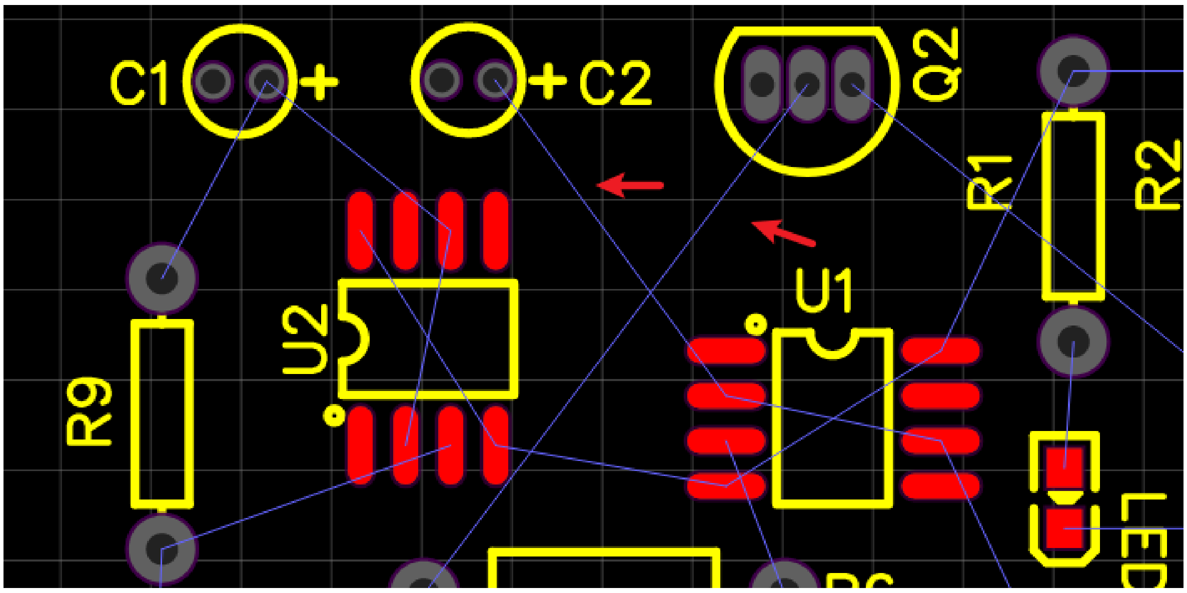
- Route the track and copper on the bottom layer, and without placing via.
- If you are using the footprints which have the multi-layer pads, that will appear on the top and bottom layer, then you need to change all multi-layer pads "Plated" as "No".
- Generate the Gerber, decompress the Gerber zip file, delete the layers which you don't need(such as Gerber_TopLayer.GTL, Gerber_TopSilkLayer.GTO, Gerber_TopSolderMaskLayer.GTS, Gerber_TopPasteMaskLayer.GTP).
- And re-compress the Gerber to a zip file, and order it.

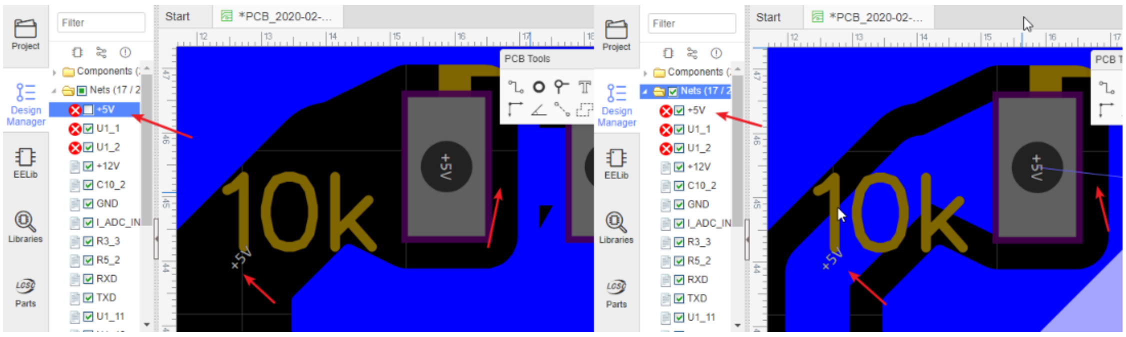
Method 2:

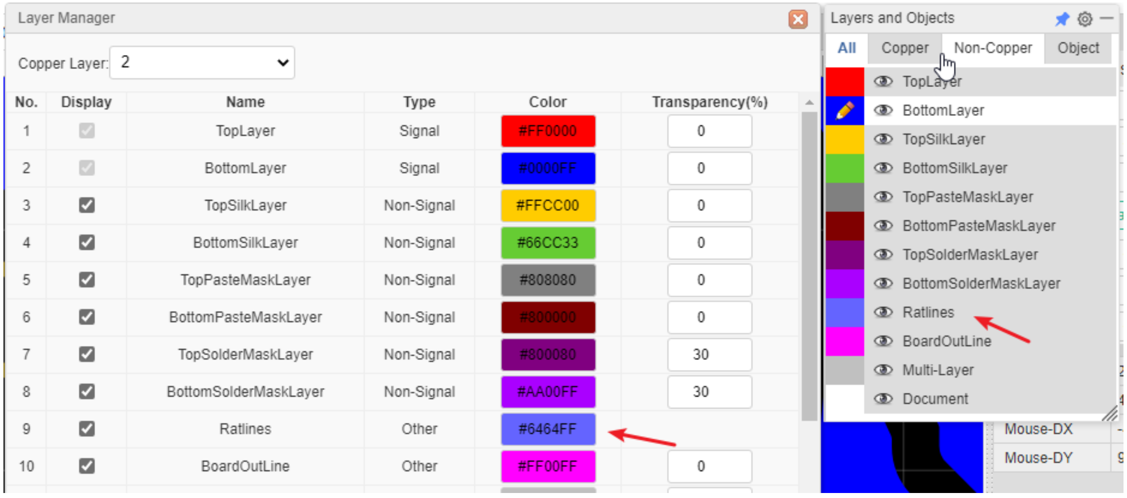
- Design your PCB at one side, if other side has pads etc, you don't need to deal with them.
- Generate the Gerber.
- Add the comment for mention that you need to order the signle layer PCB when order the PCB.

Ratline

When you layout the track in the PCB, Between Pad and Pad as they have the same net name, a Ratline will be automatically shown among them to reveal that they can be connected with a track.



1. If you want one ratline do not show on the PCB editor, you can deselect the net in the design manager, as below deselect `+5V`:
   If you still draw a track in `+5V` after deselecting, canvas will not display this track and ratline , but it will show a net text with `+5V` as below.
   

   Based on this skill， you don't need to lay GND net before copper area in the PCB.
2. If you want to check the ratlines with highlight, you can click the pencil on the Ratlines Layer as below, and you can change the ratline's color at Layer Manager.
   
   Layer Manager

   Copper Layer: 2

No.	Display	Name	Type	Color	Transparency(%)
1	✓	TopLayer	Signal	#FF0000	0
2	✓	BottomLayer	Signal	#0000FF	0
3	✓	TopSilkLayer	Non-Signal	#FFCC00	0
4	✓	BottomSilkLayer	Non-Signal	#66CC33	0
5	✓	TopPasteMaskLayer	Non-Signal	#808080	0
6	✓	BottomPasteMaskLayer	Non-Signal	#800000	0
7	✓	TopSolderMaskLayer	Non-Signal	#800080	30
8	✓	BottomSolderMaskLayer	Non-Signal	#AA00FF	30
9	✓	Ratlines	Other	#6464FF	
10	✓	BoardOutLine	Other	#FF00FF	0

   Layers and Objects

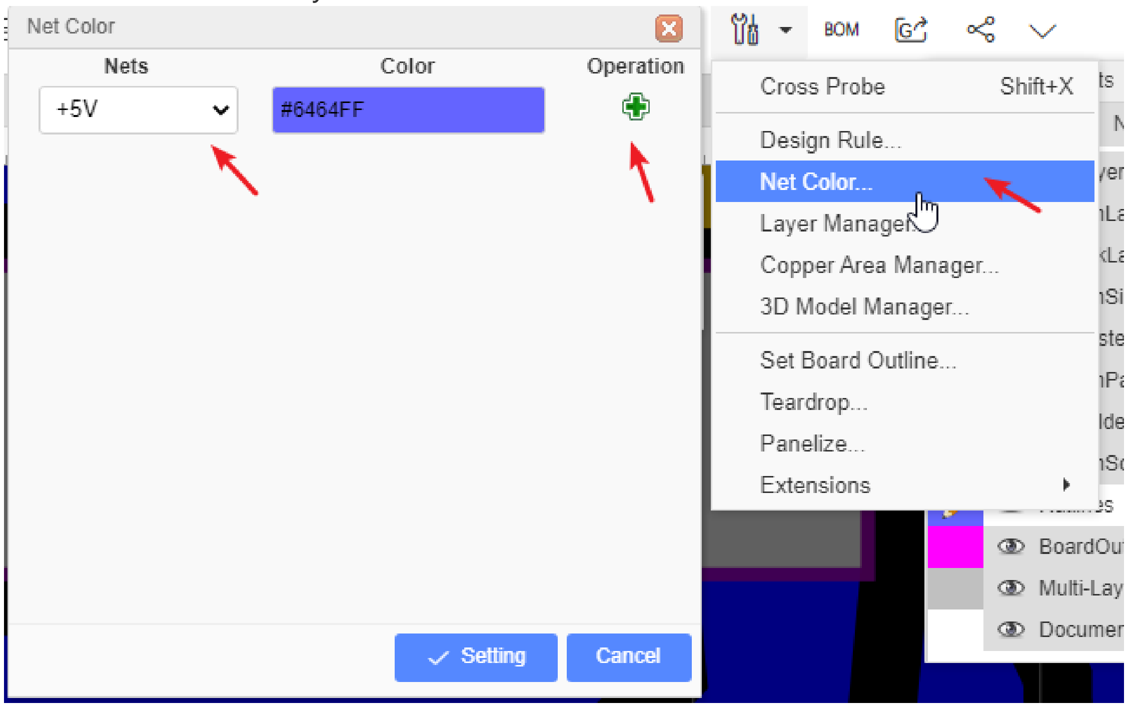
   All | Copper | Non-Copper | Object

   TopLayer
   BottomLayer
   TopSilkLayer
   BottomSilkLayer
   TopPasteMaskLayer
   BottomPasteMaskLayer
   TopSolderMaskLayer
   BottomSolderMaskLayer
   Ratlines
   BoardOutLine
   Multi-Layer
   Document

   Mouse-DX
   Mouse-DY
3. If you want to hightlight one ratline all the time, you can click a pad, press hotkey H, press it again unhighlight.

4. If you want to change one ratline's color, you can set it at: - Tools - Net Color. After setting the color, you need to click the plus icon on the right. The color is not affected by

the color of the ratline layer.



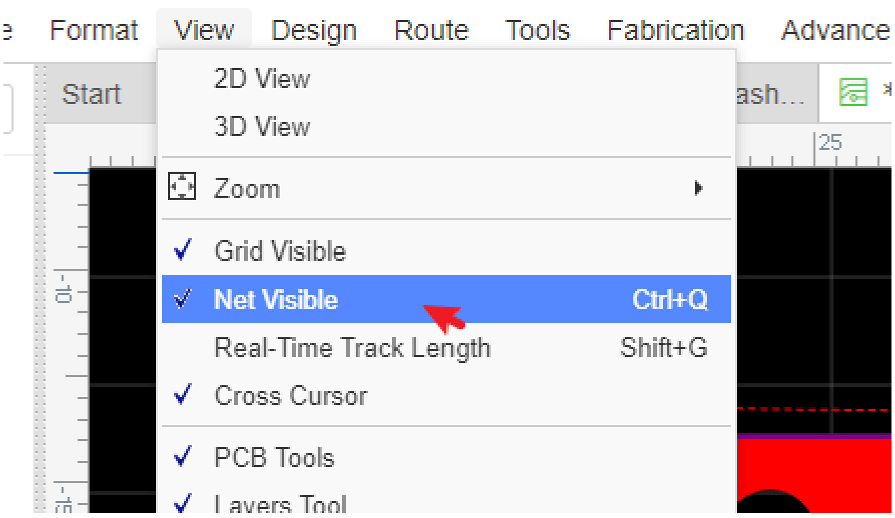
5. If you want to remove one ratline, you just need to remove objects' net. Select it and empty the net.

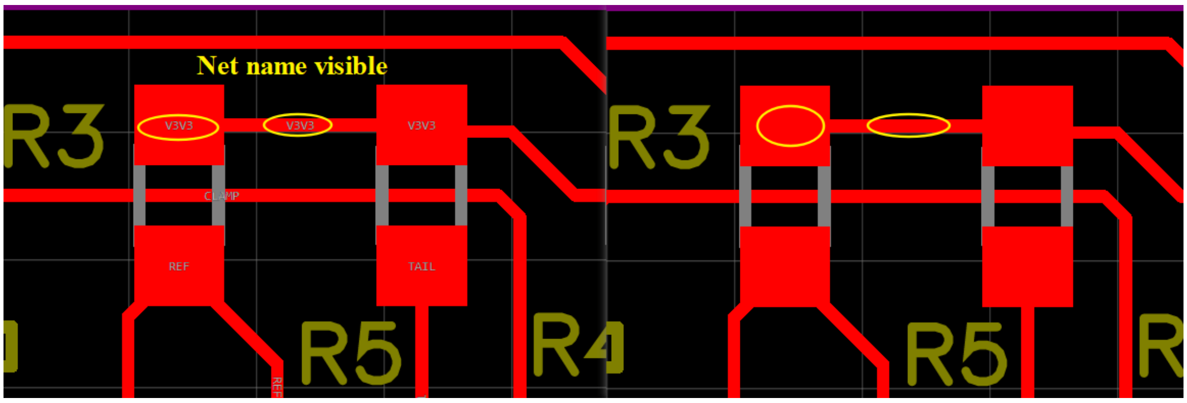
PCB Net

Net Name Visible

PCB editor can display net name in the track or Pads, if you don't need this feature, just need to turn it off via:

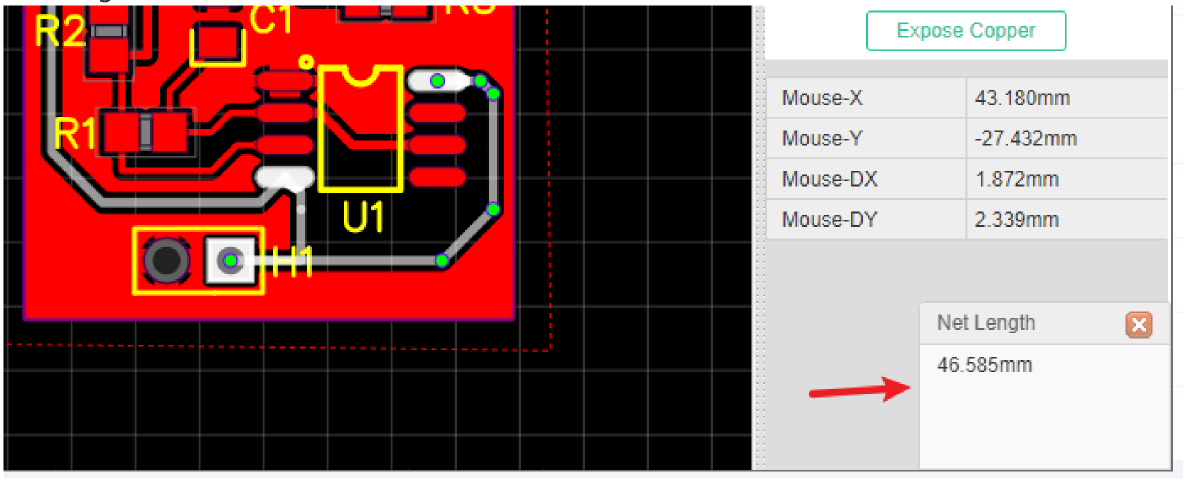
Top Menu > View > PCB Net Visible, or press hotkey `CTR+Q` .





Net Length

After selecting a track, and then pressing `H` key or click its net at Design Manager, EasyEDA will highlight the whole net and pop a message box to tell you the whole net's length. like in the image below

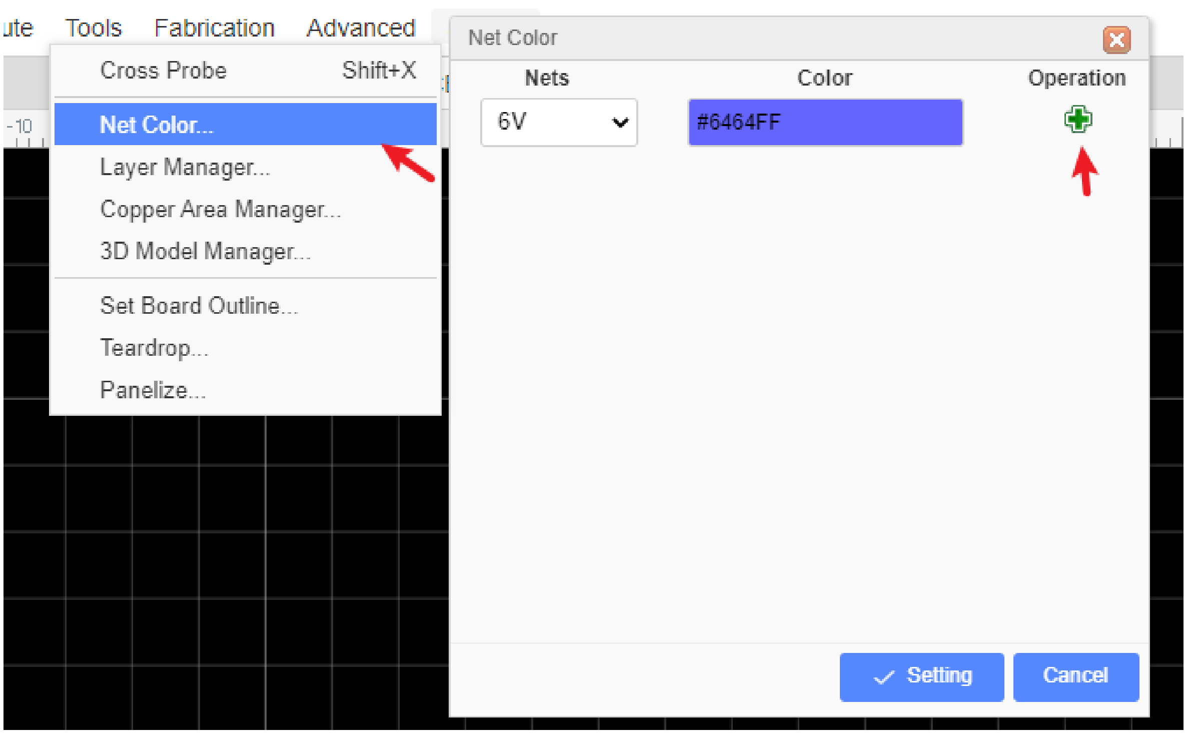


Mouse-X	43.180mm
Mouse-Y	-27.432mm
Mouse-DX	1.872mm
Mouse-DY	2.339mm

Net Length
46.585mm

Net Color

If you want to change one Ratline's or Net's color, you can set it at: **Top Menu- Tools - Net Color**. After setting the color, you need to click the plus icon on the right. The color is not affected by the color of the ratline layer.

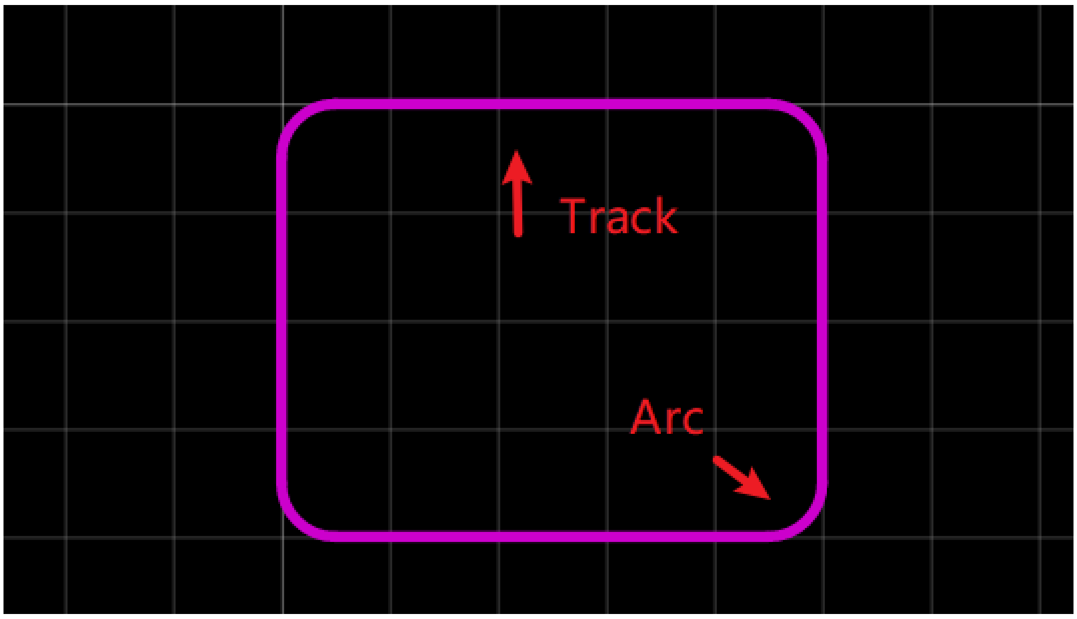


Nets	Color	Operation
6V	#6464FF	✚

When you set a color for a net, you need to click the + button to make it works.

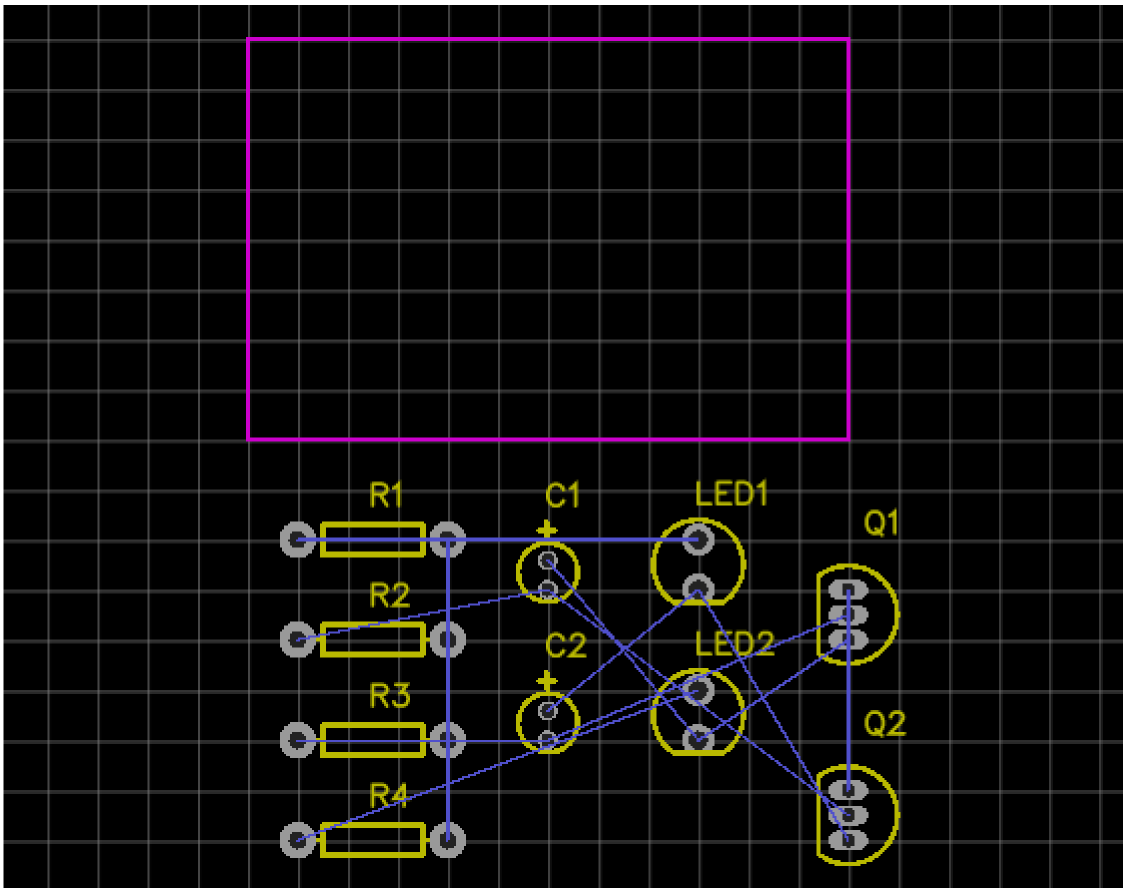
Board Outline

Before placing footprints we need to create a board outline. The board outline must be drawn on the **Board OutLine** layer. So first, set **Board OutLine** as the active layer, then draw the board outline using **Track** and **Arc** from the PCB Tools palette.



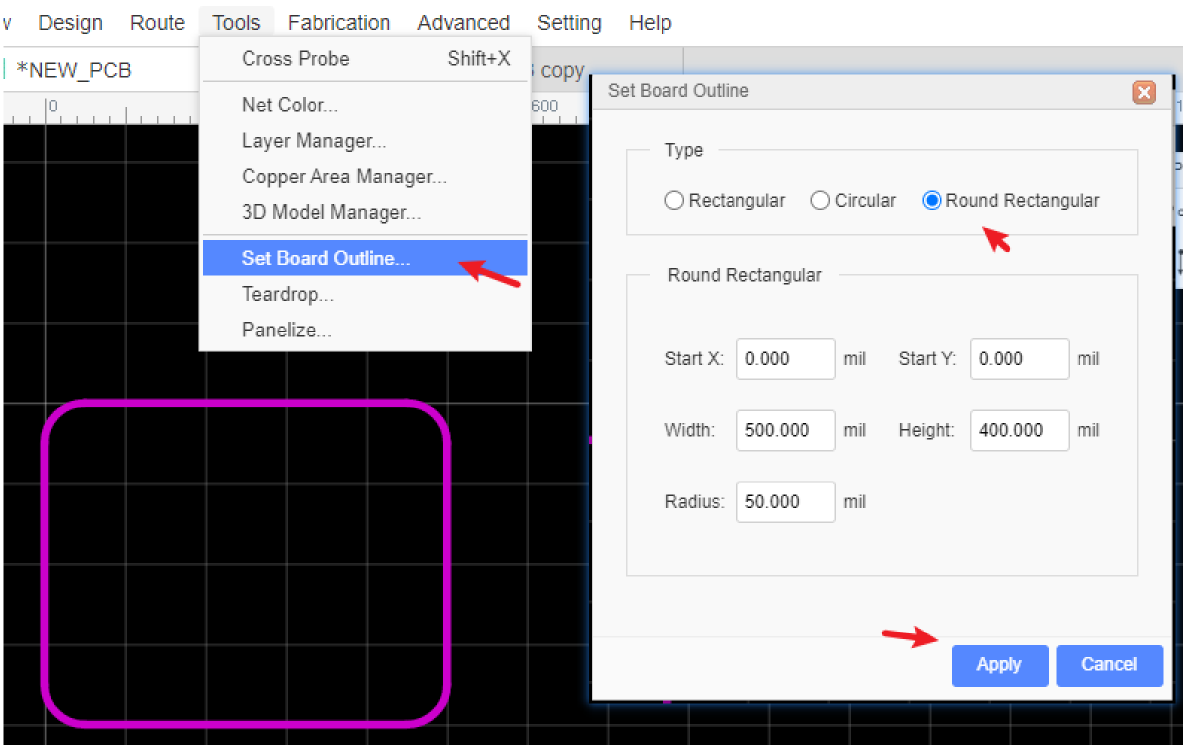
When converting a Schematic to PCB, EasyEDA will try to create a board outline for you.

The area of the default board outline area is 1.5 times the sum of the area of all of your footprints, so you can place all of your footprints into this board outline with some allowance for tracking. If you do not like the board outline, you can remove the elements it is made up from and draw your own.



To create a simple rectangular board outline, this arc can be removed and the line X and Y end points edited - either directly in the Properties panel or by dragging the line ends - to close the rectangle.

And EasyEDA provides a **Board outline wizard**, so it is very easy to create a board outline. Via: **Top Menu > Tools > Set Board Outline**, Or find it on the toolbar.



Set Board Outline

Type: Rectangular, Circular, [✓ Round Rectangular]

Round Rectangular

Start X: 0.000 mil Start Y: 0.000 mil

Width: 500.000 mil Height: 400.000 mil

Radius: 50.000 mil

In this dialog, there's a choice of 3 types of board outlines, Rectangular , Circular, Round Rect. If you need a different more complex board outline, you need to import a DXF file.

Notice:

- When generate the Gerber, EasyEDA will show error if the board outline doesn't closed or the board outline tracks overlap.
- You can cutout the hole by using the board outline, or use `Hole`, or `Solid Region(Type: Board Cutout)` to create the hole instead of using the board outline.
- You can right-click track or circle to convert to board coutout.
- If the board outline doesn't closed, the copper pour will not show up.

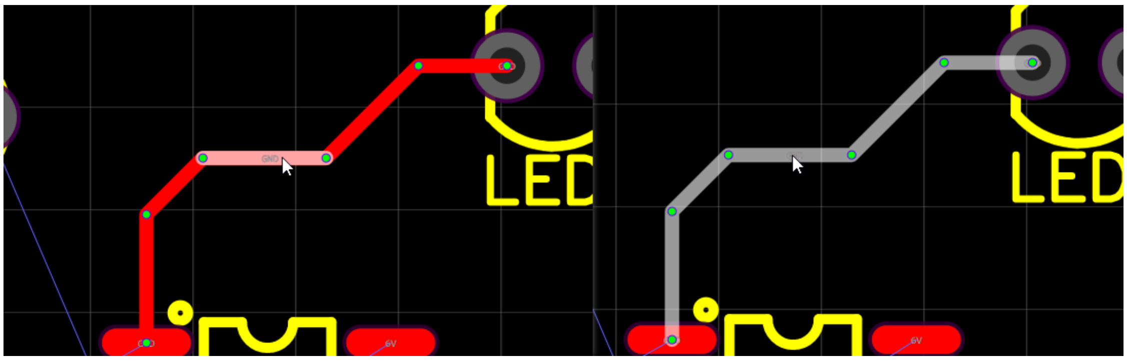
Route Tracks

Route Tracks

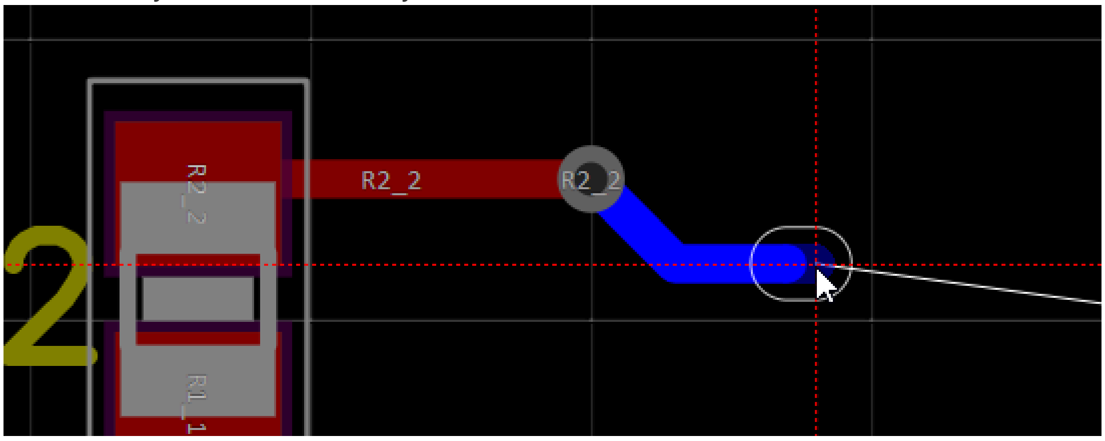
In the schematic editor, we use Wire or the `W` Hotkey to connect Pins, in a similar way in the PCB editor, we use Track to connect Pads. Track allows you to draw PCB tracks and can be found on the PCB Tools palette or using the `W` Hotkey (not T).

Some Tips about Track

- Single click to start drawing a track. Single click again to pin the track to the canvas and continue on from that point. Right click to end a track. Double right-click to exit track mode.

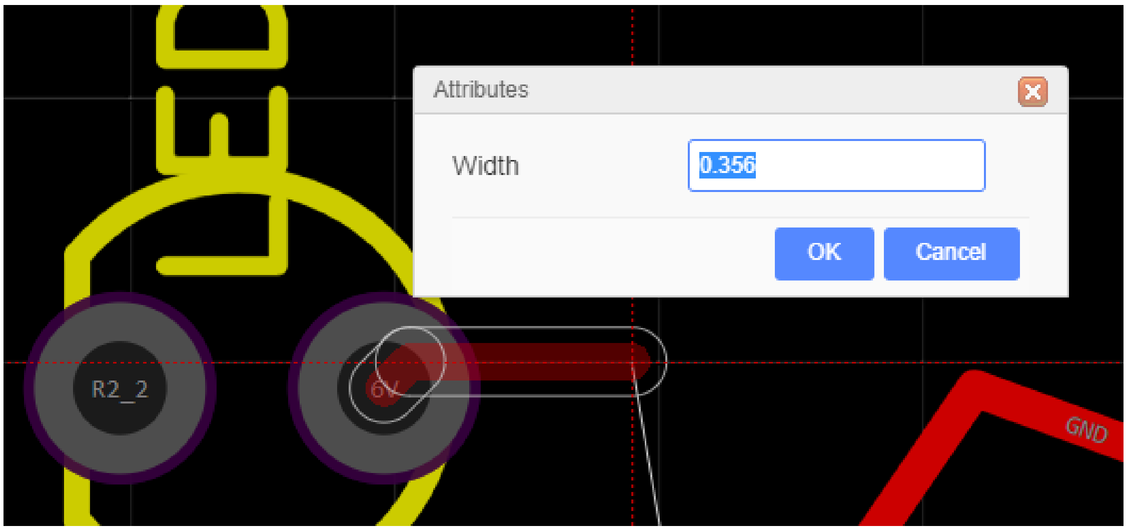


- Drawing a track at the same time as using a hotkey(for example hotkey `B`) for changing the active layer will automatically insert a Via:

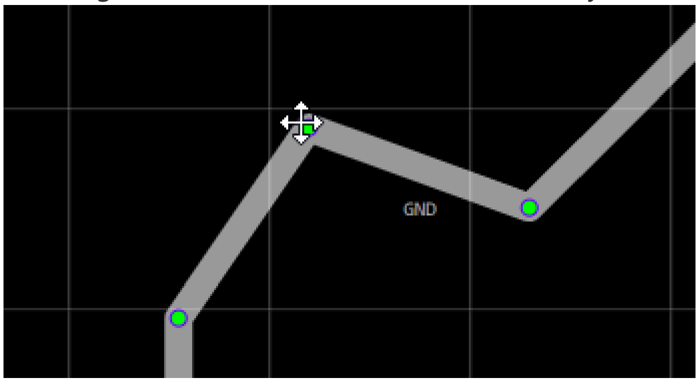


If you start drawing a track on the top layer, you will see it drawn in red, then press the B key to change to bottom layer and you will see EasyEDA insert a grey via and then the track will continue being drawn but now on the bottom layer in blue.

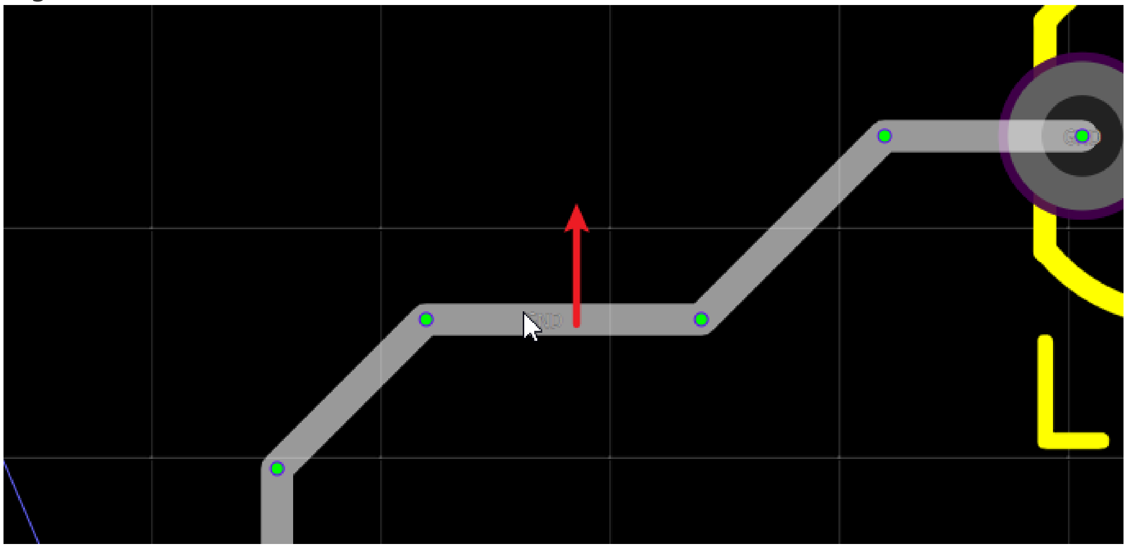
- Pressing the `+` or `-` Hotkeys when drawing the track will change the width of the track on the fly. Use the hotkey `TAB` to change the track width.



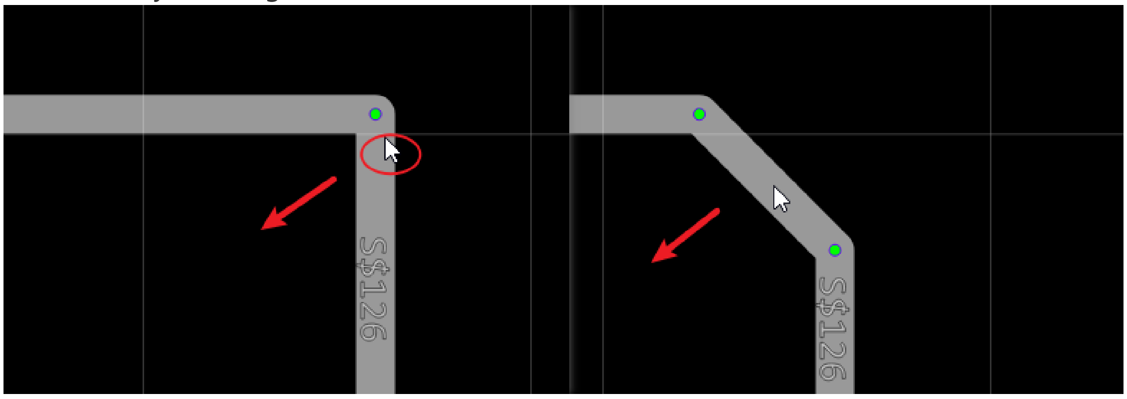
- Double clicking on a drawn section of the track will add a new vertex at that point. You can drag the vertex to form a new corner. And you can right-click the point and delete it.



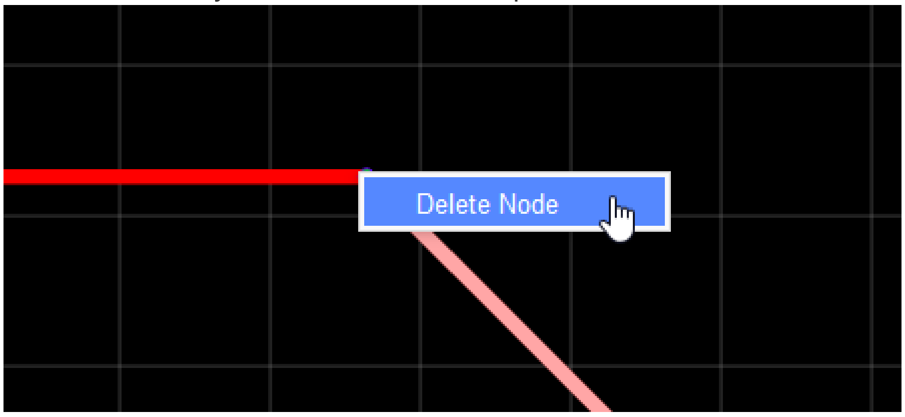
- Click to select the track and then Click and Drag on a segment of the track to adjust the segment between vertices.



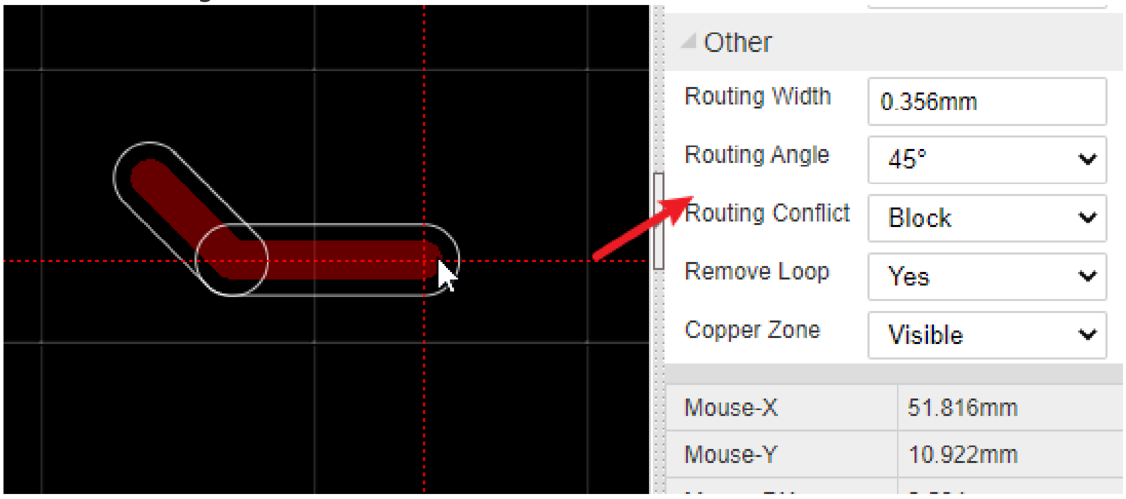
- When the track's corner is a right Angle and the routing corner is 45, drag the track next to the right Angle node to make a bevel. Instead of dragging a node directly, dragging a node directly will drag the entire track.



- Selected a track, you can delete its node point.



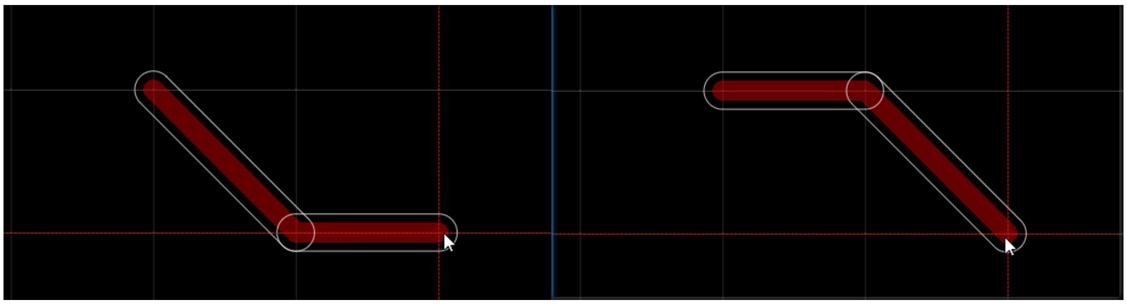
- Pressing the `L` Hotkey when drawing the track will change the track's Route Angle on the fly. And you can change Route Angle on the Canvas Attributes of the right panel before the next drawing.



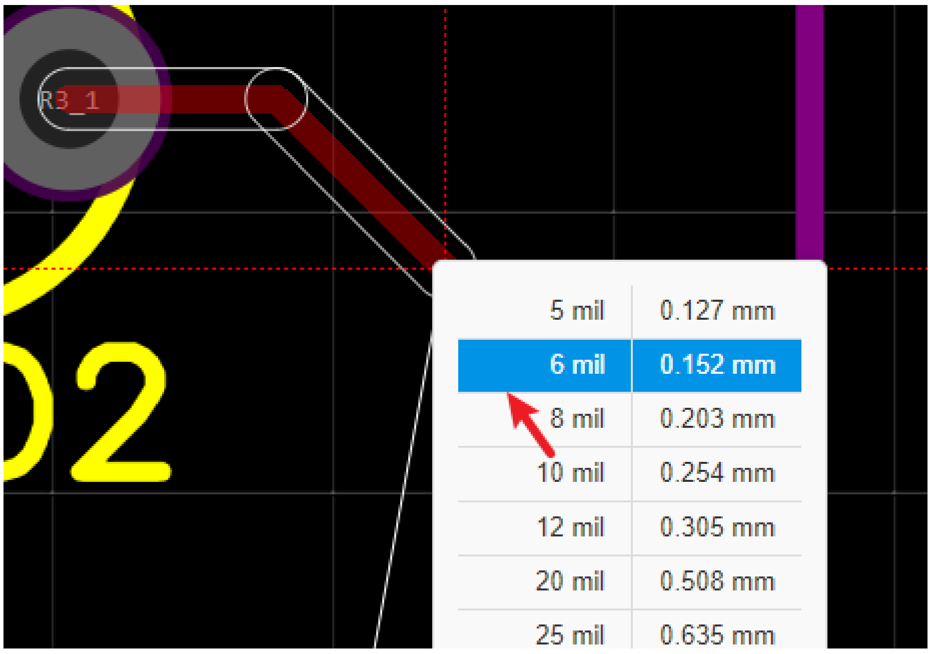
Other

Routing Width	0.356mm
Routing Angle	45°
Routing Conflict	Block
Remove Loop	Yes
Copper Zone	Visible
Mouse-X	51.816mm
Mouse-Y	10.922mm

- You can change inflection direction when routing, just press `space` key.

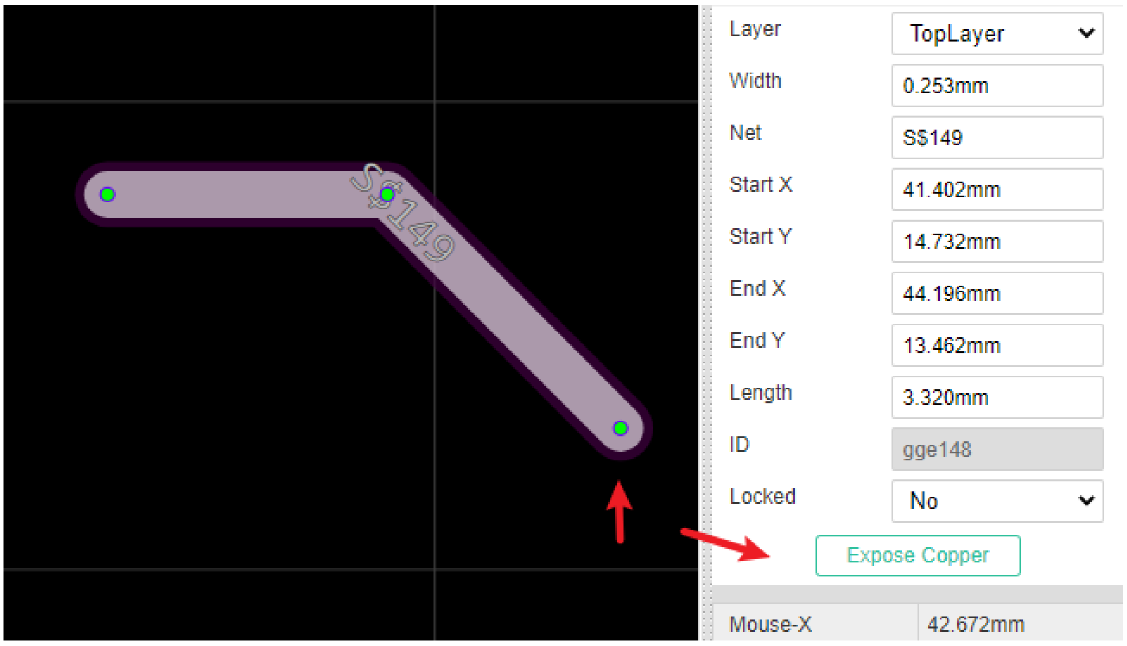


- If you want to route a track and use "L", and the then press "+", you will get two different size track segment. or press "SHIFT+W".



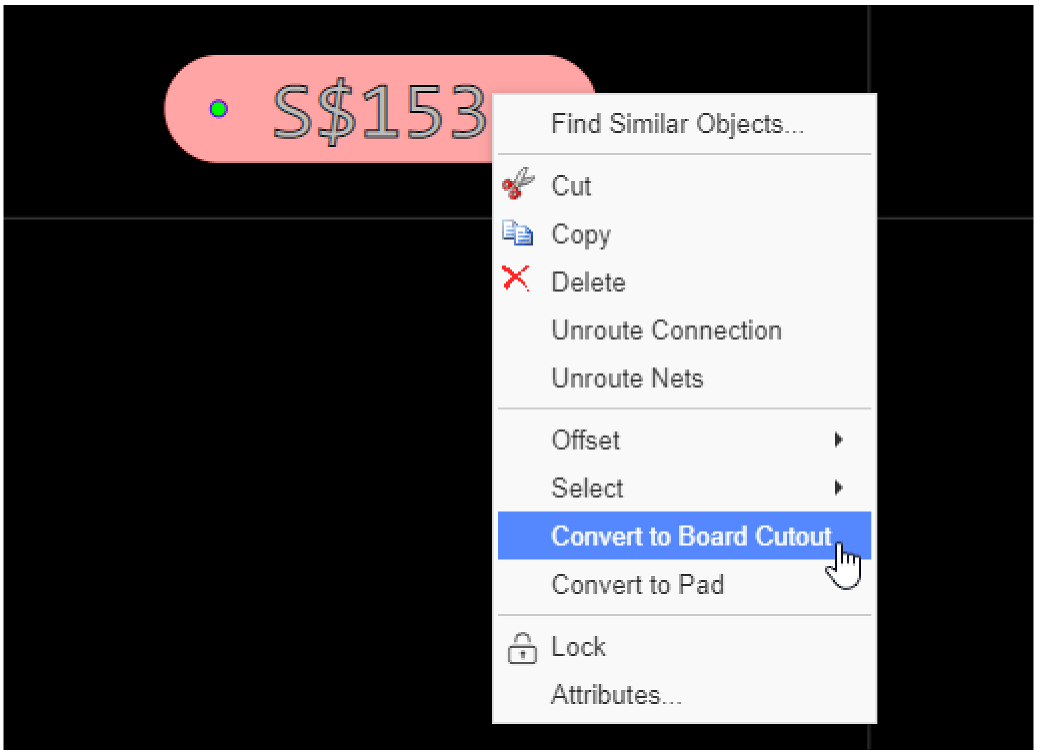
5 mil	0.127 mm
6 mil	0.152 mm
8 mil	0.203 mm
10 mil	0.254 mm
12 mil	0.305 mm
20 mil	0.508 mm
25 mil	0.635 mm

- If you want to create the solder mask aperture for the track, you can use "Expose Copper" when you select the track on the right-hand panel. The solder mask will bigger 4mil than the track.

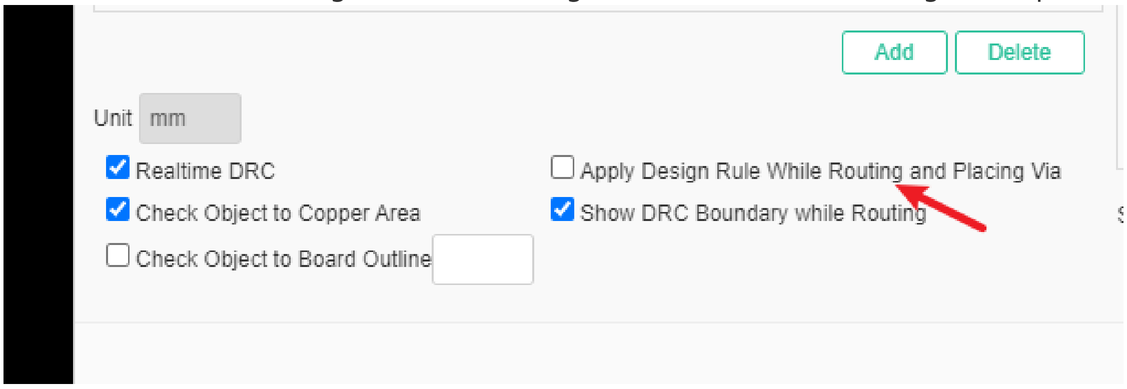


Layer	TopLayer
Width	0.253mm
Net	S$149
Start X	41.402mm
Start Y	14.732mm
End X	44.196mm
End Y	13.462mm
Length	3.320mm
ID	gge148
Locked	No
Mouse-X	42.672mm

- And if you want to create the slot hole, you can route a track, and then right-click the "Convert to Board Cutout" menu.



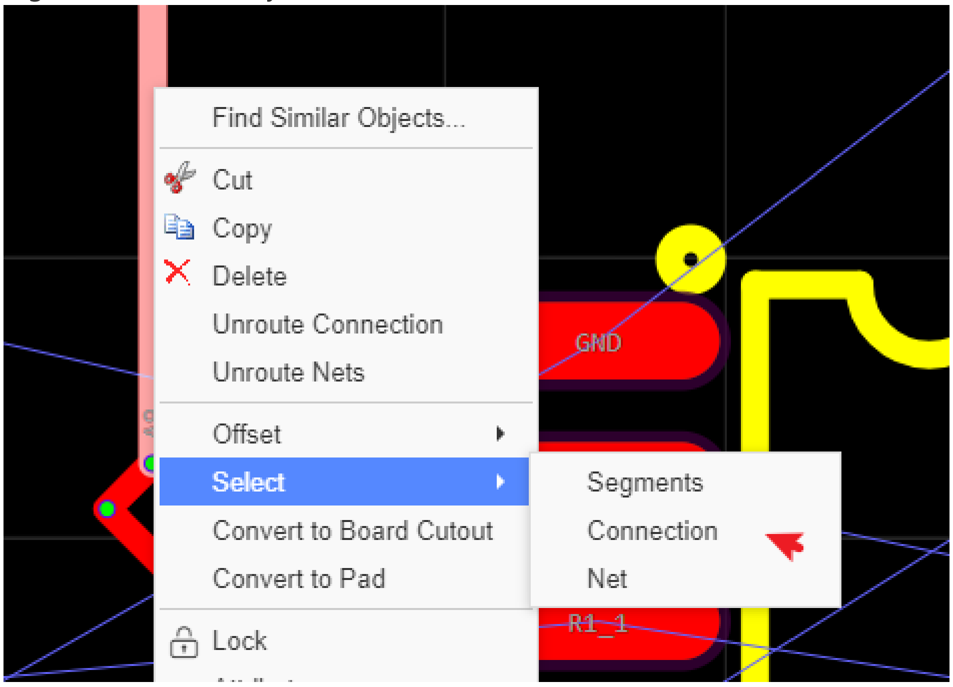
- You can make track routing width follow design rule, after enable the design rule option.



Unit mm

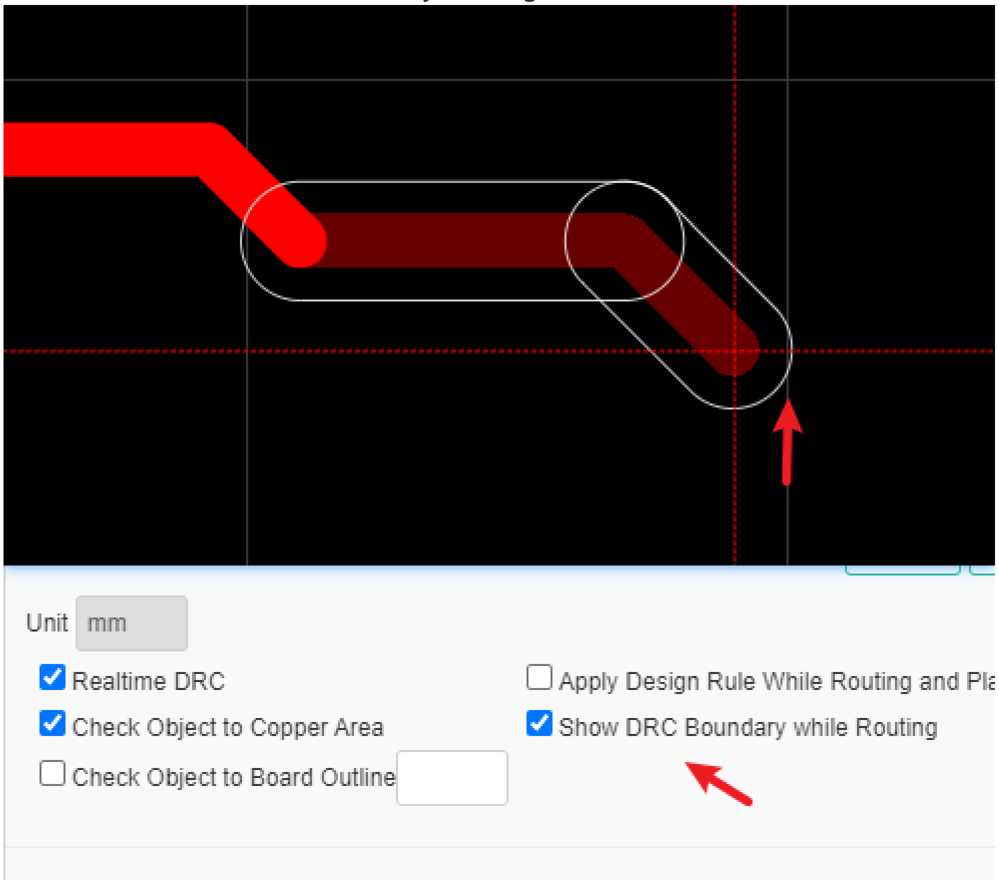
- [✓] Realtime DRC
- [✓] Check Object to Copper Area
- [ ] Check Object to Board Outline
- [ ] Apply Design Rule While Routing and Placing Via
- [✓] Show DRC Boundary while Routing

- Right-click the track, you can select the track connection or a whole same net tracks.



- If you want to the whole track, you can press `SHIFT` and move it.

- You can disable the DRC boundary at Desgin Rule. The size follow the rule.



Unit mm

[✓] Realtime DRC

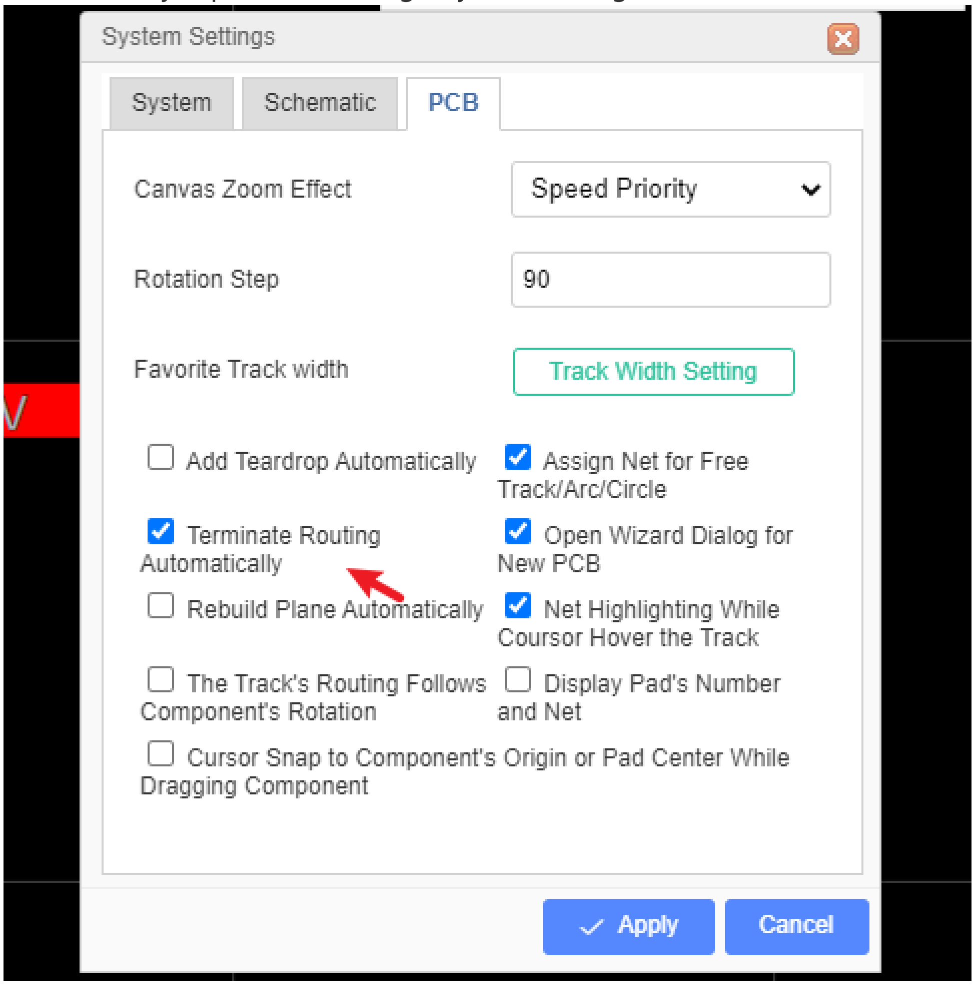
[✓] Check Object to Copper Area

[ ] Check Object to Board Outline

[ ] Apply Design Rule While Routing and Pla

[✓] Show DRC Boundary while Routing

- If you want to continue routing for a net, you can disable the "Terminate Routing Automatically" option at "Setting - System Setting - PCB".



System Settings

System | Schematic | PCB

Canvas Zoom Effect: Speed Priority

Rotation Step: 90

Favorite Track width: Track Width Setting

[ ] Add Teardrop Automatically

[✓] Assign Net for Free Track/Arc/Circle

[✓] Terminate Routing Automatically

[✓] Open Wizard Dialog for New PCB

[ ] Rebuild Plane Automatically

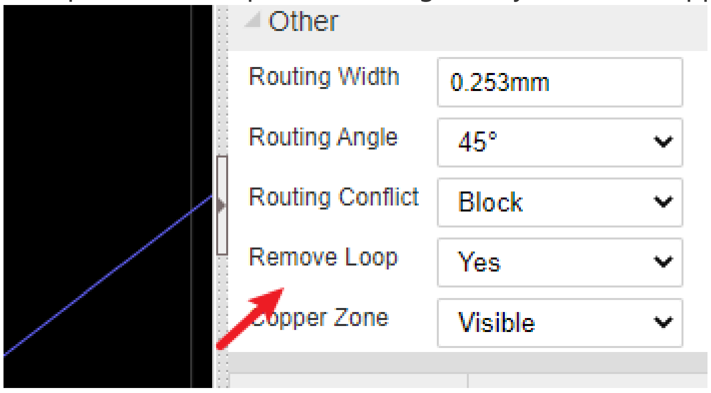
[✓] Net Highlighting While Coursor Hover the Track

[ ] The Track's Routing Follows Component's Rotation

[ ] Display Pad's Number and Net

[ ] Cursor Snap to Component's Origin or Pad Center While Dragging Component

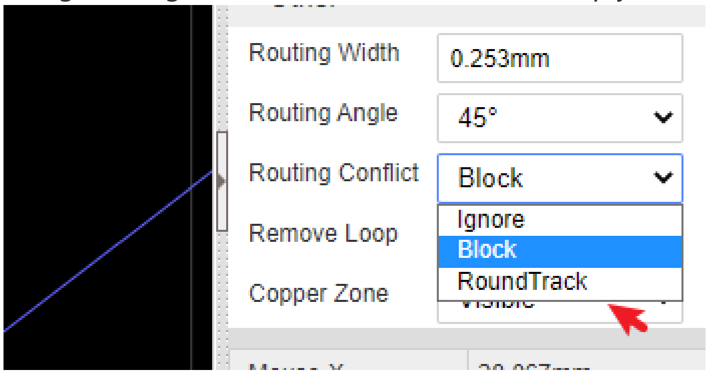
- Set up Remove Loop while routing, it only works on copper layer.



Other

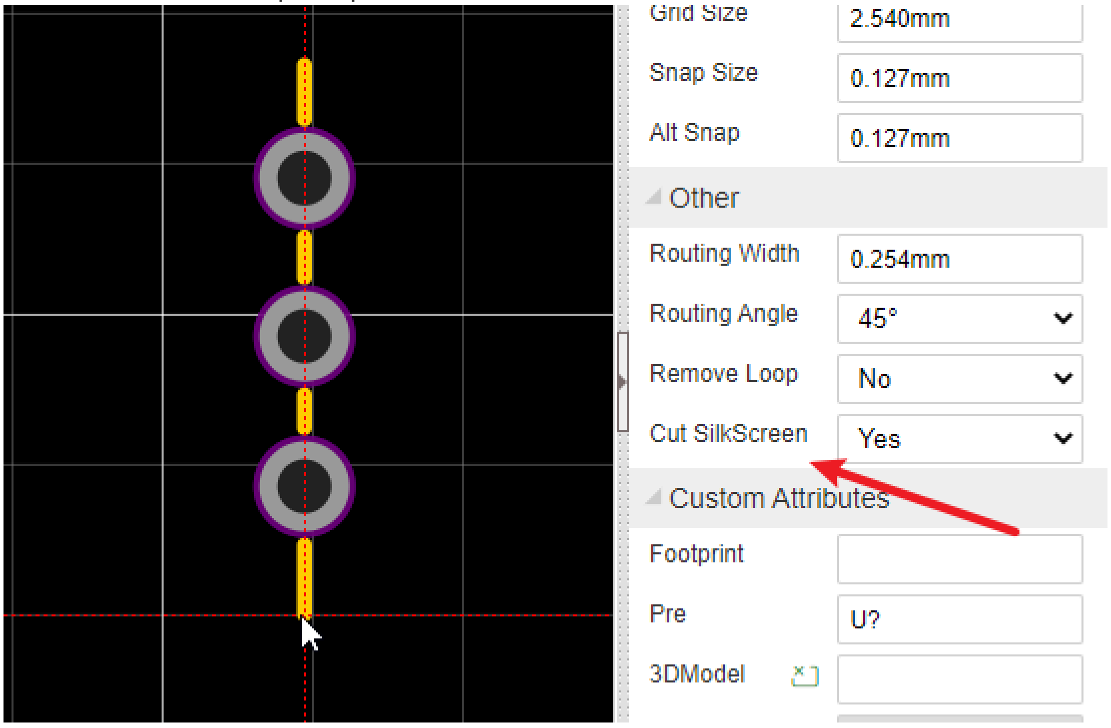
Routing Width	0.253mm
Routing Angle	45°
Routing Conflict	Block
Remove Loop	Yes
Copper Zone	Visible

- Using Routing Conflict as "RoundTrack" will help you finish routing quickly.



Routing Width	0.253mm
Routing Angle	45°
Routing Conflict	Block
Remove Loop	
Copper Zone	

- When edit the footprint document, you can set up the "Cut Silkscreen" to avoid the silkscreen track overlap the pad.



Grid Size	2.540mm
Snap Size	0.127mm
Alt Snap	0.127mm

Other

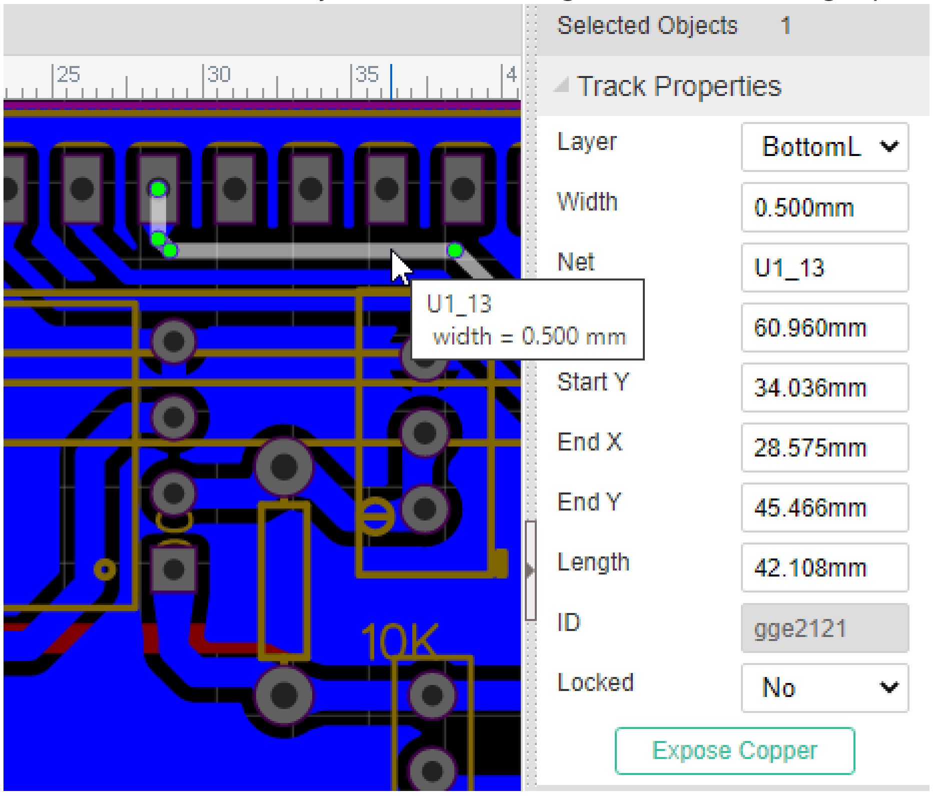
Routing Width	0.254mm
Routing Angle	45°
Remove Loop	No
Cut SilkScreen	Yes

Custom Attributes

Footprint	
Pre	U?
3DModel	

Track Length

- When a track is selected, you can find its Length attribute in the right panel.



Selected Objects 1

Track Properties

Layer	BottomL
Width	0.500mm
Net	U1_13
	60.960mm
Start Y	34.036mm
End X	28.575mm
End Y	45.466mm
Length	42.108mm
ID	gge2121
Locked	No

Expose Copper

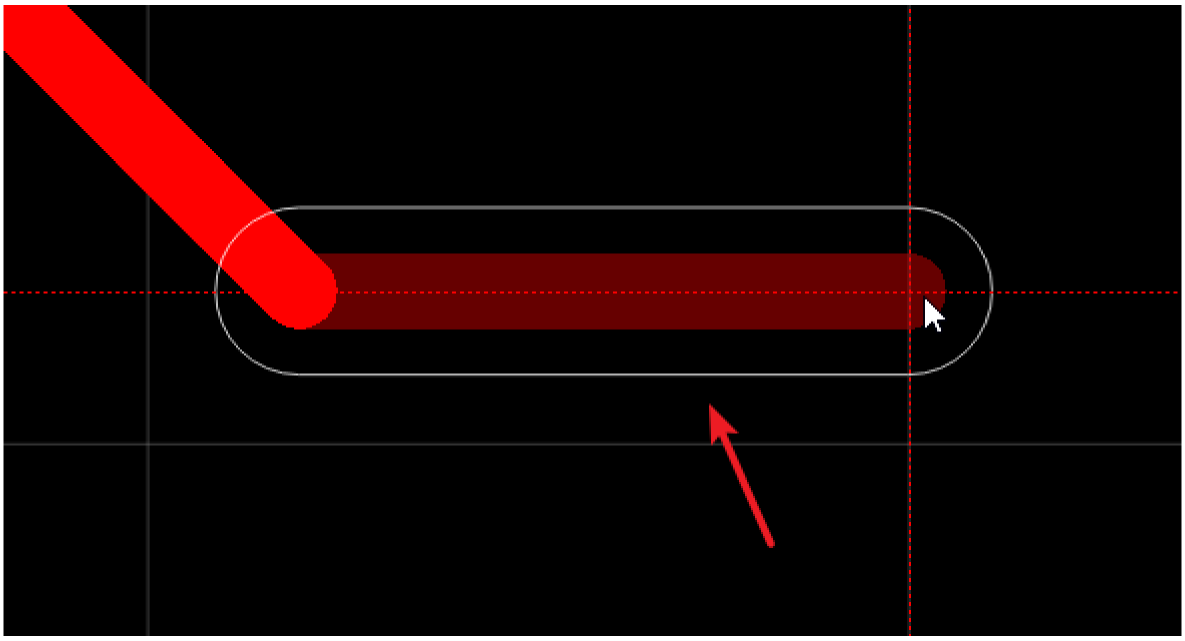
- At left-hand Design Manager, click a net, will pop up a dialog to show you this net track length.
- Click a track, press hotkey H will keep hightlight this track and net, and show this net's length.

Delete a Segment from a Track

- While routing, if you want to undo previous track path, you can press key "Delete" or "Backspace".
- Move your mouse to the segment which you want to delete, click it, then hold `SHIFT` and **double click it**. the segment will be removed. Or right-click delete the node.
- Right-click the track node to delete the track
- Click the track, right-click delete it, or press "Delete" key directly.

DRC outline

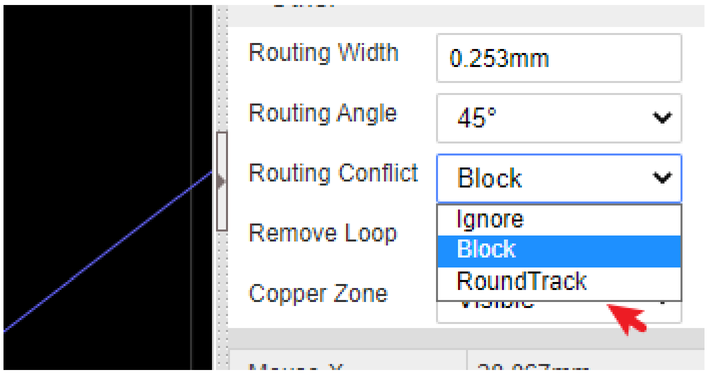
When you routing a track on the signal layer, you will see an outline around the first track, it is the DRC outline, the clearance from outline to the track edge depends on your Design Rule(DRC) clearance setting.



Routing Conflict

When the PCB comes from the schematic converted, the "Routing Conflict - Block" will be opened automatically.

At the right-hand attributes panel - others, you can find a "Routing Conflict" option:



Routing Width: 0.253mm
Routing Angle: 45°
Routing Conflict: Block

- Ignore: You can route the track overlap the different net name objects.
- Block: If the track net name different with other objects, this track will be blocked when routing.
- RoundTrack: The track while routing will walk arroud the different net objects.
- Push： Doesn't develop yet.

Differential Pair Routing

EasyEDA provide a easy experience for the differential pair routing.
Via: Top Menu - Route - Differential Pair Routing

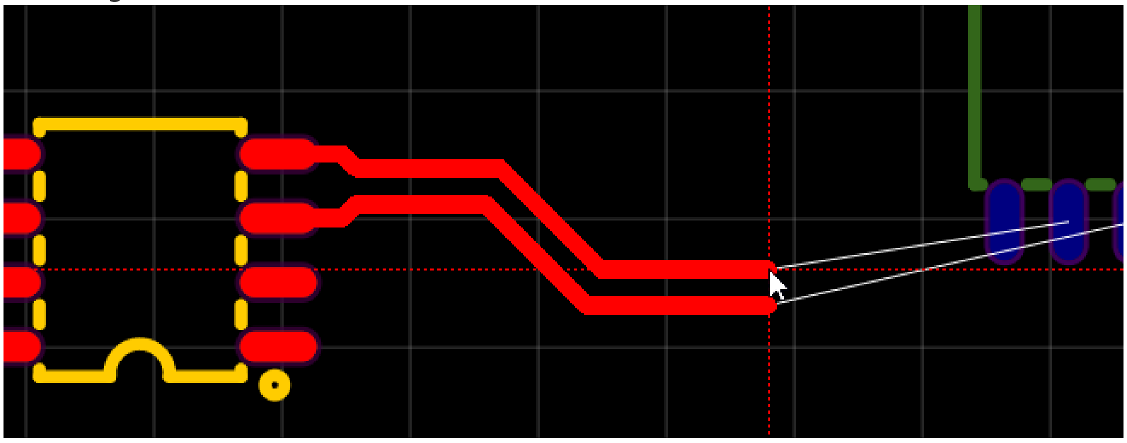
You must make sure the Differential Pair net names must be `XXX_N, XXX_P` or `XXX+,XXX-`.

and you need to set Differential Pair net rule at the "Top Menu - Tool - Design Rule" first.

How to route Differential Pair:

- 1.Set the Differential Pair net name as `XXX_N, XXX_P` or `XXX+,XXX-`, and set the rule for the Differential Pair net at the "Design Rule"
- 2.Click the menu `Top Menu - Route - Differential Pair Routing`
- 3.Click the one pad of the Differential Pair pads

- 4.Routing

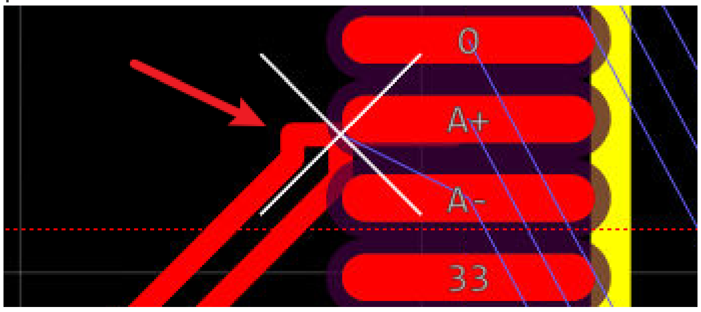


Notice:

- Only for 45 degrees routing, doesn't support hotkey L and Space key.
- Doesn't support the fanout routing.
- Doesn't support the DRC blocking.

Known Issue:

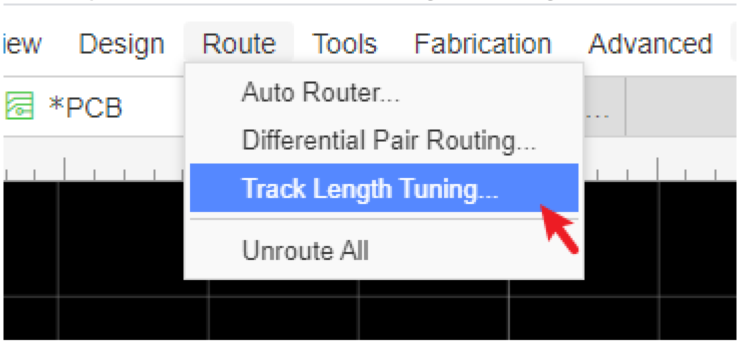
- When finish previous routing location too close with the finish pads, the track will generate the extra segments, please finish the previous location far away from finish pads.

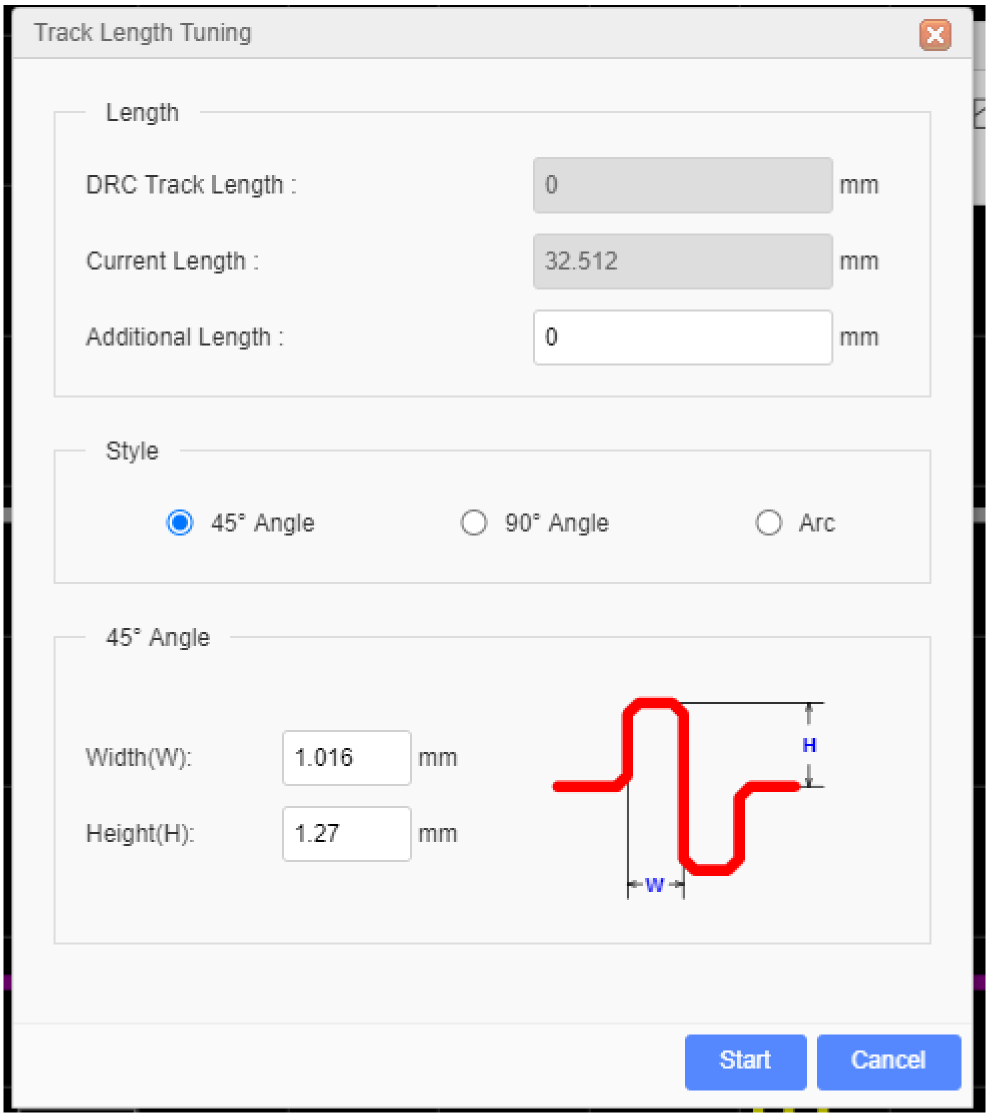


Track length Tunning

You can tunning your track very easy on the editor.

Via: Top Menu - Route - Track length Tuning




Track Length Tuning

Length

DRC Track Length : 0 mm

Current Length : 32.512 mm

Additional Length : 0 mm

Style

[✓ 45° Angle] 90° Angle Arc

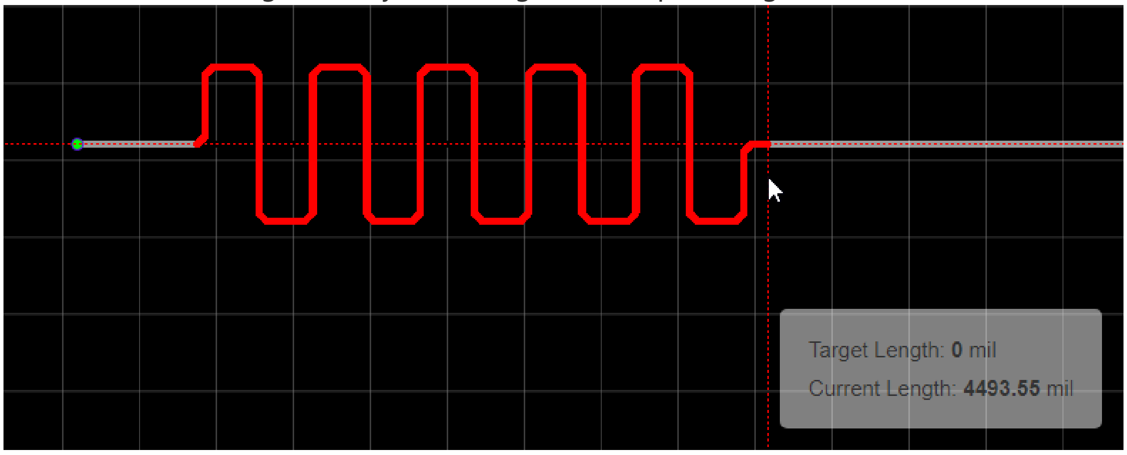
45° Angle

Width(W): 1.016 mm

Height(H): 1.27 mm

How to use:

- 1.Select the track which is you want to tune
- 2.Click the menu: `Top Menu - Route - Track length Tunning`
- 3.Set the parameter, start
- 4.Left-Click the track where is you want to start, and then move the mouse
- 5.When the track length close your setting, it will stop tunning.


Target Length: **0** mil

Current Length: **4493.55** mil

Notice：

- Doesn't support one side tunning for a track yet
- Doesn't support auto push or avoid the nearby tracks yet

Cloud Auto Router

For some simple or prototype PCBs, you may want to use the auto router function to save time. Layout is a time costly and dull job. EasyEDA spends lots of time to provide such a feature and it is loved by our users.
Before using the auto router, you need to set the board outline for the PCB.

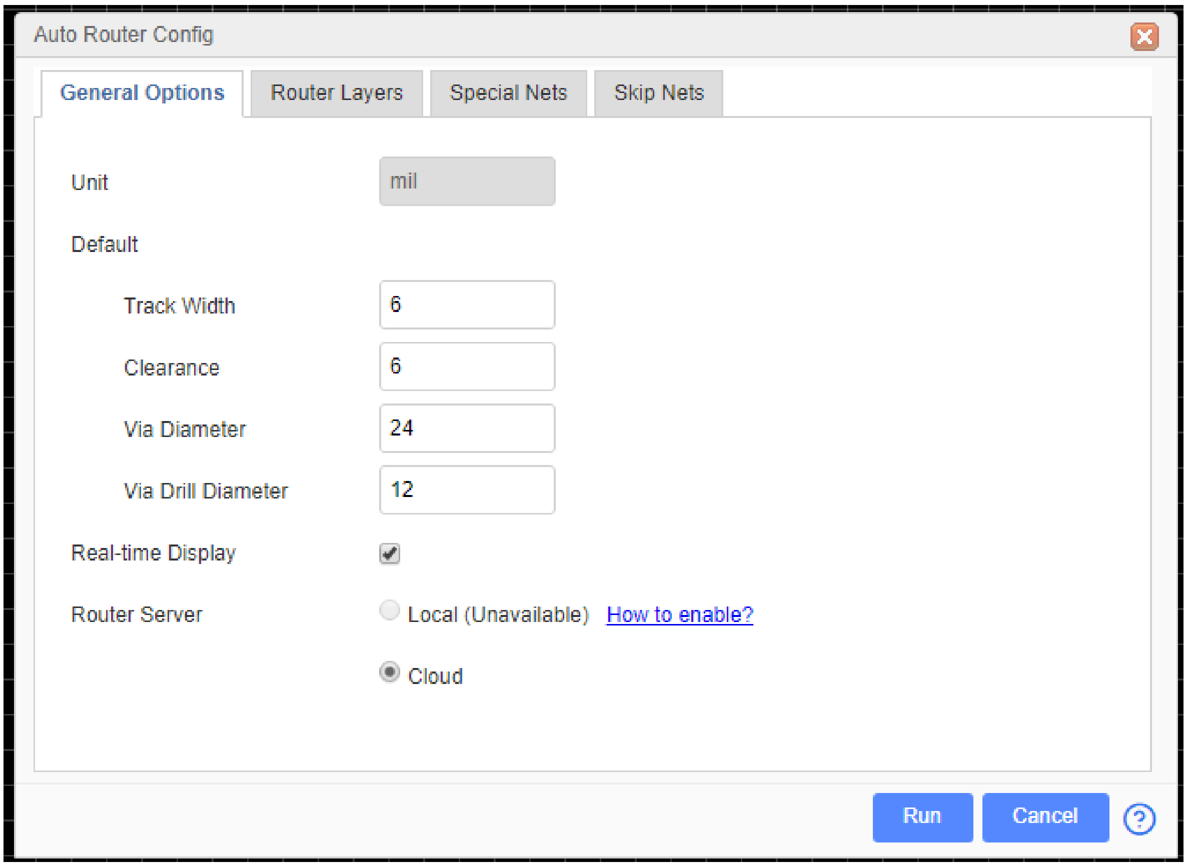
Auto router is not good enough! Suggest routing manually! You can use "RoundTrack(Walk Arroud)" option to route tracks, via right-hand panel - Routing Conflict.

Steps:

1 Click the the auto router button from the Top Menu"Top Menu> Route > Auto Router"

2 Config the auto router

After you click that button, you will get a config dialog like in the image below.


Auto Router Config

General Options | Router Layers | Special Nets | Skip Nets

Unit	mil
Default	
Track Width	6
Clearance	6
Via Diameter	24
Via Drill Diameter	12
Real-time Display	✓
Router Server	○ Local (Unavailable) How to enable?
	● Cloud

In the config dialog, you can set some rules to make the auto router result professional. These rule must equalize or more than DRC setting.

General Options

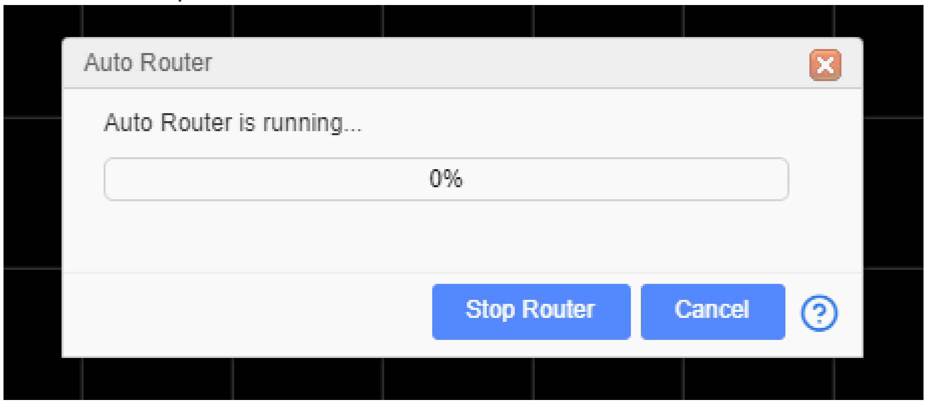
- **Unit**: The unit follows PCB canvas unit.
- **Track width**: The auto-route track width.
- **Clearance**: The clearance of the objects.
- **Via Diameter/Via Drill Diameter**: The via placing by auto-router.
- **Realtime Display**: when you select it , the real time routing status will show on.
- **Router Server**:
  - **Cloud**: Using EasyEDA online server.
  - **Local**: Using the local auto router server, when you click the Auto Router icon, the editor will check the local router server available or not automatically. How to use

please see as below.

- **Router Layers**: If you want to route inner layer, you have to enable the inner layer first.
- **Special Nets**: For the power supply track, you may want it to be bigger, so you can add some special rules.
- **Skip Nets**: If you like to keep the a net with no route, you can skip it. For example, if you want to use copper area to connect `GND` net, you can skip the `GND` net. If you want to reserve the routed track, you need to select the `Skip Routed Nets`.

3 Run it

After click the **"Run"** button , The real time check box will let you see how it is going, but it will make the process a little bit slow.

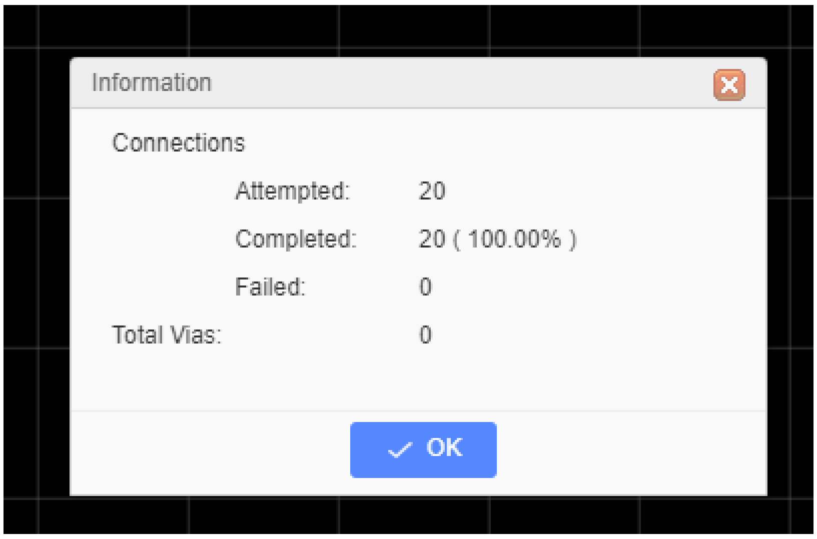

Auto Router

Auto Router is running...

0%

Waiting for a few minutes, after adding bottom and top copper area, you will get a finished PCB board.

When finish, will pop up a window.


Information

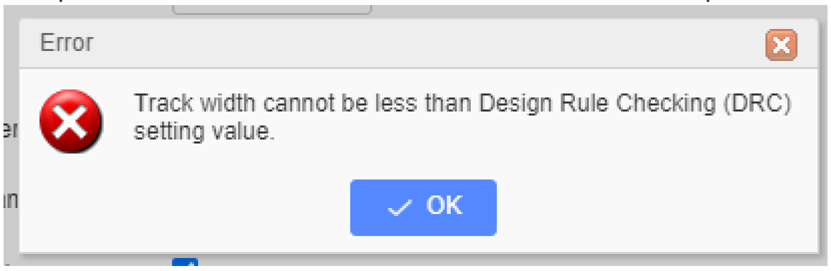
Connections

Attempted:	20
Completed:	20 (100.00%)
Failed:	0
Total Vias:	0

The connection means the track connect times.

Notice:

- The parameter can't less than DRC rule, otherwise will report error.


Error

Track width cannot be less than Design Rule Checking (DRC) setting value.

Local Auto Router

EasyEDA suggest that using local auto router rather than using the cloud server, because when many users using cloud server, the cloud auto router will fail. Only support 64bit system.

For the local auto router, please follow the steps as below:

- **1.Download the local auto router server.**

Download EasyEDA Router:
easyeda-router-windows-x64-v0.8.11.zip
easyeda-router-linux-x64-v0.8.11.zip
easyeda-router-mac-x64-v0.8.11.zip

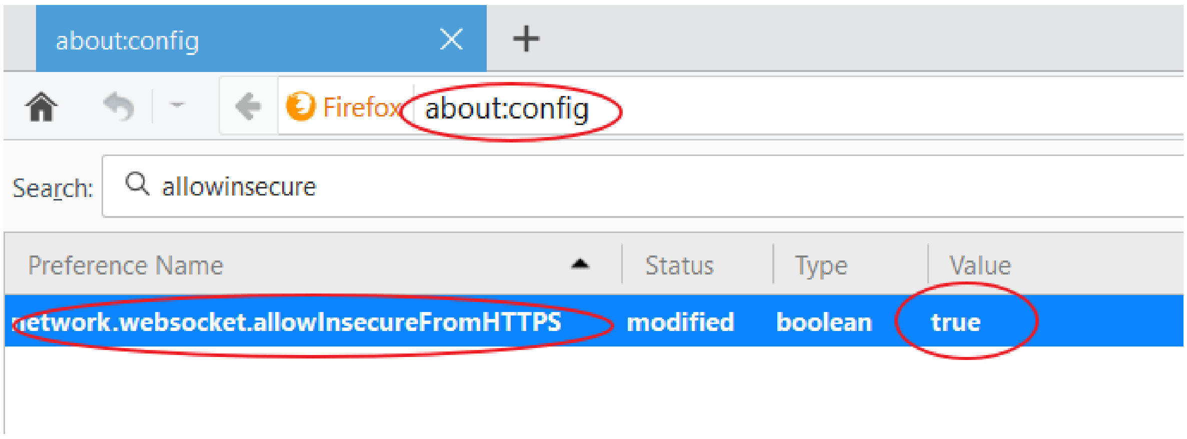
Supported OS:

```
-   Windows7(x64) or later 64bit Windows
-   Ubuntu17.04(x64) or other 64bit Linux, Linux recommend [Deepin]
(https://www.deepin.org)
-   macOS(x64)
```

- **2.Unzip it to the User folder, such as driver D.**
- **3.Configure the browser.**
  Notice: *Please use the latest Chrome or Firefox !!!*
  - **1)Chrome**
    The Chrome Browser don't need to be configure, If the local auto router is unavailable, you have to upgrade Chrome to version 60.0.3112.78 or later.
  - **2)Firefox**
    - 1.Type "about:config" into the address bar then press enter.
    - 2.Search and double click the options as below (change the values to "true"):

`network.websocket.allowInsecureFromHTTPS`

`security.mixed_content.block_active_content`



Preference Name	Status	Type	Value
network.websocket.allowInsecureFromHTTPS	modified	boolean	true

  - 3.Re-open Firefox.

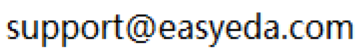
- **4.Open the decompress folder, Start local Auto Router(don't need to install, just run it and keep the command window open)**:
  - Double click `win64.bat` in Windows.

    - Run `sh lin64.sh` on command terminal in Linux. Open the terminal, use the `cd` command to change the directory to the `lin64.sh` location, and type `sh lin64.sh`, then enter.
    - Run `sh mac64.sh` on command prompt in MacOS. Open the terminal, use the `cd` command to change the directory to the `mac64.sh` location, and type `sh mac64.sh`, then enter.
- **5.Open the editor, open the PCB, Click the** Auto Router** icon at editor to start auto-router.**
  If the local router server is available, the dialog will tell you. Click the **Run** button, the dialog will show the process.

Tips

Auto router is not good enough, suggest routing manually, recommed "Routing Confilct - RoundTrack" at the PCB right-hand panel.
Sometimes, if you can't get it done, try the tips below.

- Use local auto router rather than cloud server.
- Make sure the net of PCB doesn't contain the special charaters, such as ` { } ^ ; ~ \ / [ ] = etc. the chrarter - and _ are supported.
- Make sure the board oultine is closed, doesn't has board oultine overlap situation.
- Make sure there are no DRC cleance errors (short circuit issue), such as two different network pads overlapping, or different net pads in the same location within the package.
- Make sure no footprint outside the board outline.
- Make sure your canvas Gird size is set to 10mil, make sure the components align with the grid via: top menu - format - align grid.
- Place some vias at suitable position, and modify the vias' net as you want.
- Make sure PCB rule doesn't have 3 decimal places, EasyEDA auto router only support 2 decimal places.
- Skip the GND nets, add copper area to GND net.
- Use small tracks and small clearance, but make sure the value is more than 6mil.
- Route some key tracks manually before auto routing and ignore them when auto routing.
- Add more layers, 4 layers or 6 layers. but that will make more cost.
- Change the components layout, make them have more space between each other.
- Don't make any via/pad overlap the different objects which can be set the net.
- Stop the auto-router, close the script, and re-open the script to run local auto-router server again.
- Sometimes, The auto-router won't work at the first time, please modify the layout and try serveral times.
- If only a few remaining networks cannot be completed, it is a normal phenomenon. Please manually route the remaining networks.
- Tell the error detail to us and send your PCB file as EasyEDA Source file to support@easyeda.com.
  https://docs.easyeda.com/en/Export/Export-EasyEDA-Source-File/index.html
  via email.


support@easyeda.com

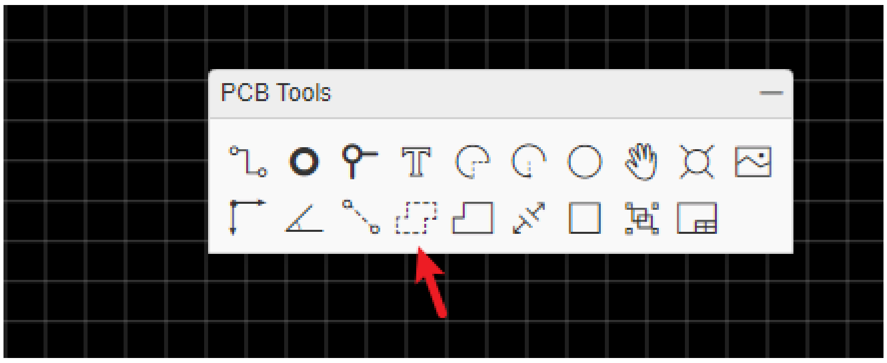
Some professional people don't like the auto router, because they think auto router is not professional, but you can use the auto router to check your placement to check the density of your PCB.

At present, the auto router is not good enough, suggest routing manually, we will improve it in the future.

Copper Area

Copper Area

Sometimes you will want to fill in or flood an area with copper(Copper Pour). Normally after drawing the copper area, set the net it is to be connected to (floating copper areas are not recommended because they can cause EMC and Signal Integrity (SI) problems).

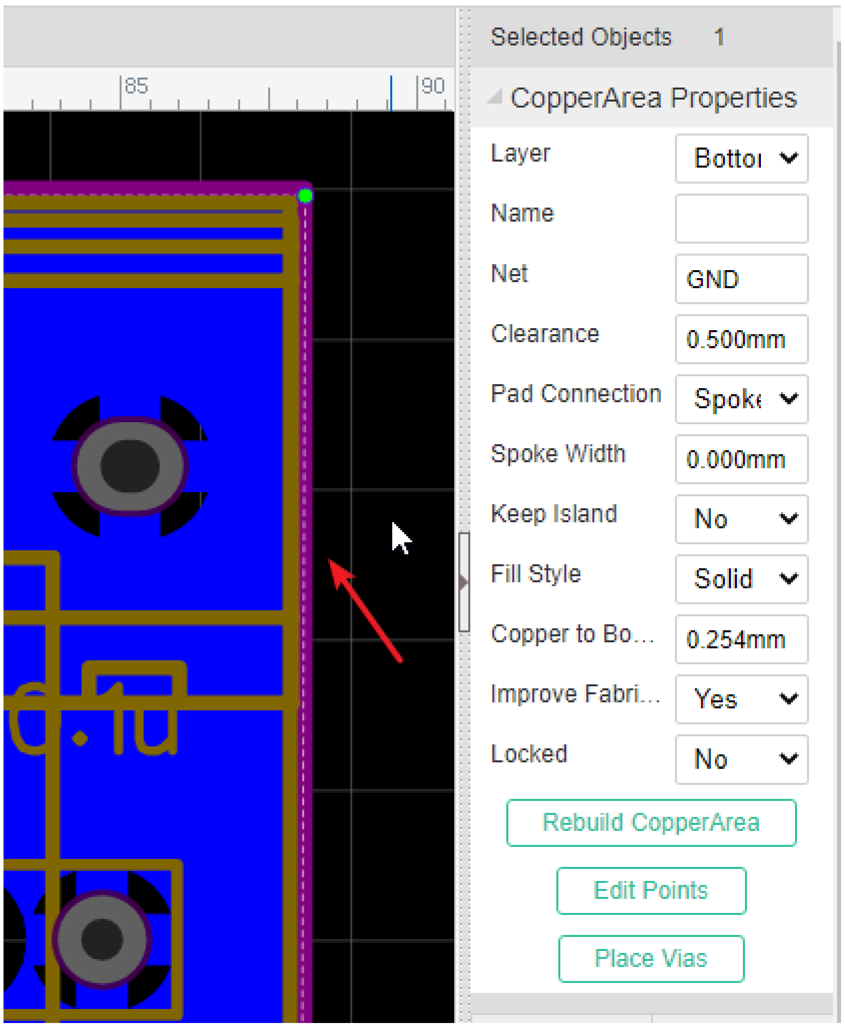


Before using Copper Area, please make sure your PCB has a closed board outline!

Usually this copper area will be connected to a net such as **GND** or a supply rail. You can draw the outline of a flood using the **Copper Area** button from the PCB Tools palette.

Copper Area Attributes

When selecting the copper area outline, you can find its attributes from the right hand **Properties** panels.


Selected Objects 1

CopperArea Properties

Layer	Bottom
Name	
Net	GND
Clearance	0.500mm
Pad Connection	Spoke
Spoke Width	0.000mm
Keep Island	No
Fill Style	Solid
Copper to Bo...	0.254mm
Improve Fabri...	Yes
Locked	No

Layer: Bottom, Top, Inner1, Inner2, Inner3, Inner4 etc.;

Net: the net that the copper area is connected to;

Name: set a name for it.

Clearance: clearance of the copper area from other nets and floods;

Pad Connection: direct or spoke (i.e. a cross shaped heat shunt);

Spoke Width: When Pad Connection is Spoke, you can set the Spoke width, which is copper area fill connect with Pads.

Keep Island: Yes/No. This keeps or removes any isolated areas of copper created as part of the flooding process. It is usually good practice to removes these unless you really need them to maintain a more even spread of copper (copper balance) on your PCB.

Fill Style: Solid/No Solid/Grid. Selecting **No Solid** will removes the fill so that you can see the tracks more clearly; when select Grid, you can set the grid spacing and grid width.

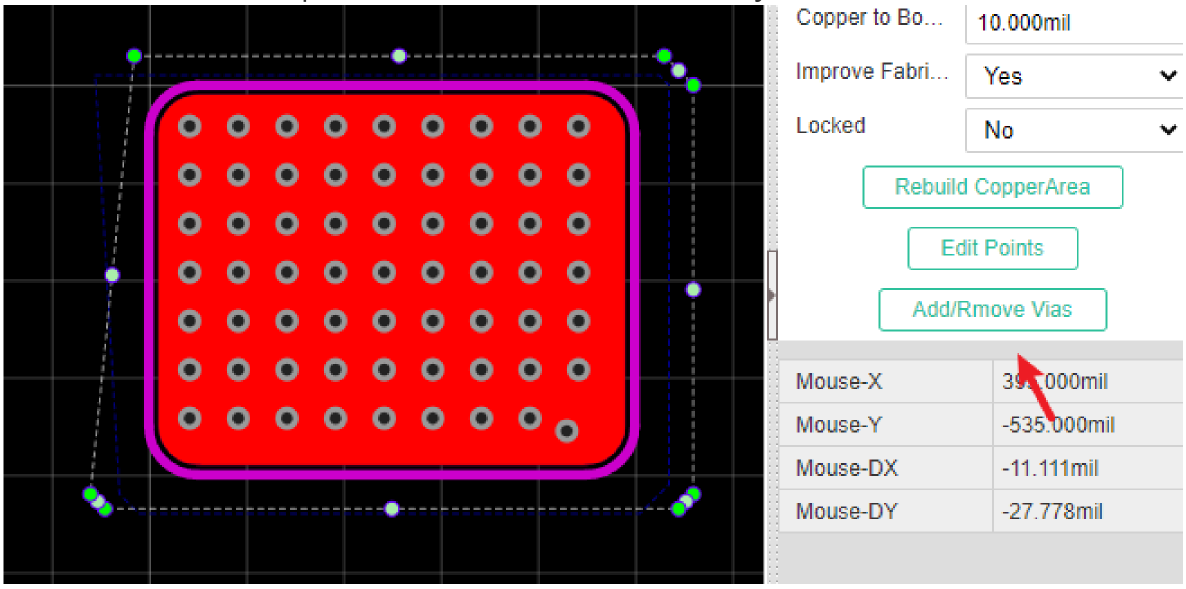
Copper to BoardOutline: Setting the clearance between copper with board outline.

Improve Fabrcation: Yes/No. If you set as No, you will see much sharp copper corners, that is not good for PCB fabrication.

Rebuild CopperArea: Click the button to Rebuild Copper Area if you make any changes.

Edit Points: You can edit the copper area shape manually, any shap as you want.

Add/Remove Vias: When you add copper areas at two and more layers which are having same net, you can add multiple vias for the copper fill area, just click the "Add/Remove Via" button, then set the via parameter. The vias will avoid the objects if the via conflict the DRC.

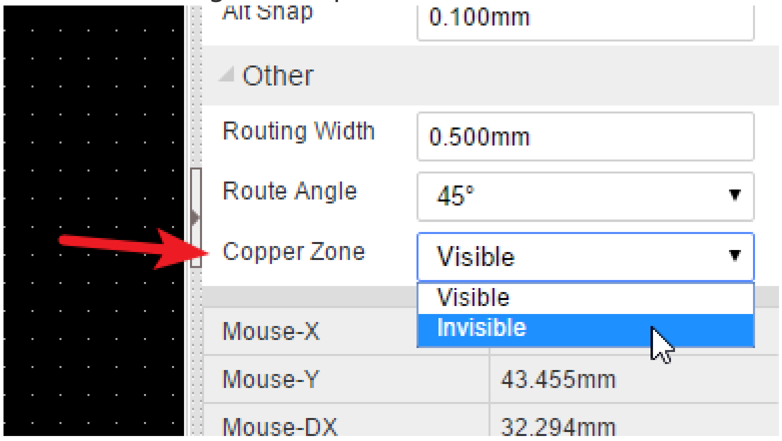


Copper to Bo...	10.000mil
Improve Fabri...	Yes
Locked	No

Mouse-X	39 .000mil
Mouse-Y	-535.000mil
Mouse-DX	-11.111mil
Mouse-DY	-27.778mil

Tips

- Hotkey `E` to start draw copper area.
- Hotkey `L` to change drawing type(90 degrees or 45 degrees or Arc)
- Hotkey `Shift+B` to build all of the copper areas.
- Hotkey `Shift+M` to hide copper areas fill zone, just show the copper outline.
- Hotkey `Delete` or `BackSpace` to redo previous steps.
- If you after copper pours but no copper fills show up, you need to set it a net same one of the PCB nets, or keep the island as YES, and the rebuild the copper area via "Rebuild Copper Area" button or "SHIFT+B".
- If you want to hide the copper area and keep routing tracks, you can set the copper zone invisible at the right-hand panel.



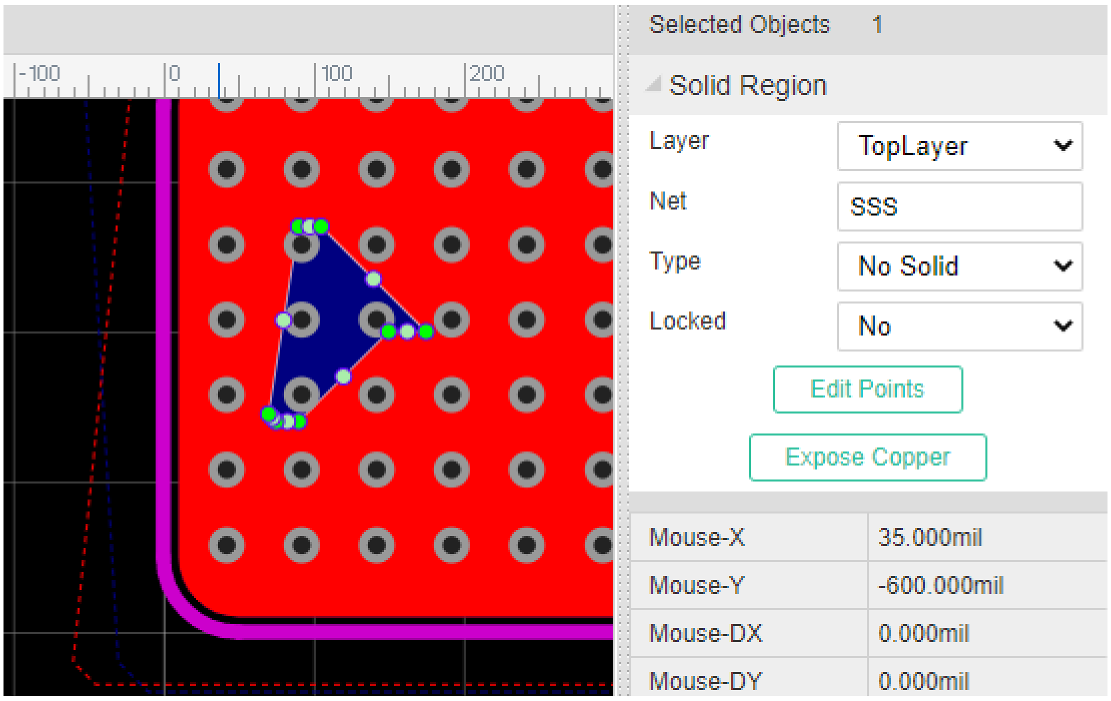
Alt Snap	0.100mm

Other

Routing Width	0.500mm
Route Angle	45°
Copper Zone	Visible

Mouse-X	
Mouse-Y	43.455mm
Mouse-DX	32.294mm

- If you want to cutout some copper corners, you can use "Solid Region - No Solid", and then set different net for it, and rebuild the copper area.



Selected Objects 1

Solid Region

Layer	TopLayer
Net	SSS
Type	No Solid
Locked	No

Mouse-X	35.000mil
Mouse-Y	-600.000mil
Mouse-DX	0.000mil
Mouse-DY	0.000mil

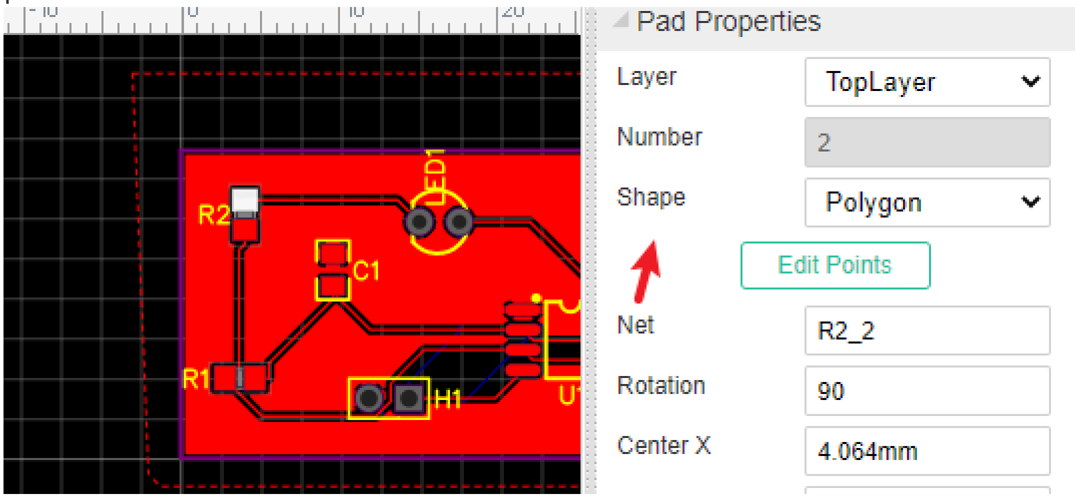
Notice

- *Because of the browser's performance issue, EasyEDA doesn't support the real-time copper pour, after PCB modifying, please rebuild copper area via Hotkey* `shift+B` .
- *EasyEDA doesn't support click the copper zone, you need to click the copper outline to select it.*
- **If the PCB size is more than 15MB, the copper filled data is stored in the client or browser(that is because some copper filled data is too large to save at server), and the copper area outline data is stored in the file. Therefore, when the PCB is opened for the first time, the copper area filled data will be automatically pouring and saving at local, and the second time the PCB is opened, the filled data will be automatically loaded from the local storage. When you need to draw the forbidden copper-laying area, please use the "No Solid" property of "Fill Type" to cutout the copper area and rebuild it, do not use the operation of drawing the area with wires or circles and then removing the wires or circles to create the forbidden copper pour area. If the PCB size is less than 15MB, the copper filled data will be saved into file, and you can open the PCB in another PC without rebuild copper area.**

FAQ

Why sometimes it takes a long time to copper pour

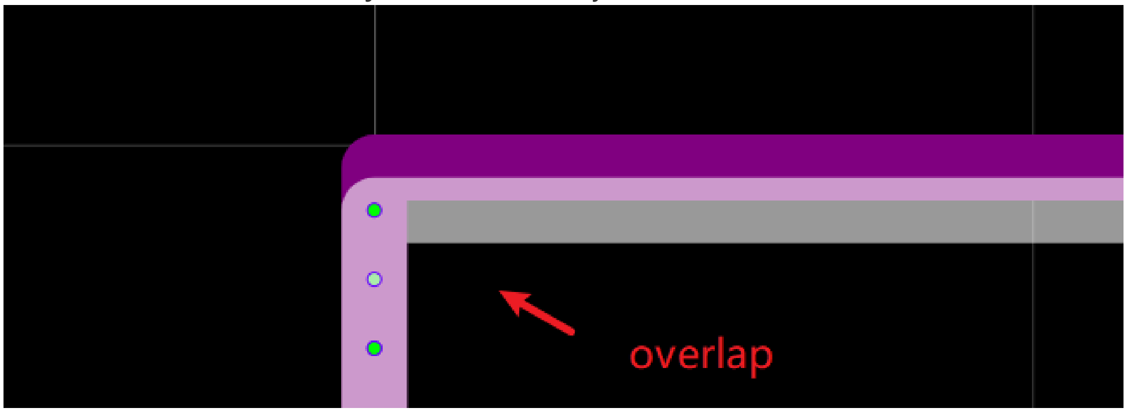
- Check that the PCB has a large number of polygon pads, which generally appear in the PCB imported Altium Design files, and if so, manually modify them to Round or Rectangle pads.



Pad Properties

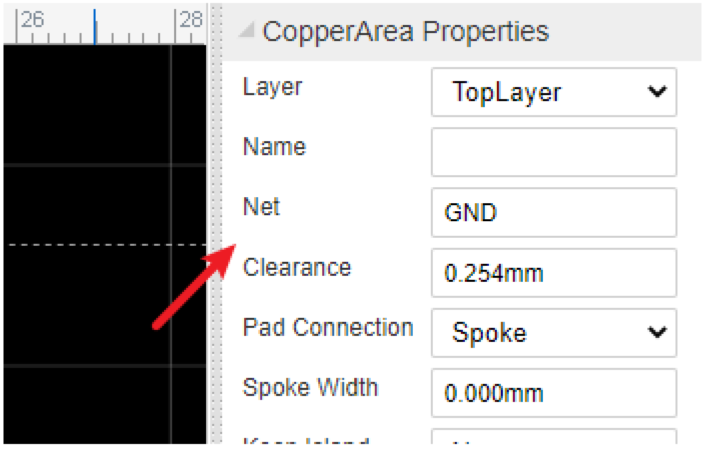
Layer	TopLayer
Number	2
Shape	Polygon
Net	R2_2
Rotation	90
Center X	4.064mm

- Check if there are a large number of wire arcs, generally appear in the imported Altium Design PCB, Altium Design picture is a large number of track segments combined, need to be manually removed.
- Check that the board outline is complicated, with overlapping board outlines, or a large number of board outlines, adjust them manually to reduce the number of board outlines.



Why did I not show the copper fill after copper poured

- Your copper area net must have the same pad or via same as the current layer, otherwise it will be considered an island to be removed. Click on the copper wire frame to modify the net in the property panel on the right. For example, your pad net is `VCC`, you lay copper net needs to be set to `VCC`.

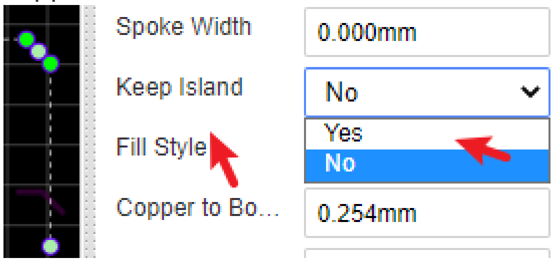


CopperArea Properties

Layer	TopLayer
Name	
Net	GND
Clearance	0.254mm
Pad Connection	Spoke
Spoke Width	0.000mm

- If you don't change the copper area ne, you can click on the copper outline and modify the property "Keep Island" to Yes in the right property panel.
The copper area logic of the EasyEAD is based on whether there is a connection or not to decide whether it is an island, and if there is no element connection to the same net, the

copper area will be considered an island.



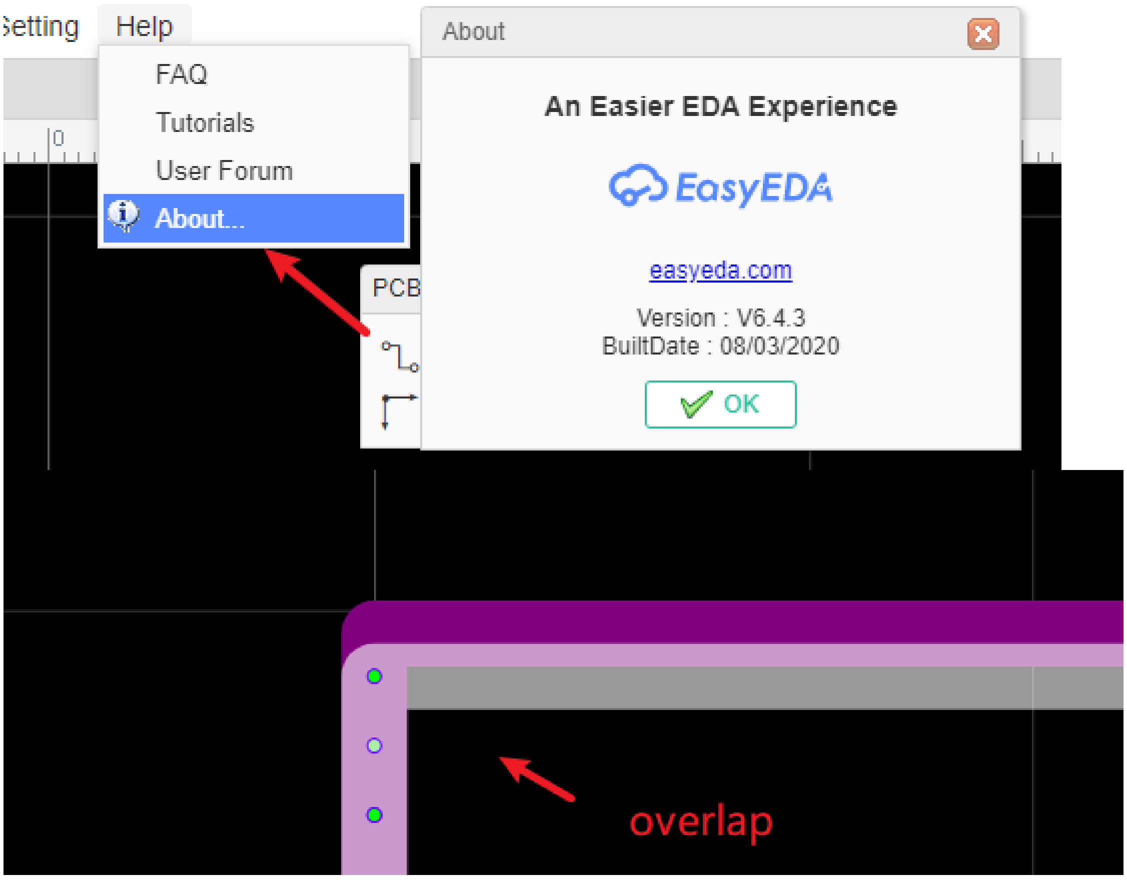
Spoke Width 0.000mm

Keep Island No

Fill Style

Copper to Bo... 0.254mm

- Check that the editor version is 6.3 above, 6.3 PCB board open in version 6.2 can not properly copper pour. Please CTRL+F5 refresh editor page upgrade to 6.3, if it is true that can not upgrade to 6.3, you must remove the copper area and redraw.



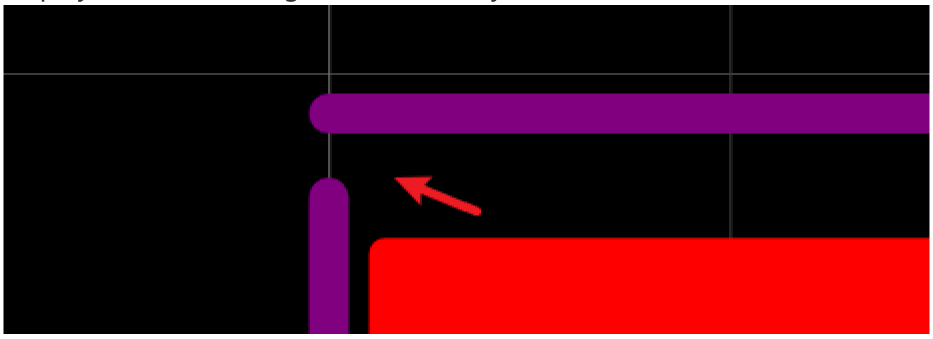
About

An Easier EDA Experience

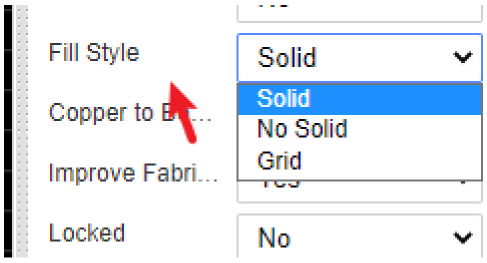
easyeda.com

Version : V6.4.3
BuiltDate : 08/03/2020

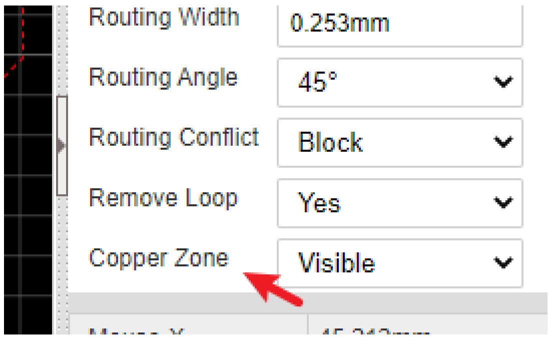
- Check that the board outline is closed and that endpoints need to be closed between the tracks, and that there are overlapping segments of the board outline (usually inside the imported PCB). Once you can hide all layers, only the board outline layer view is displayed, and each segment is carefully examined.



- Check that the copper area property is set to type No Solid and needs to be set to Solid or Grid.


Fill Style
Solid
Solid
No Solid
Grid
Copper to B...
Improve Fabri...
Locked
No

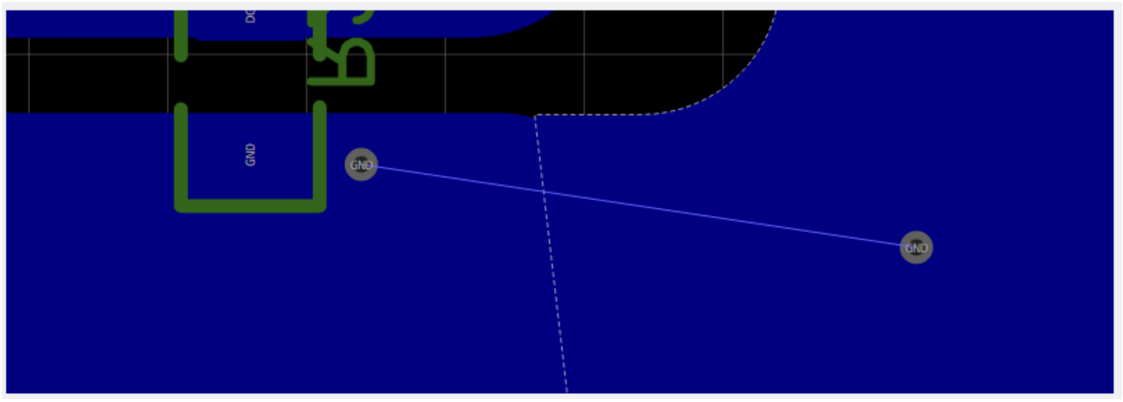
- Whether to make the copper area invisible, on the right side of the canvas, set the copper zone to Visible.


Routing Width 0.253mm
Routing Angle 45°
Routing Conflict Block
Remove Loop Yes
Copper Zone Visible

- Still unable to copper pour may be an editor bug, please contact us.

Why the ratline doesn't disappear when two copper area overlap which are the same net

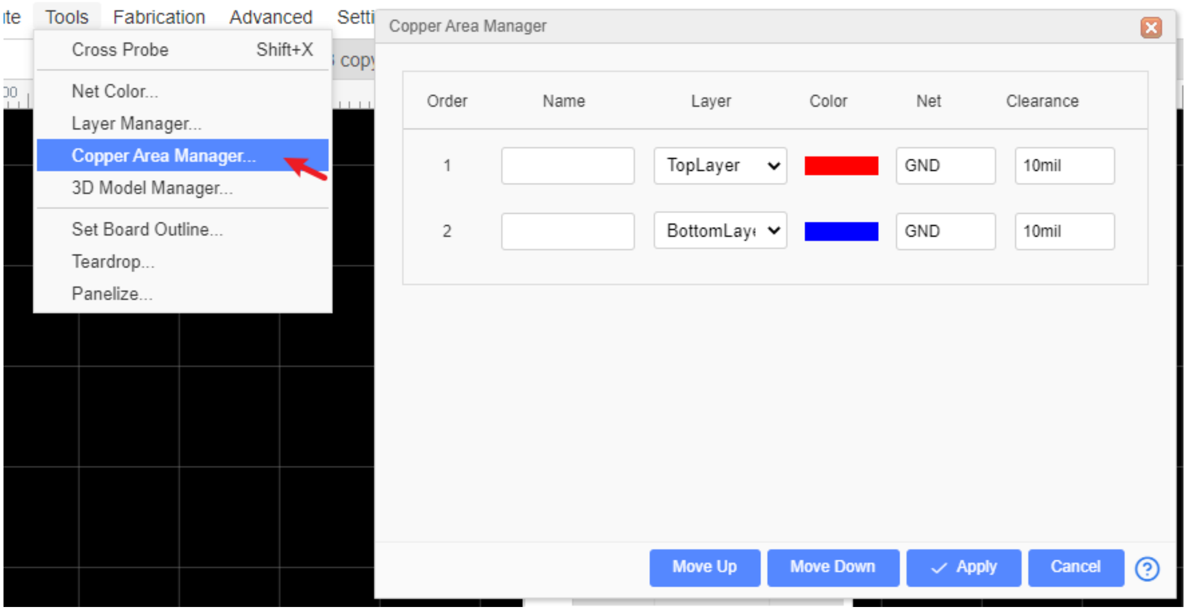
- At present, doesn't support this, that will make the ratline generation very slow. Please route a track manually to connect these copper areas.



Copper Area Manager

EasyEDA support copper area manager now, you can set the copper order and apply, the forward copper area will be poured first.

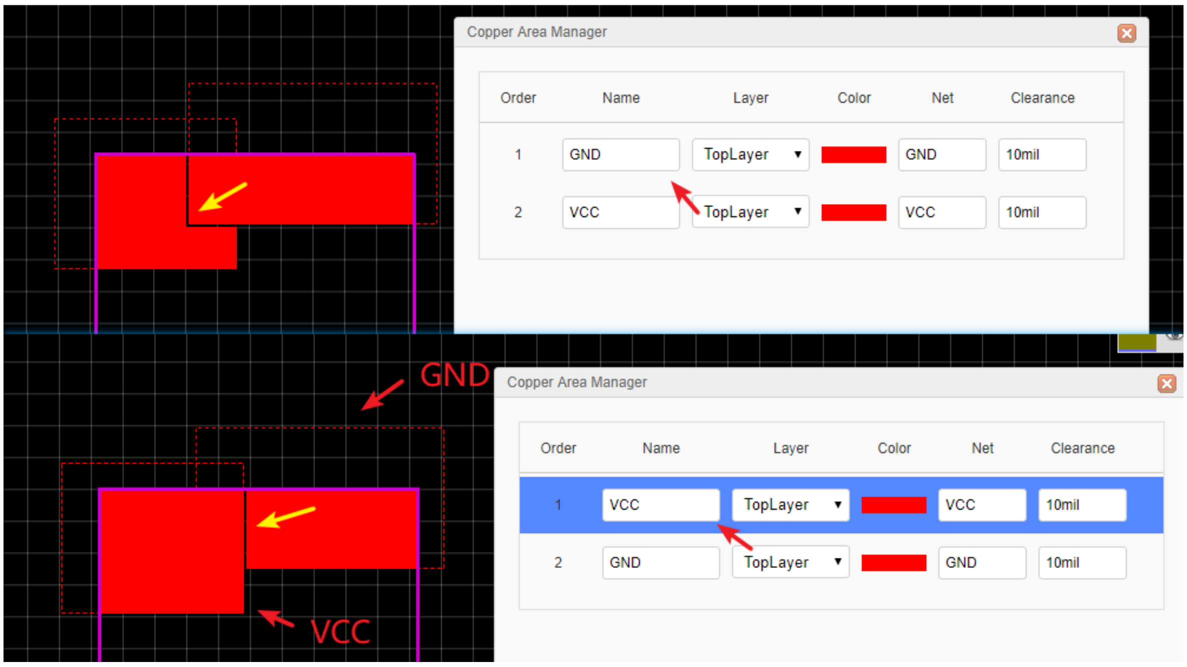
Via: Top Menu - Tools - Copper Area Manager



Copper Area Manager

Order	Name	Layer	Color	Net	Clearance
1		TopLayer	🟥	GND	10mil
2		BottomLay	🟦	GND	10mil

For example:
The GND on the top and VCC on the top, you can see the clearance is different.



Copper Area Manager

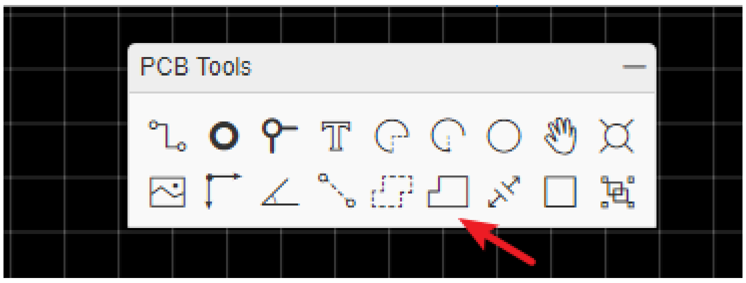
Order	Name	Layer	Color	Net	Clearance
1	GND	TopLayer	🟥	GND	10mil
2	VCC	TopLayer	🟥	VCC	10mil

Copper Area Manager

Order	Name	Layer	Color	Net	Clearance
1	VCC	TopLayer	🟥	VCC	10mil
2	GND	TopLayer	🟥	GND	10mil

Solid Region

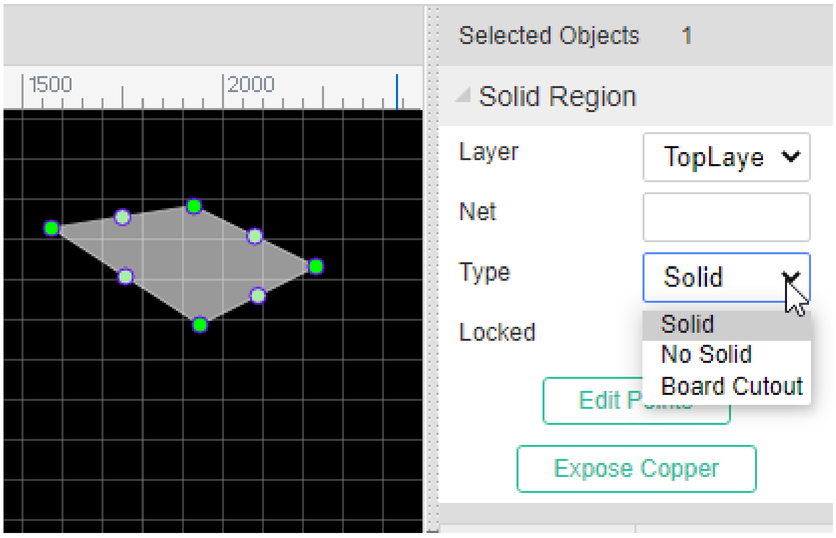
EasyEDA has added a new tool Solid Region for PCB design



This is a very useful, quick way to connect Pads. You can draw a Solid Region to include all of these pads with same net name, then set the region to the same net name as the pads. It is like Copper Area but easier to use for small areas. To use Solid Region like this, set the Type attribute (in the right hand Properties panel) to Solid.

When you drawing the solid region, you can use the hotkey `L` and `space` to change the route type(Arc, 90 degrees, 45 degrees, Free Angle), just like the track routing.

When you finish drawing, you can click the solid region and change its attributes at the right-hand panel.



Selected Objects 1

Solid Region

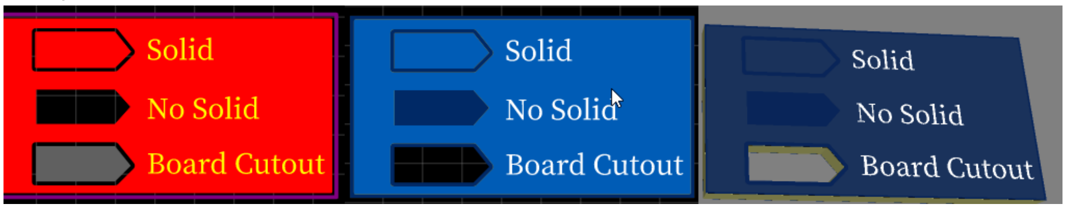
Layer: TopLaye

Net:

Type: Solid

Locked

- **Layer**: Solid Region su pport many layers, you need to enable the layer at the Layer Manager first.
- **Net**: When change to top or bottom or other inner signal layer, the solid region can be set a net to connect other objects. Sometimes, you can use solid region to make the copper instead of "Copper Area".
- **Type**: Solid,Board Cutout,No Solid ,
    - **Solid**: It will fill the solid area.
    - **No Solid**: It will cutout the area such as copper area. **Notice, if you cutout a copper area, the solid region's net must different than copper area's net.** After setting to this option, you need to rebuild the copper area with SHIFT+B.
    - **Board Cutout**: you can use this feature to create a slot hole(Non Plated Through Hole).

- **Edit Points**: You can edit the solid region's outline points as you want.
- **Expose Copper**: ou can create an aperture in the solder mask by one click. It's very easy to do.

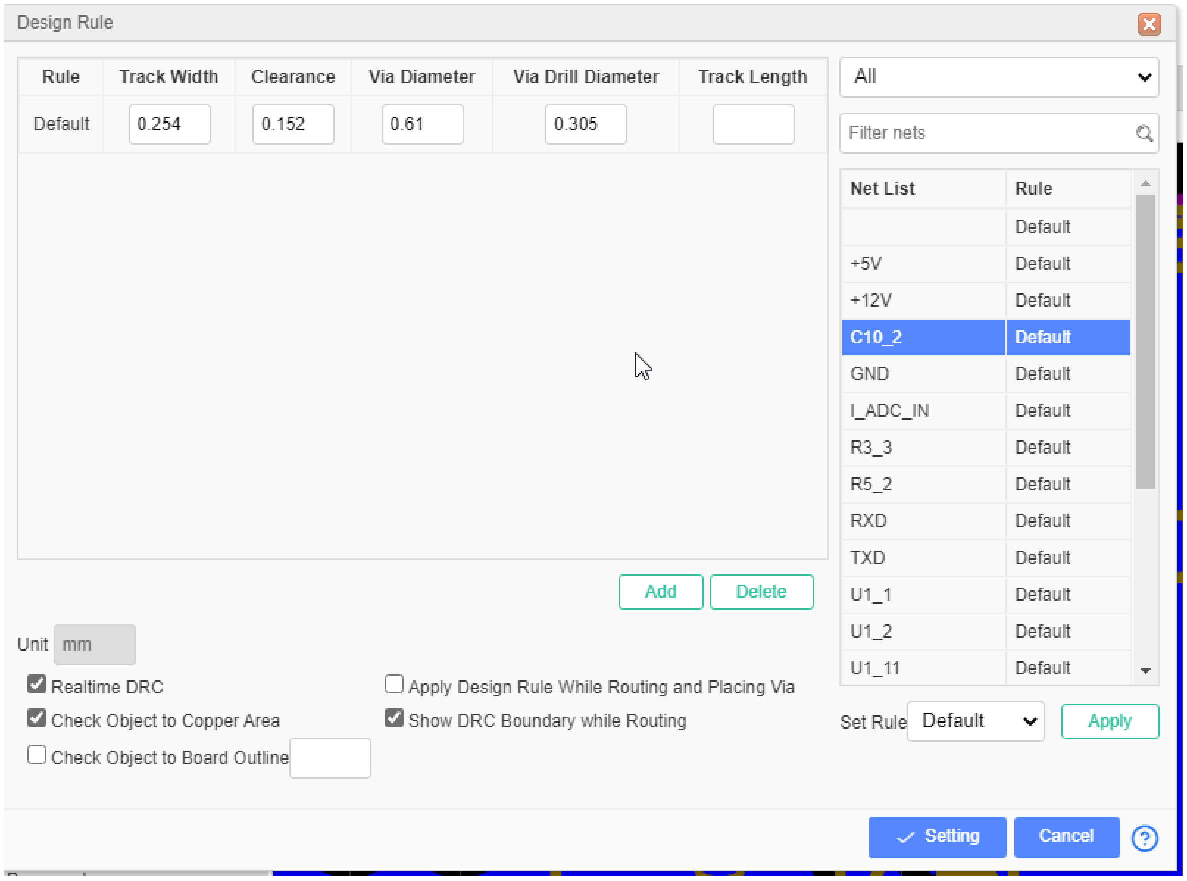
The outline of the solid region can not be self-intersection, when it happens, please delete the self-interation point at "Edit Points".

Design Rule Check(DRC)

EasyEDA provides a real time DRC(Design Rule Check) function. This is a big feature of EasyEDA. It is hard to fix DRC errors after laying out the PCB. Now EasyEDA will let you know the error in routing. You will find an `X` flag to mark the error.

Design Rule Setting

Via at: **Tools > Design Rule...**, or Via: **right-click the canvas - Design Rule...** to open the **Design Rule** setting dialog:



Design Rule

Rule	Track Width	Clearance	Via Diameter	Via Drill Diameter	Track Length
Default	0.254	0.152	0.61	0.305	

Net List	Rule
	Default
+5V	Default
+12V	Default
C10_2	Default
GND	Default
I_ADC_IN	Default
R3_3	Default
R5_2	Default
RXD	Default
TXD	Default
U1_1	Default
U1_2	Default
U1_11	Default

Unit mm

[✓] Realtime DRC
[✓] Check Object to Copper Area
[ ] Check Object to Board Outline
[ ] Apply Design Rule While Routing and Placing Via
[✓] Show DRC Boundary while Routing

Set Rule Default

The unit follow the canvas unit.

Rule: The default rule named "Default", you can add the new rule you can rename and set parameters for it. Each net can be set a rule.

Track Width: Current rule's track width. The PCB track width can not less than this value.

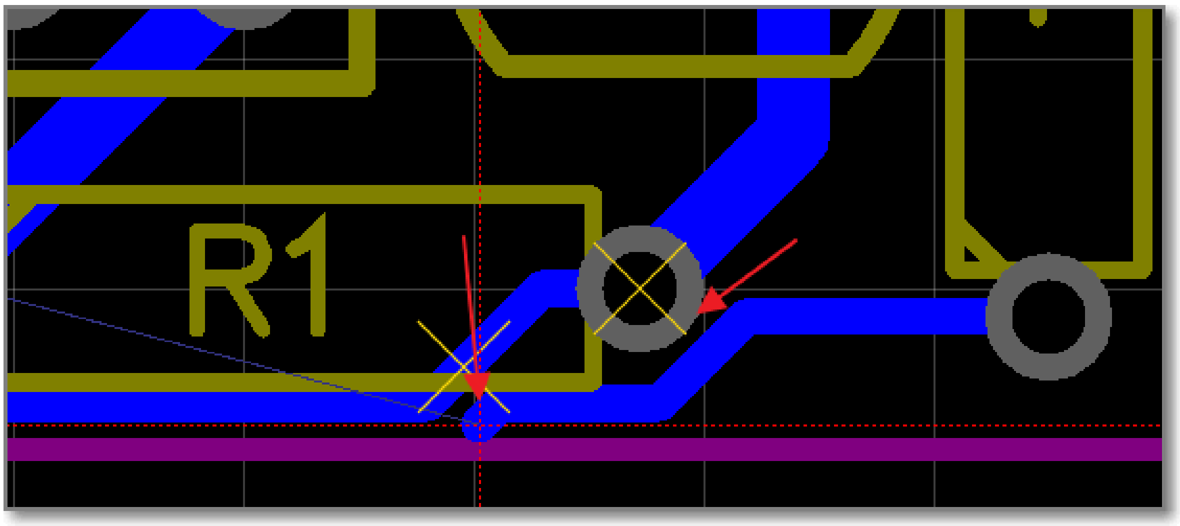
Clearance: The clearance of different objects which have different net. The clearance of the PCB can not less than this value.

Via Diameter: The via diameter of current rule. The via diameter of the PCB can not less than this value. Such as the Hole/Multi-layer Pad's diameter.

Via Drill Diameter: The via drill diameter of current rule. The via drill diameter of the PCB can not less than this value.

Track Length: All track length of current rule. The length of tracks belong to a same net should not be longer than this value.Including the arc length. When the input box is empty the length will be unlimited.

Realtime DRC: After enable, when you routing the DRC will checking all the time, when appear the error the canvas will show the "X" marking.



Check Object to Copper Area: Check the clearance of the objects to copper area. If you disable this option, you must rebuild the copper area before generating the Gerber with SHIFT+B.

Check Object to Board Outline: When you enable, you can set a value to check the clearance of the objects to board outline.

Apply Design Rule while Routing and Placing Via: When you routing and placing a new via, them will follow the design rule to set them width and size.

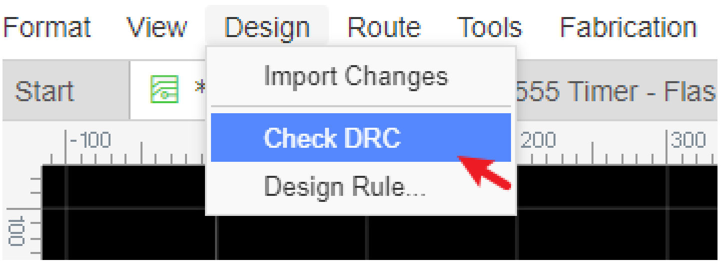
Show DRC Boundary while Routing: When routing you will see a oultine around the track. Its diameter depends on desgin rule.

Set Rule for a Net

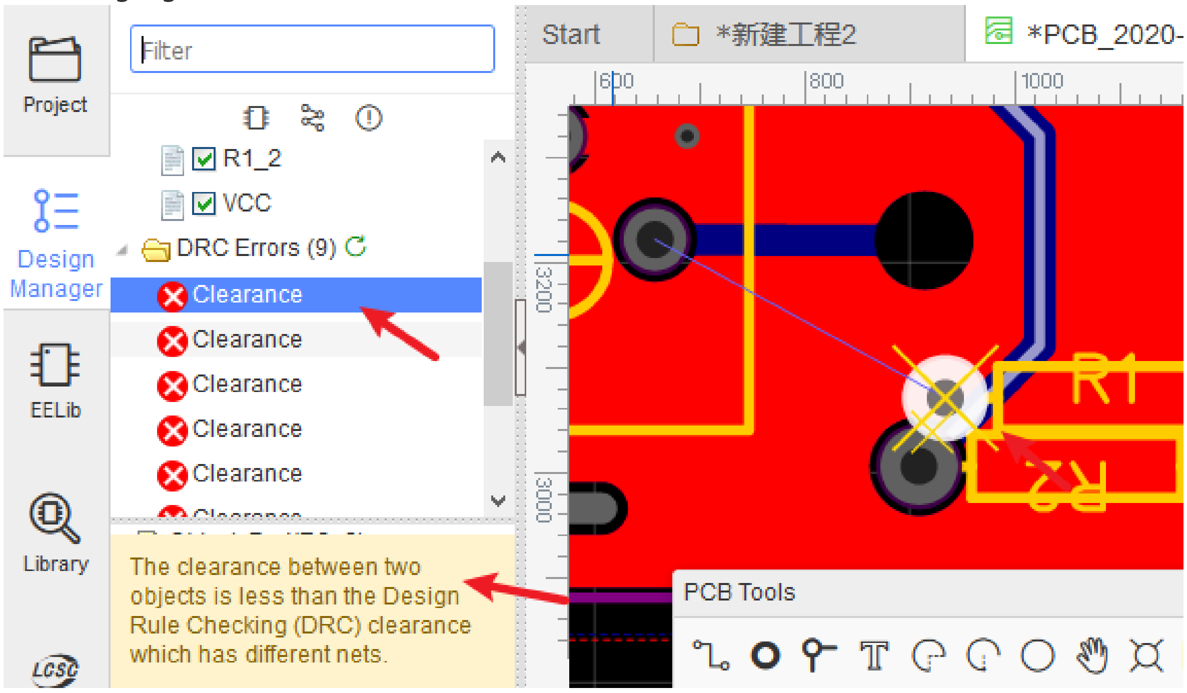
1. Click the "new" button to create a rule, or use the default rule
2. Select one or more networks on the right, support holding down the CTRL key for multiple selection, and also can perform keyword filtering and rule classification filtering
3. Then select the rule you want to set in the "set rules" section below and click the "apply" button. The network applies the rule.
4. Click the "Settings" button to apply the rule.

Check the DRC Error

Via "**Design Manager - DRC Error**" or "**Top Menu - Design - Check DRC**", click the refresh icon to run the DRC. If your PCB is a big file, and have the copper area that will take some times to check the DRC, please wait a while.



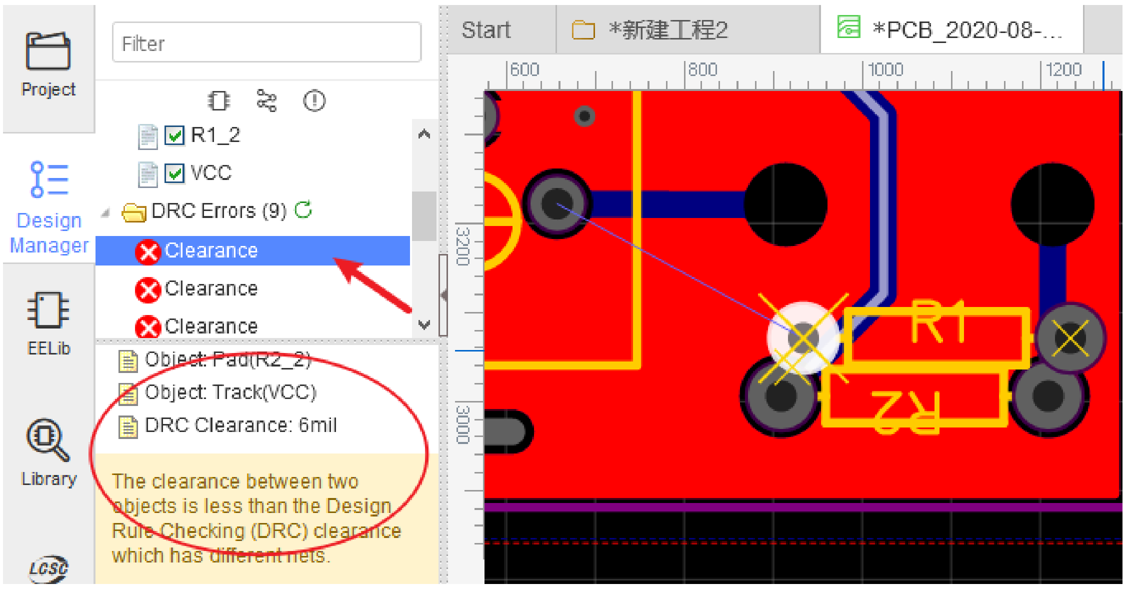
After checking, you can view all the error at the "DRC Error", click the error the related objects will be highlighted.



The clearance between two objects is less than the Design Rule Checking (DRC) clearance which has different nets.

DRC error type

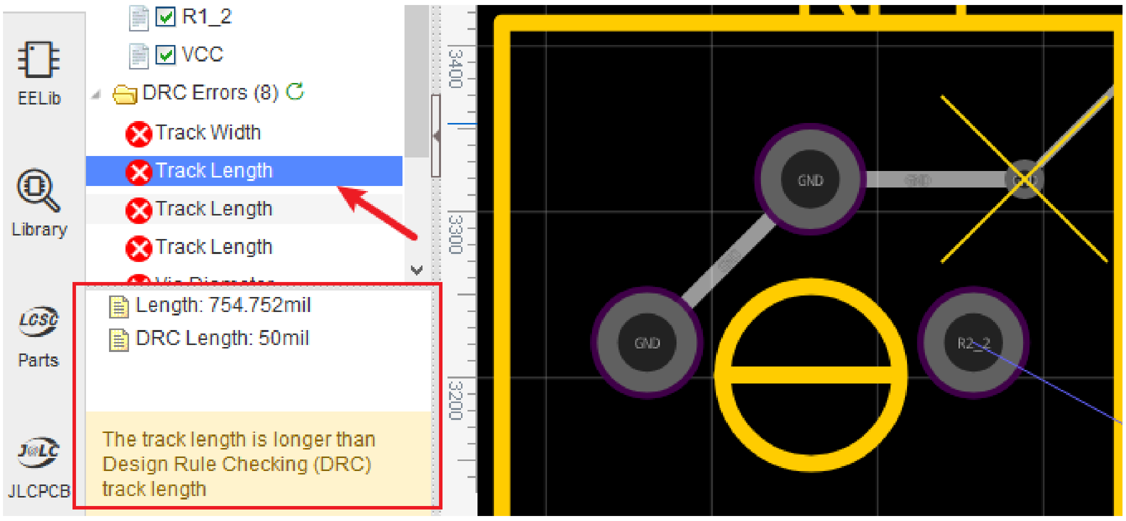
- Clearance: Object to Object. If two different net objects too close, and the distance less than the Design Rule clearance, it will show the Clrearance error.



Object: Pad(R2_2)
Object: Track(VCC)
DRC Clearance: 6mil

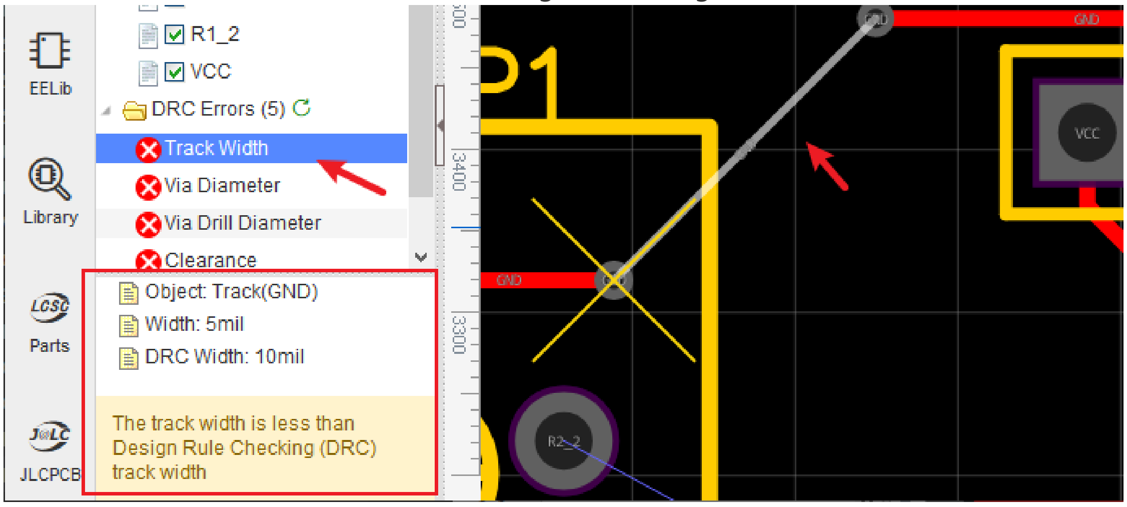
The clearance between two objects is less than the Design Rule Checking (DRC) clearance which has different nets.

- Track Length: The track Length of the all same net tracks must less than Design Rule track Length.



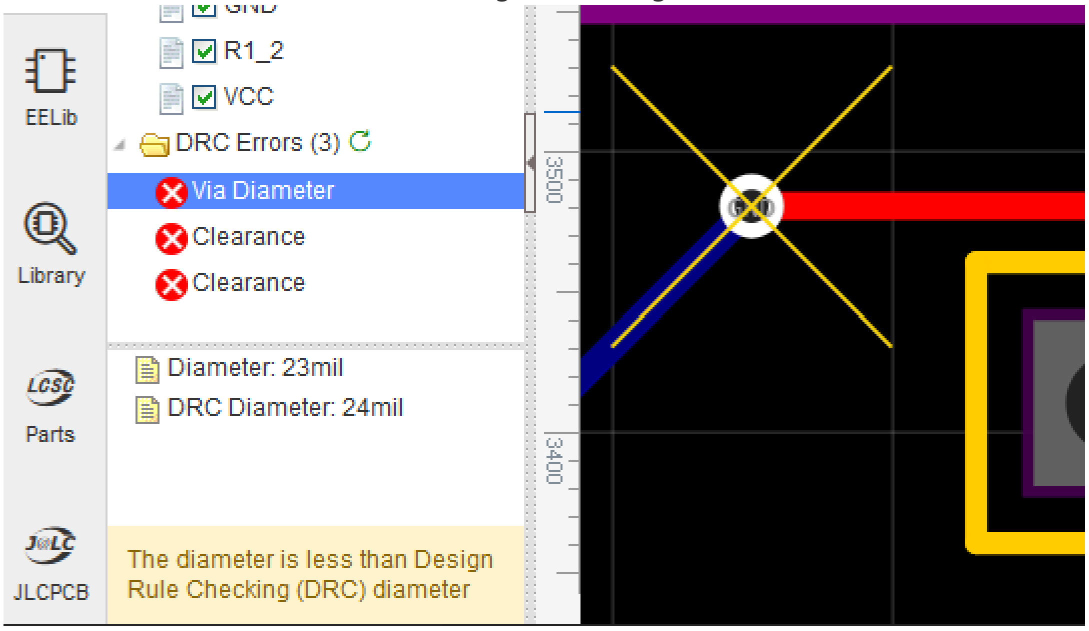
Length: 754.752mil
DRC Length: 50mil
The track length is longer than Design Rule Checking (DRC) track length

- Track Width: The track width must must large than Design Rule track width.



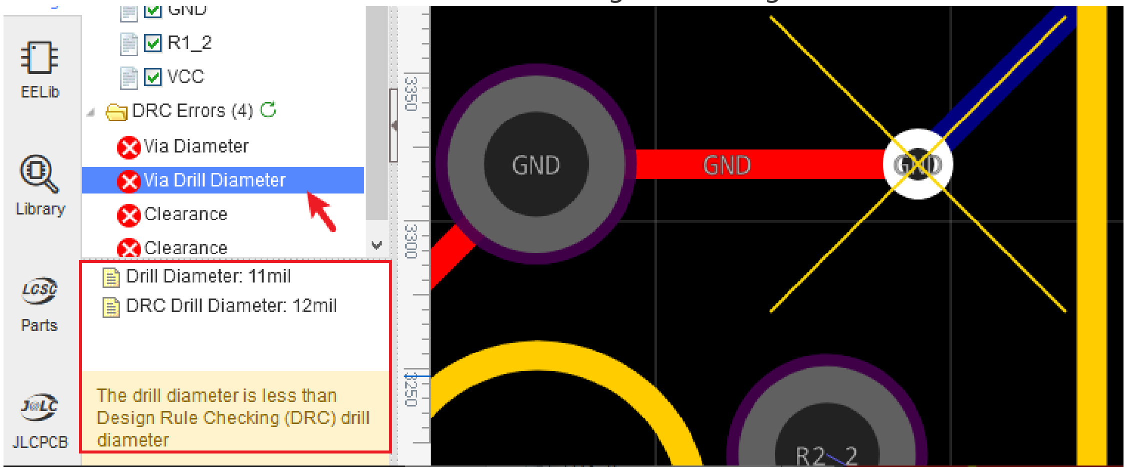
Object: Track(GND)
Width: 5mil
DRC Width: 10mil
The track width is less than Design Rule Checking (DRC) track width

- Via Diameter: The via diameter must large than Design Rule diameter.



Diameter: 23mil
DRC Diameter: 24mil
The diameter is less than Design Rule Checking (DRC) diameter

- Via Drill Diameter: The via drill diameter must large than Design Rule drill diameter.



Drill Diameter: 11mil
DRC Drill Diameter: 12mil

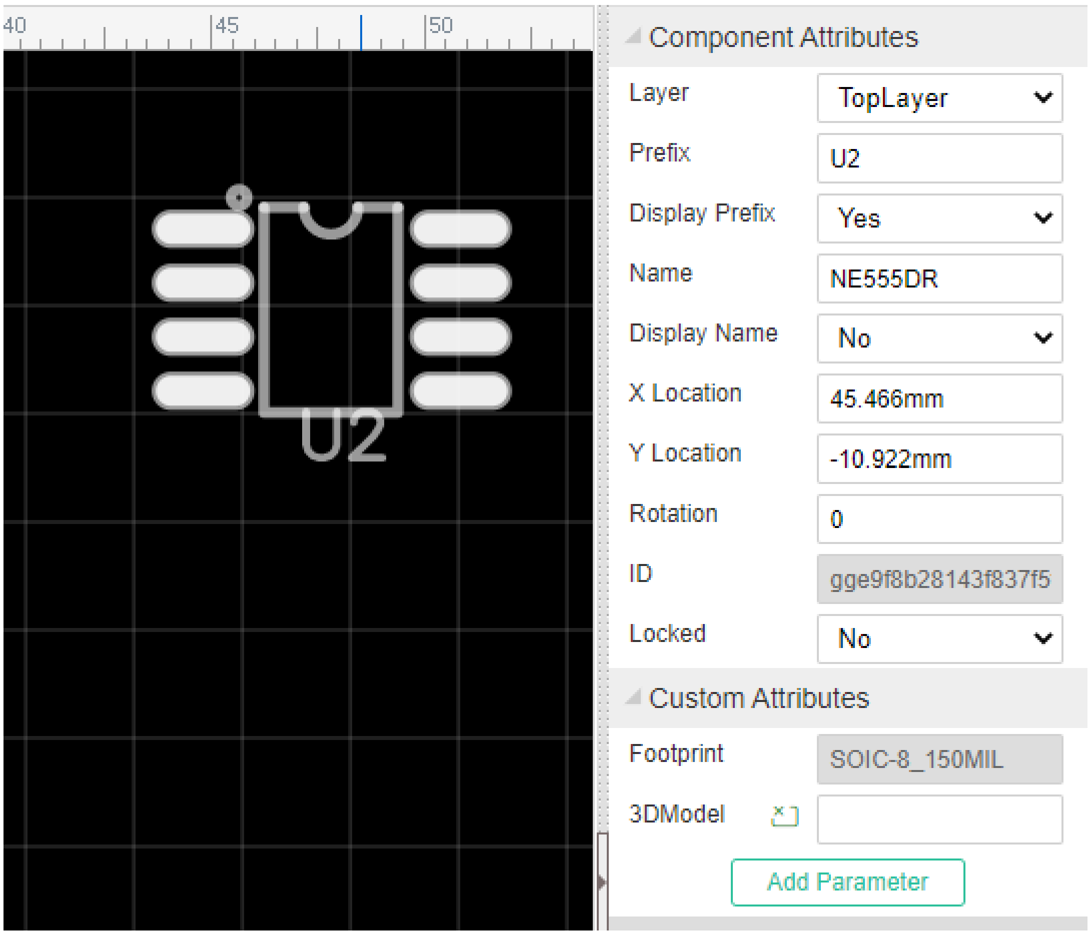
The drill diameter is less than Design Rule Checking (DRC) drill diameter

Note:

- *When you convert a schematic to PCB, the real time DRC is enable. But in the old PCB, the real time DRC is disable. you can enable it in the image as above.*
- *Design rule checking can only help you find some obvious errors.*
- *The color of the DRC error can be set in the layer manager.*

Footprint Attributes

When selecting a Footprint, you can find its attributes at the right hand Properties panel.



Component Attributes

Layer	TopLayer
Prefix	U2
Display Prefix	Yes
Name	NE555DR
Display Name	No
X Location	45.466mm
Y Location	-10.922mm
Rotation	0
ID	gge9f8b28143f837f5
Locked	No

Custom Attributes

Footprint	SOIC-8_150MIL
3DModel	

Prefix: It is same as the schematic. If you move the prefix too far away from the footprint, it will be dragged back to the footprint when you open the PCB again, if you don't need the prefix please set the prefix display as No.

Layer: You can set a footrpint to be on the TopLayer or BottomLayer, it same as board side. *Note: The footprint mirrors when it swapping layers. it doesn't support to mirror at current layer.**

X-Location and Y-Location: Moves the origin of the footprint to a precise position.

Rotation: Rotates the footprint about its origin over the range from 0o to any angle in 1o steps (visually of course multiples of 360o will appear identical).

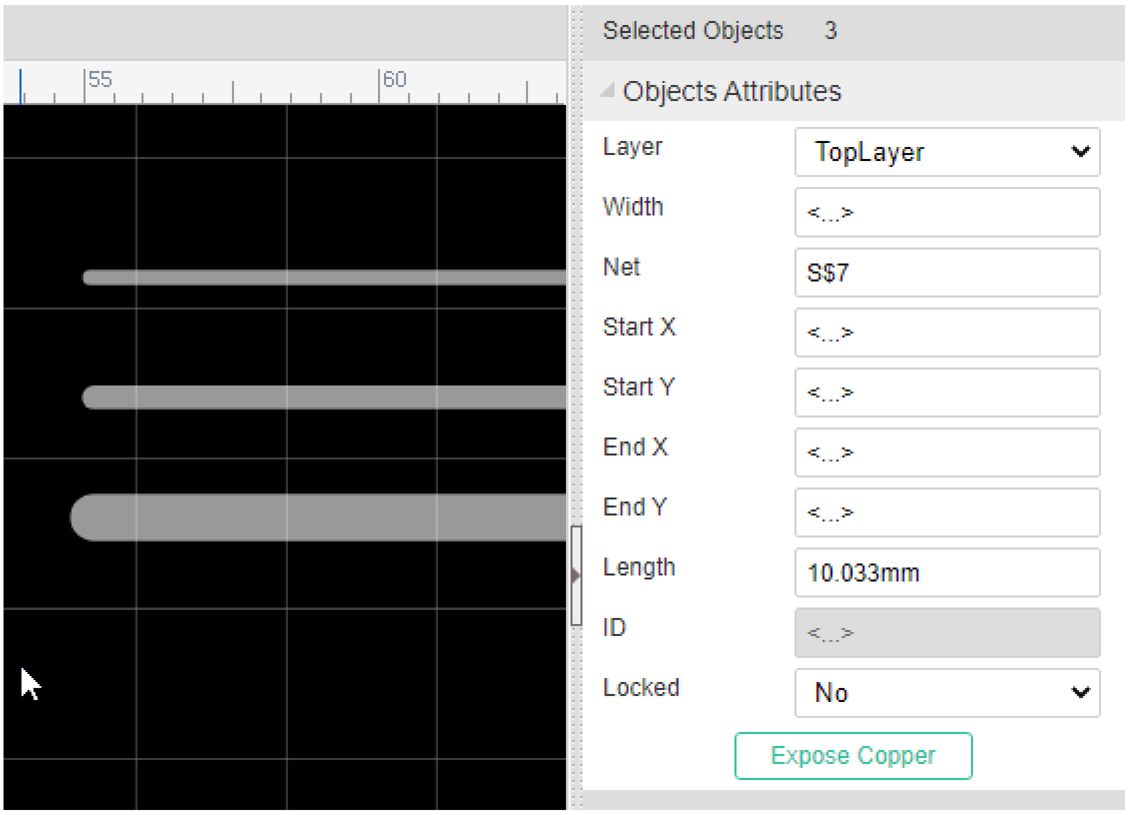
ID: EasyEDA will assign a unique ID for each footprint automatically, you can't modify it.

Lock: when locked the object, you can not move it by mouse, but if you set the X Y at the right-hand property panel, the object still moves, and the Locked will lock the object layer too.

Change Attributes in Batch on PCB Editor

Sometimes, we need to change some attributes of multiple objects together, such as the track width, hole size and font size.
Now, you can select them and do some changes.

Taking the track for an example. If you select 3 tracks, now you can change their `Width`, `Layer`, `Net` together. The difference property values will combine as `<...>`, change it directly will apply to all seleted objects.



Selected Objects 3

Objects Attributes

Layer	TopLayer
Width	<...>
Net	S$7
Start X	<...>
Start Y	<...>
End X	<...>
End Y	<...>
Length	10.033mm
ID	<...>
Locked	No

Design Manager

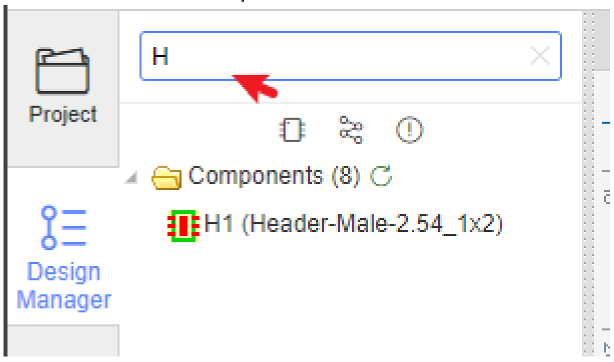
Just like Schematic's Design Manager, PCB's Design Manager can be found via:

Left Navigation panel > Design

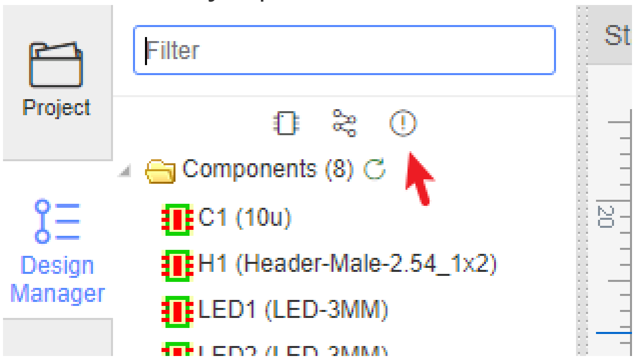
or just press the `CTRL+D` hotkey to open the Design Manager dialog.

Design Manager function:

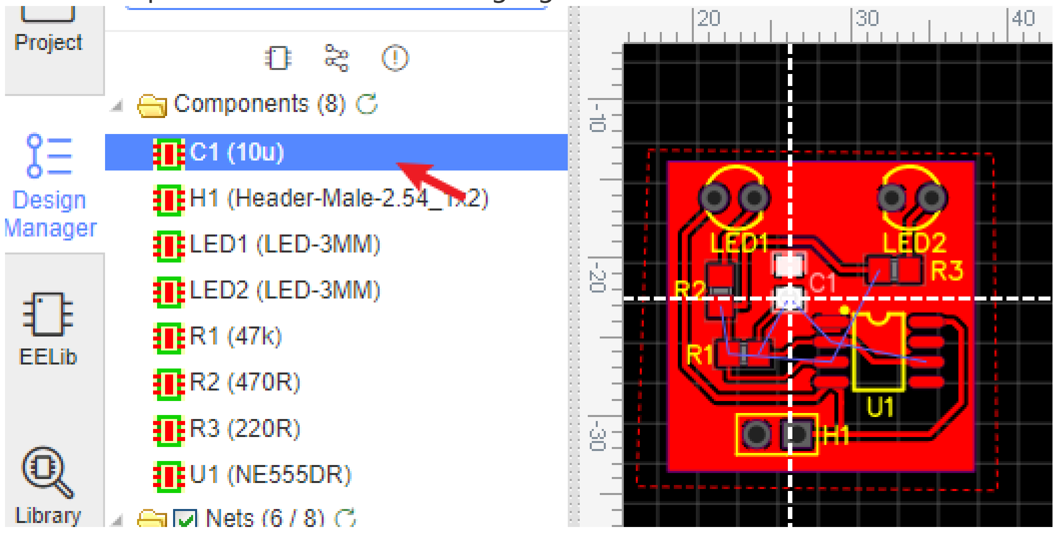
- Filter
    - Filter to find a component or net.



- Jump
    - Click the icon to jump to folder.

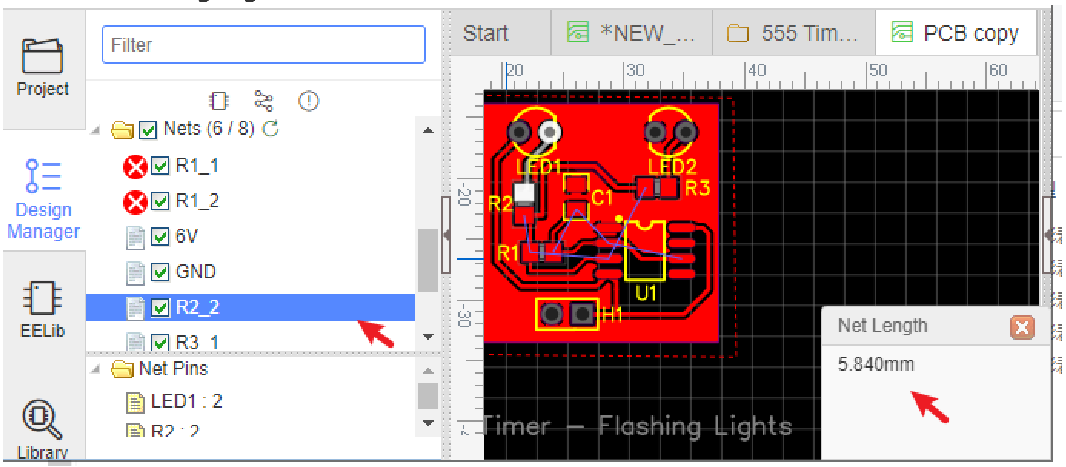


- Component
    - Click a component/Net/DRC Error to highlight it.

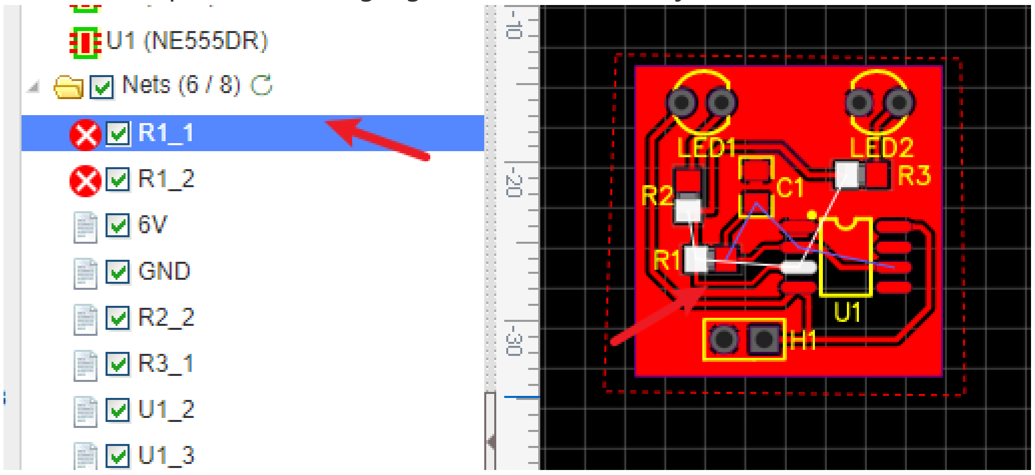


- Net

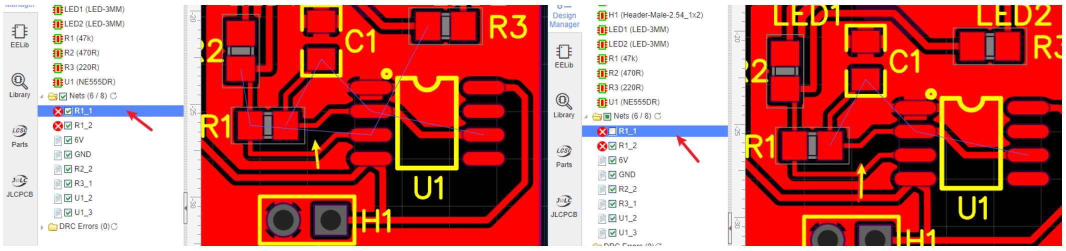
- Click a net to highlight the tracks/vias with the same net.



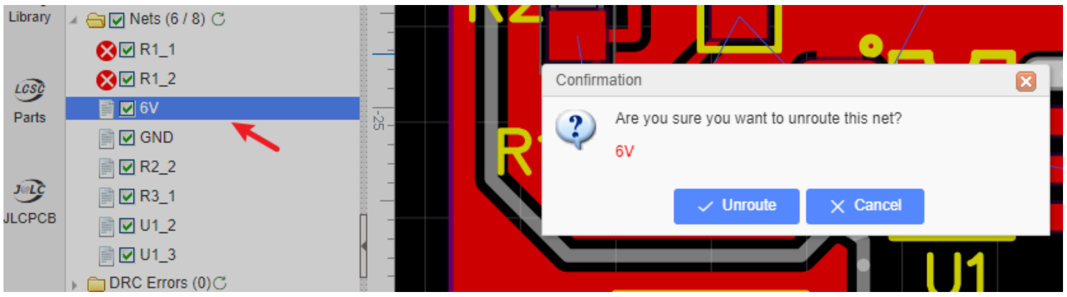
- Click a incomplete net will highlight the ratline and objects.



- Check/uncheck the net to show/hide the ratline of the net.



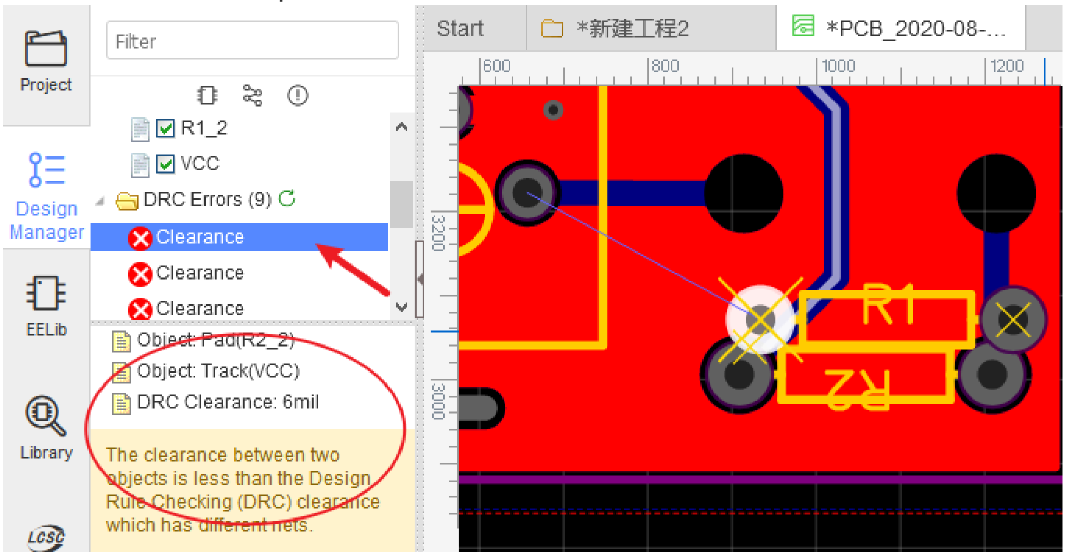
- Double click the net to remove all of the tracks and vias with the net name. If you want to reroute a net, this is the recommended method to use to un-route it first.



Are you sure you want to unroute this net?
6V

- DRC

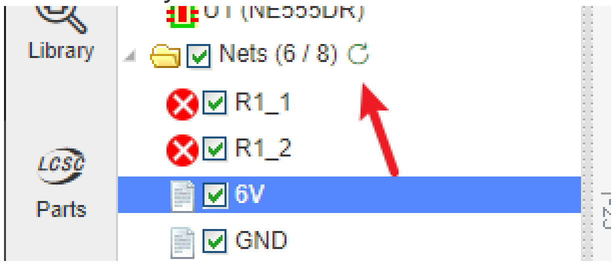
- Click the DRC list, will position the DRC mark on the canvas.



Object: Pad(R2_2)
Object: Track(VCC)
DRC Clearance: 6mil
The clearance between two objects is less than the Design Rule Checking (DRC) clearance which has different nets.

Notice:

- Design Manager list doesn't support to refresh automatically, you must click the refresh icon manually.



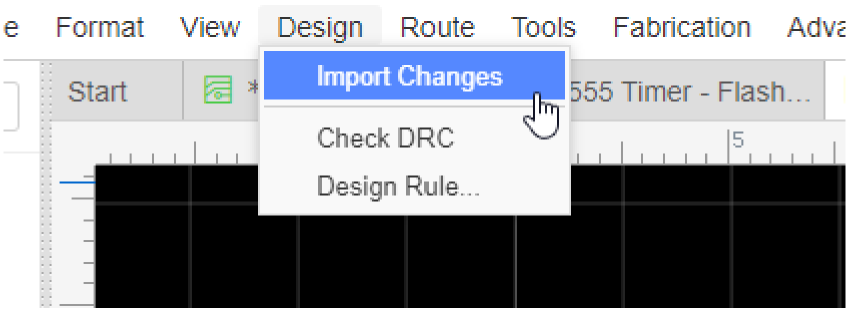
Import Changes

Import Changes

Sometimes, while working on a project, you need to make changes to the schematic and then update your board, to incorporate them.

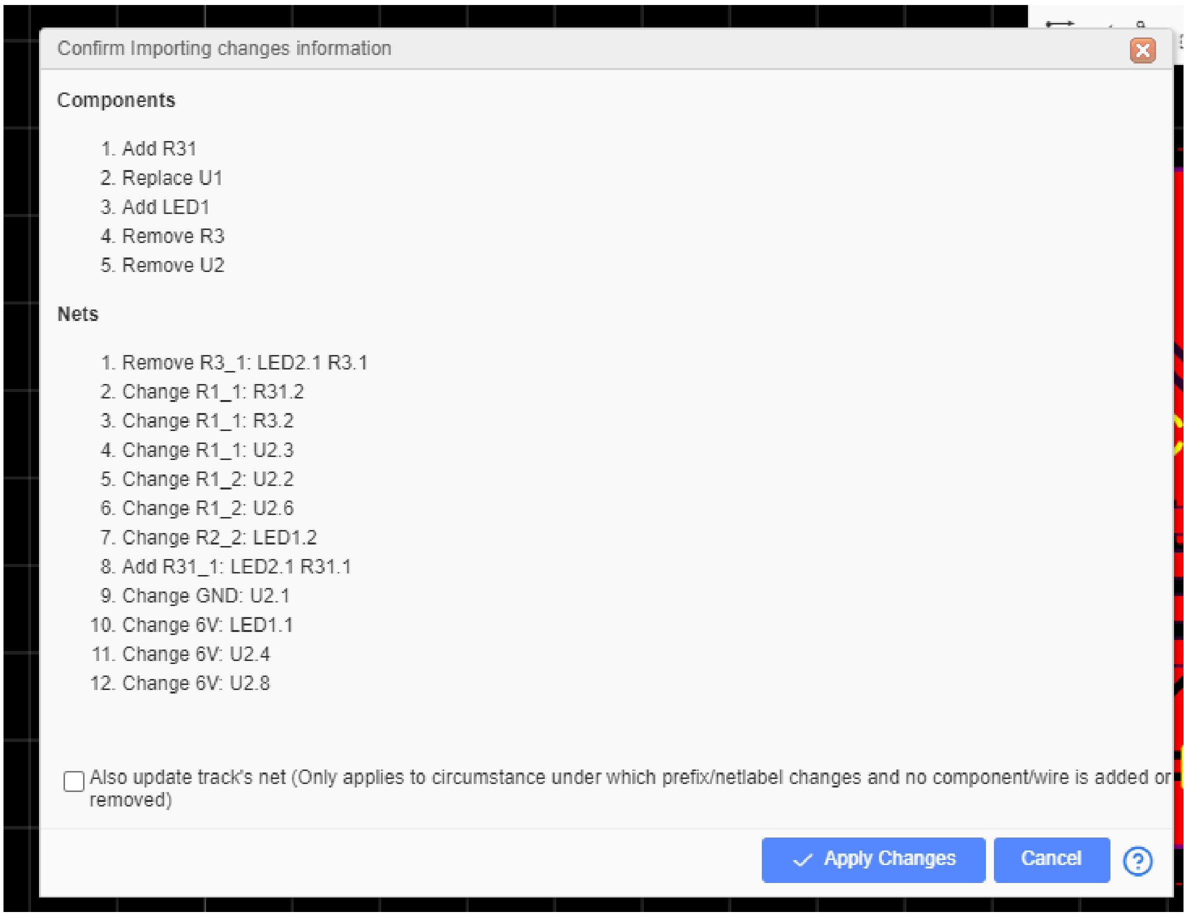
It's easy to do this with EasyEDA.

Go to the **PCB Editor**, via: **Top Meun** > **Design** > **Import Changes**



If there are some errors at schematic, such as prefix duplicated, no footprint, it will pop up notice dialog, the more information please refer： Schematic - Convert to PCB

If no errors, you will get a "Confirm Importing changes information" dialog:



Confirm Importing changes information

Components

1. Add R31
2. Replace U1
3. Add LED1
4. Remove R3
5. Remove U2

Nets

1. Remove R3_1: LED2.1 R3.1
2. Change R1_1: R31.2
3. Change R1_1: R3.2
4. Change R1_1: U2.3
5. Change R1_2: U2.2
6. Change R1_2: U2.6
7. Change R2_2: LED1.2
8. Add R31_1: LED2.1 R31.1
9. Change GND: U2.1
10. Change 6V: LED1.1
11. Change 6V: U2.4
12. Change 6V: U2.8

Also update track's net (Only applies to circumstance under which prefix/netlabel changes and no component/wire is added or removed)

If you are happy with your changes, just click the Apply Change button.

If you want to update the PCB tracks net same as the schemtiac, you need to enable "Also update track's net" option. The editor will update the related track's net depends on the pad's net.

The changes will then be passed into the PCB layout and you can then adjust the tracking to suit.

Notice:

- Because of the net of the schematic is generated after calculating, when you change some netlabel, after Import Changes, the PCB track will not be deleted.
- When enable the "Also update track's net" option, after Import Changes, the related tracks vias will update the net from the pads, there will be some nets changed isn't you want, you need to change them manually, such change prefix, modify the parts connection, delete or add part at the schematic, you can change the tracks net via: right-click the track - click Select menu - Connection, and them all connection will be seleted, you can change them net at the right-hand property panel.
- After Import Changes, there are some action can not be undo.

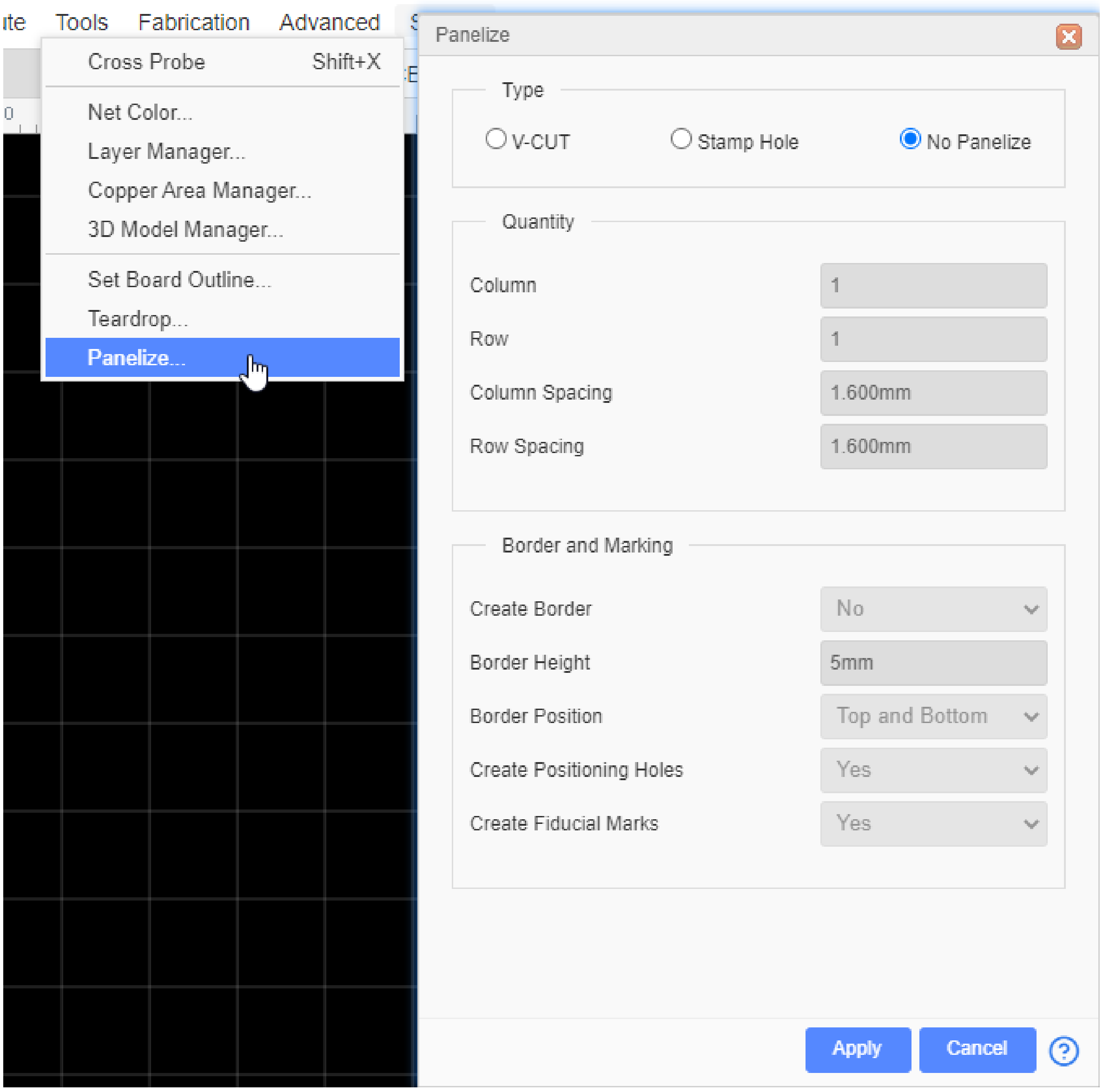
Panelize

via: Top Menu - Tools - Panelize

Panelize by Editor

At present, EasyEDA only support to panelize PCB itself, in order to decrease the file size, the panelized file only panelize the board outline.
Normally, all the PCB factory will support this panelized file, if you not sure, you need to contact your PCB factory support.

via: **Top Menu - Tools - Panelize**



Panelize

Type: V-CUT / Stamp Hole / [✓ No Panelize]

Quantity

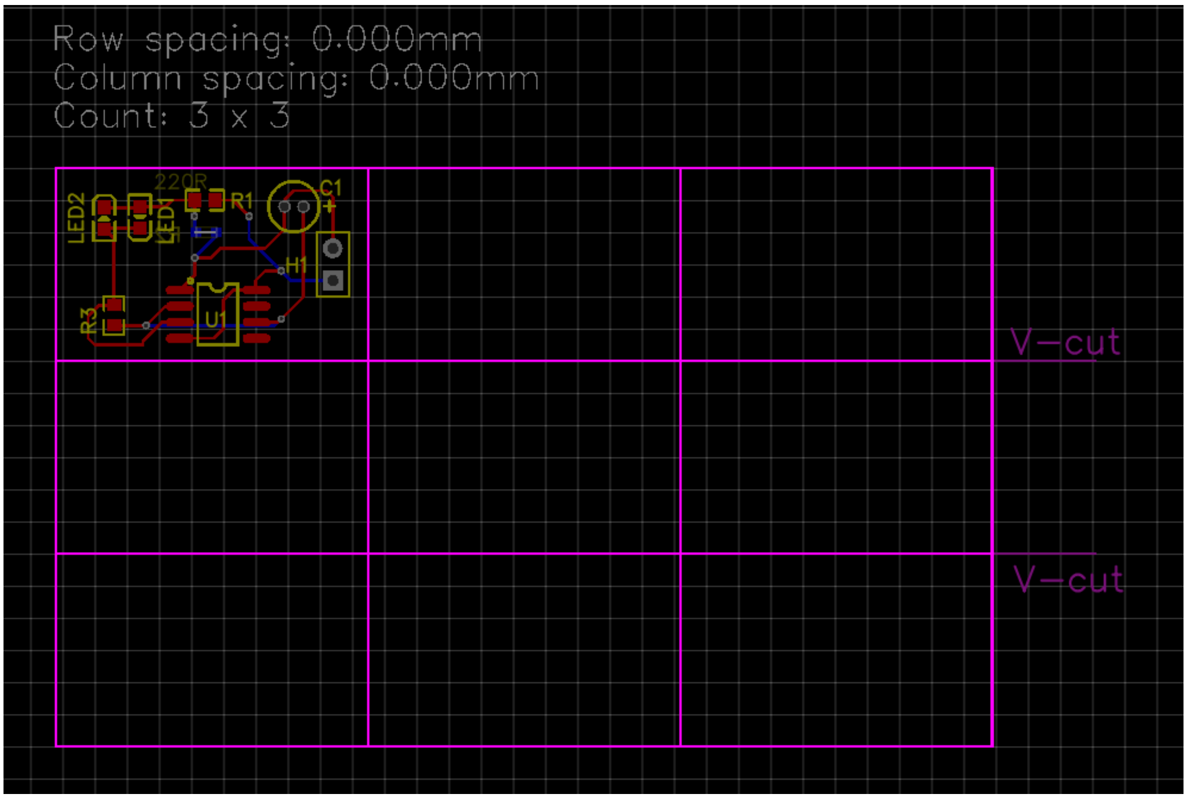
Column	1
Row	1
Column Spacing	1.600mm
Row Spacing	1.600mm

Border and Marking

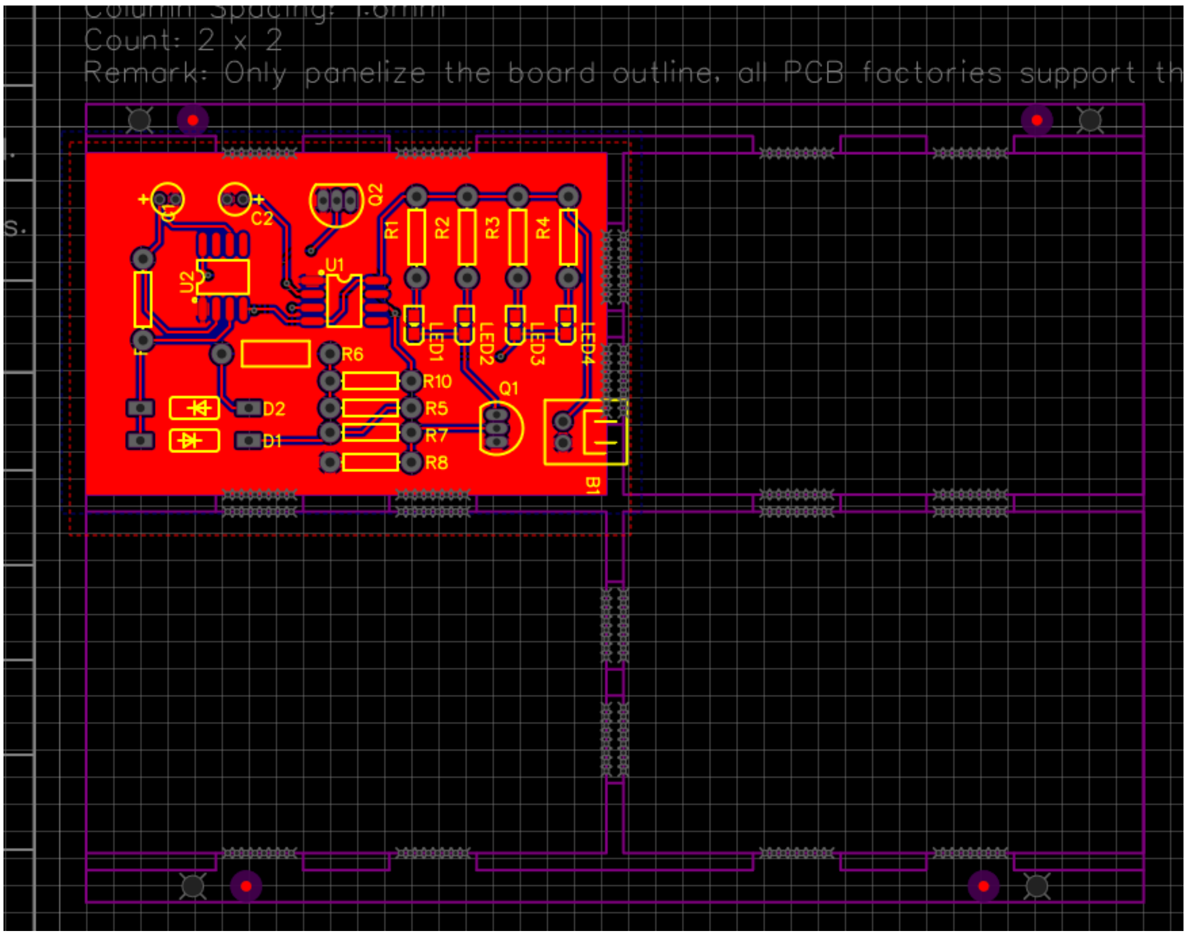
Create Border	No
Border Height	5mm
Border Position	Top and Bottom
Create Positioning Holes	Yes
Create Fiducial Marks	Yes

The Border height can not less then 3mm.

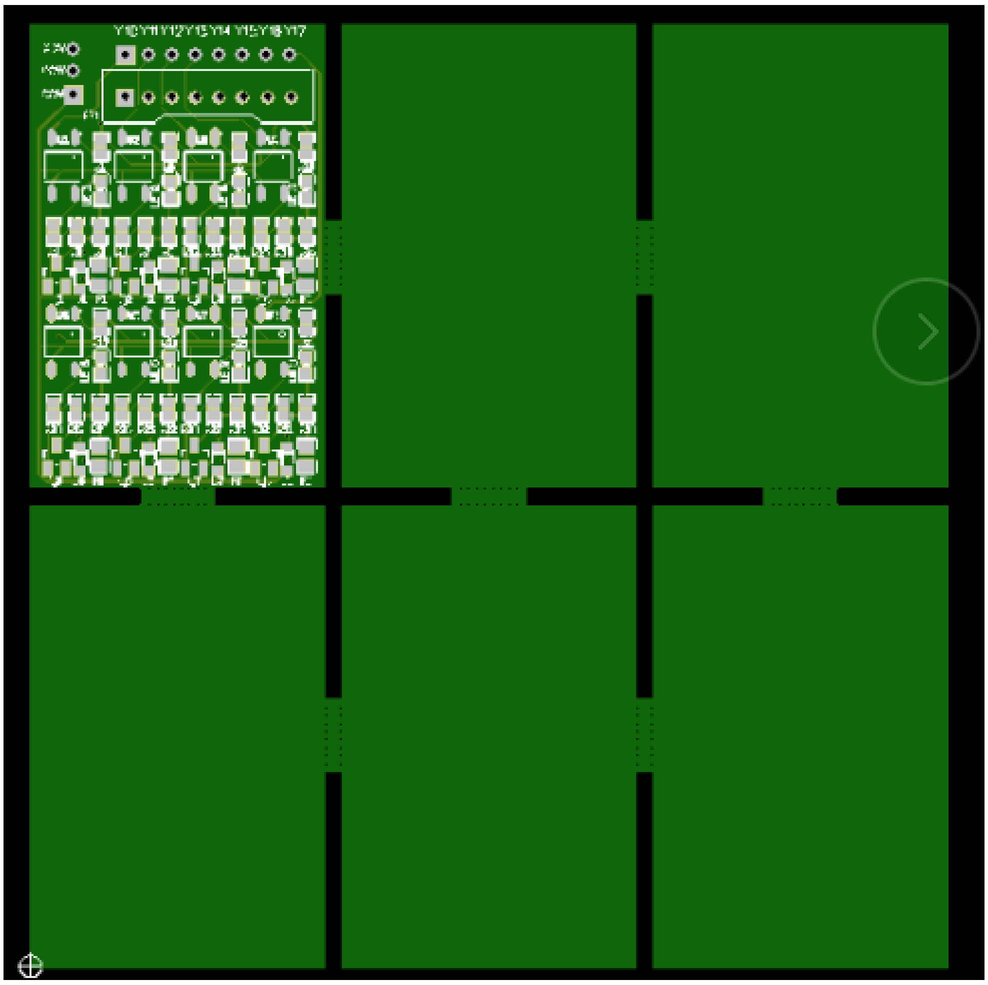
V-cut:
If you choose V-Cut, the editor will add the v-cut indication track on Board Outline layer at the Gerber.


Row spacing: 0.000mm
Column spacing: 0.000mm
Count: 3 x 3

Stamp Hole:


Column Spacing: 1.0mm
Count: 2 x 2
Remark: Only panelize the board outline, all PCB factories support th

When you preview the Panelize Gerber at JLCPCB.com, you will get the image like below:



JLCPCB will take care of your design, they know how to do.

Panelize by Manually

Process:

1. Select the whole board, hotkey `CTRL+A`.
2. Copy the whole board by reference point, hotkey `CTRL+SHIFT+C` or `CTRL+C`. You can only copy and paste the board outline to become the panelize board.
3. Paste the board via hotkey `CTRL+SHIFT+V`, this hotkey will keep the prefix and hide the ratline layer.
4. Paste repeatly, after finish, rebuild the copper area with `SHIFT+B`, recommend draw copper area at the end.

Notice

- If the board contains plane layer, it can not be panelized by manually, it will not generate the plane zone as you want.

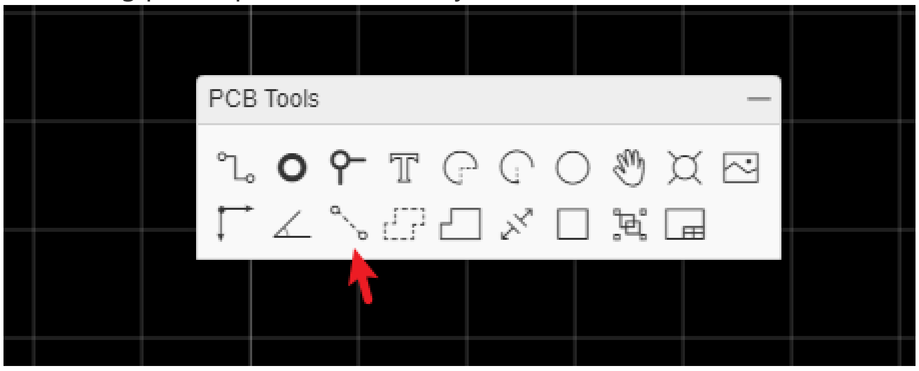
Layout a PCB Without Schematic

For some small PCB projects, maybe you don't need a schematic. EasyEDA allows you to lay the PCB directly from the PCB Editor.

1. Start a new PCB
2. add footprints directly from the Footprints from Left Navigation Panel **Library - Footprint**
3. and then just route track for them.

The PCB created by New PCB menu directly, it will hide the ratline layer defaultly.

For setting pad to pad connections, you can check the above **Connect Pad to Pad** section.

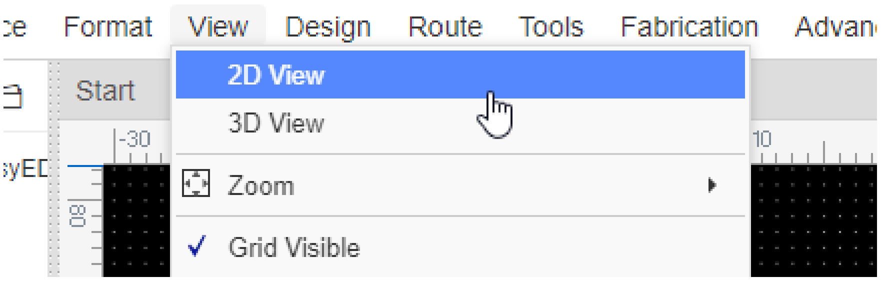


PCB Preview

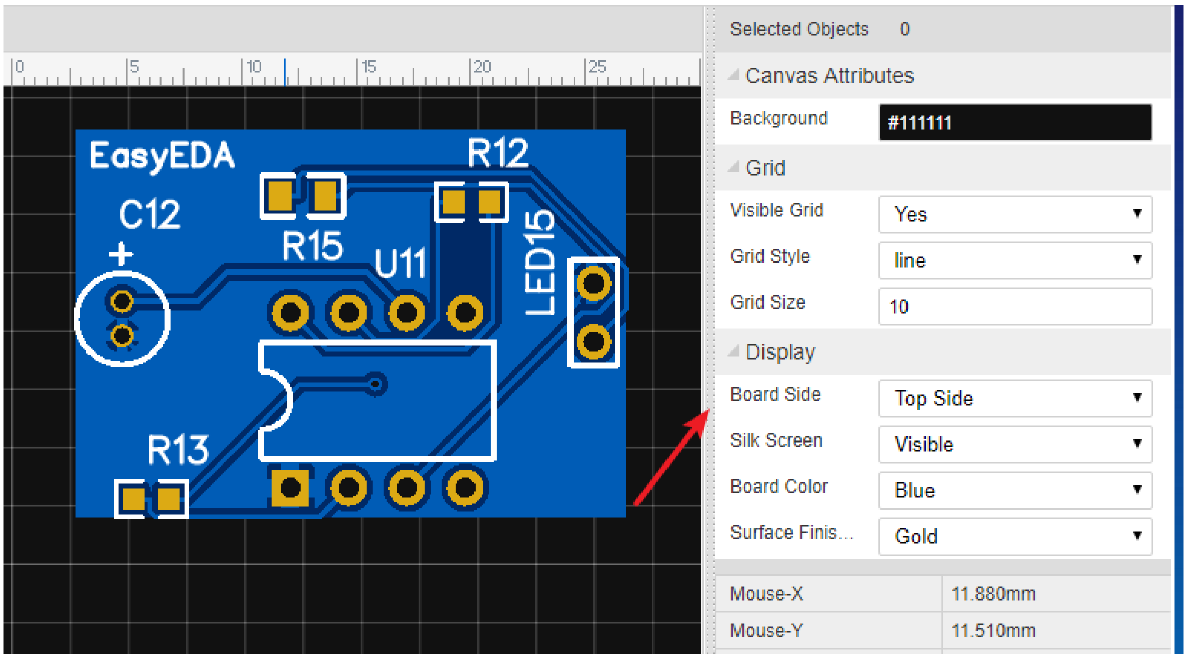
2D View

EasyEDA provide a nice Photo View to help you to check the PCB.

Via: Top Menu - View - 2D View.



After converting the PCB to Photo View, you can see the result as in the image below.


Selected Objects 0

Canvas Attributes

Background	#111111

Grid

Visible Grid	Yes
Grid Style	line
Grid Size	10

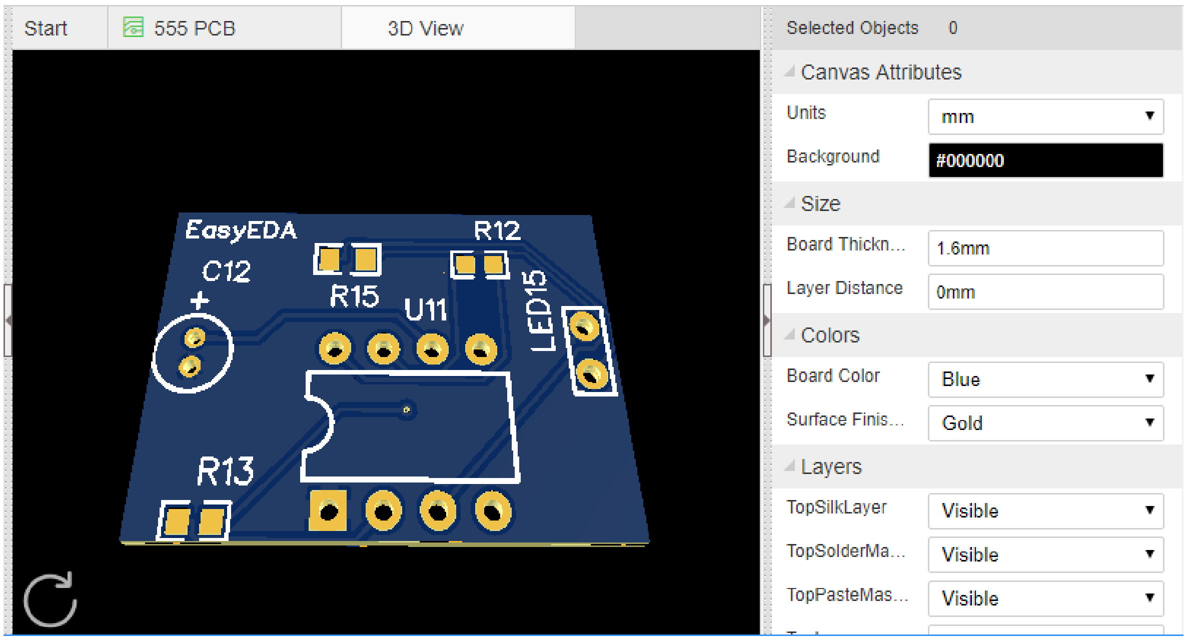
Display

Board Side	Top Side
Silk Screen	Visible
Board Color	Blue
Surface Finis...	Gold

Mouse-X	11.880mm
Mouse-Y	11.510mm

3D View

After click 3D view menu, the server will generate the 3D view file, when the editor loading finish, you will see a pretty cool 3D view.



Selected Objects 0

Canvas Attributes

Units	mm
Background	#000000

Size

Board Thickn...	1.6mm
Layer Distance	0mm

Colors

Board Color	Blue
Surface Finis...	Gold

Layers

TopSilkLayer	Visible
TopSolderMa...	Visible
TopPasteMas...	Visible

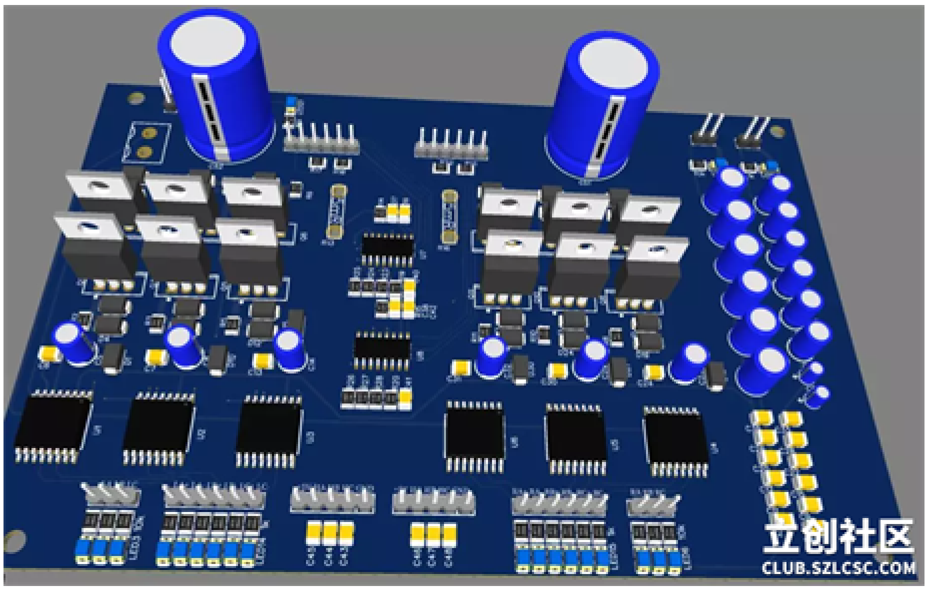
- Change 3D view attributes at the right-hand panel;
- Reset the 3D PCB position at the left-bottom corner icon;
- Keep left-click and drag the canvas can change the view direction;
- Keep right-click and pan can change the 3D PCB position.

3D model view of the component please check "PCB - 3D Model Manager" and "Footprint - Import 3D Model" chapter.

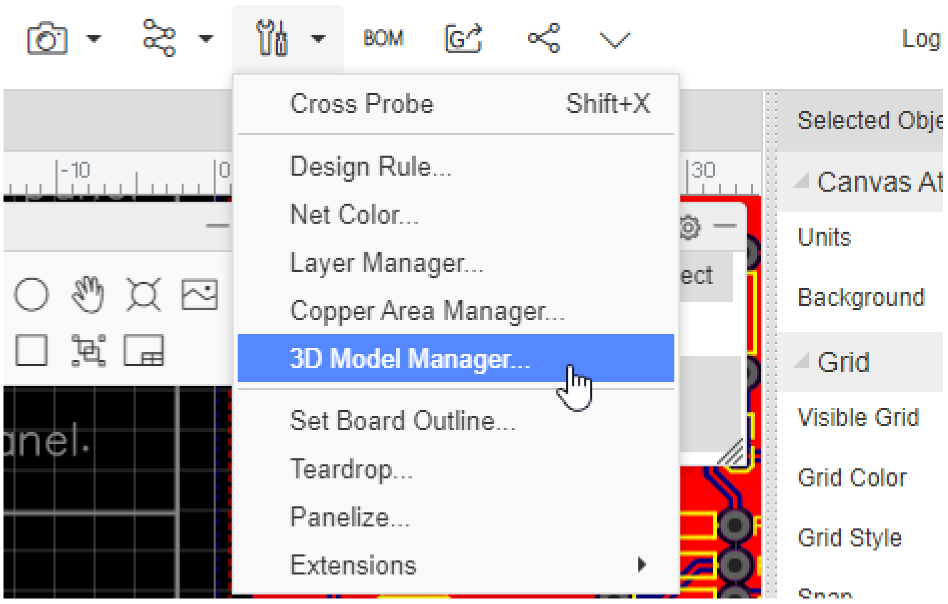
3D Model Manager

3D Model Manager

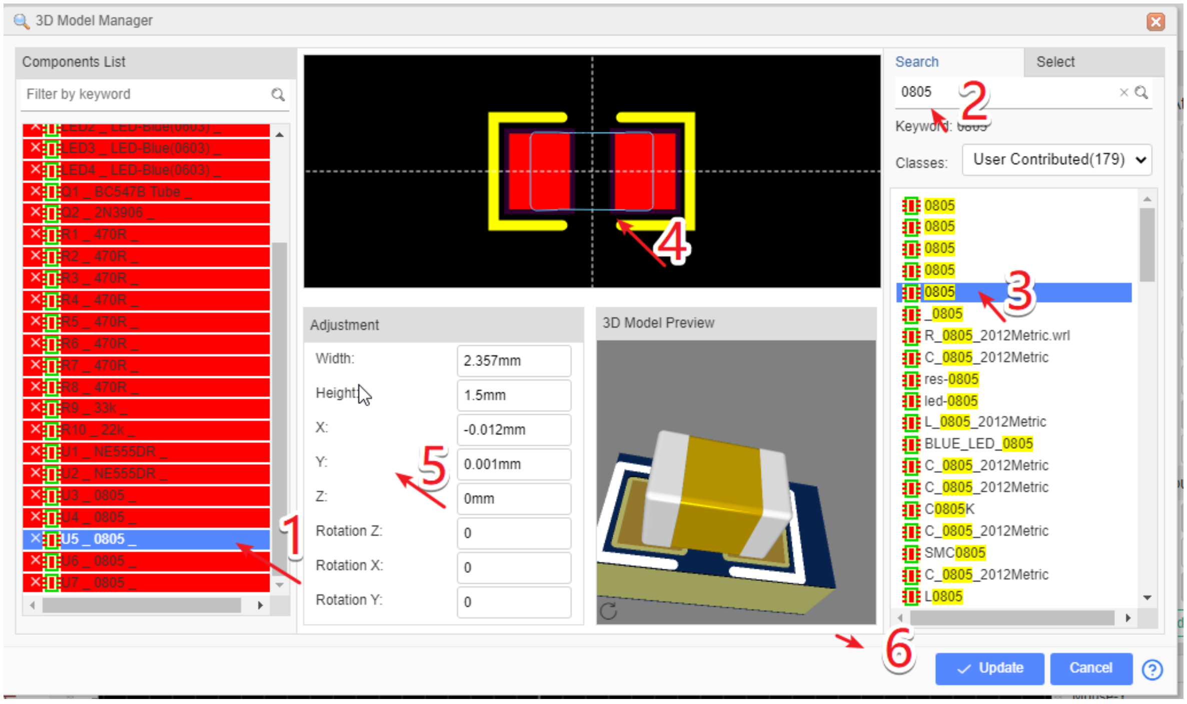
EasyEDA supports for importing 3D models, PCB can view cool 3D models when doing 3D preview. Exporting PCB 3D model files is not supported yet.



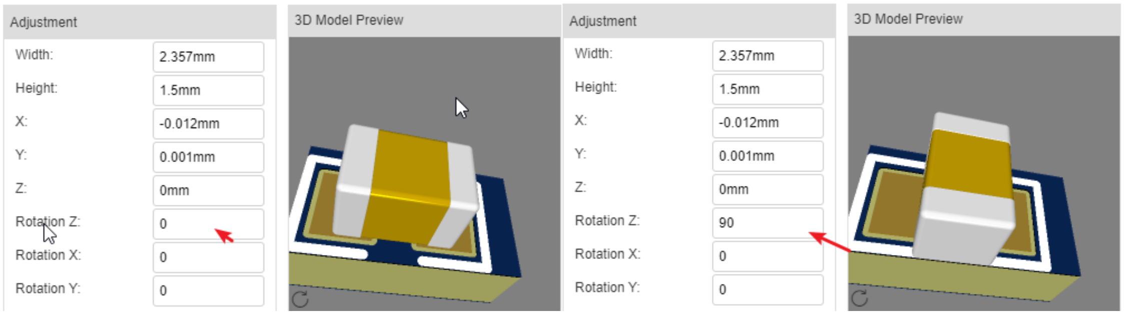
Open 3D model manager: - tools - 3D model manager



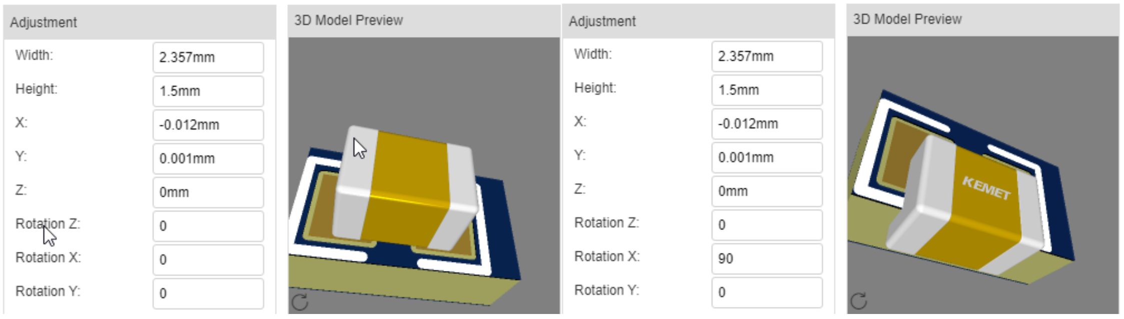
When you open it, you can bind 3D models for Footprint, and you can import or search for user-contributed 3D models. Import tutorial please see: [Footprint - Import 3D Model]



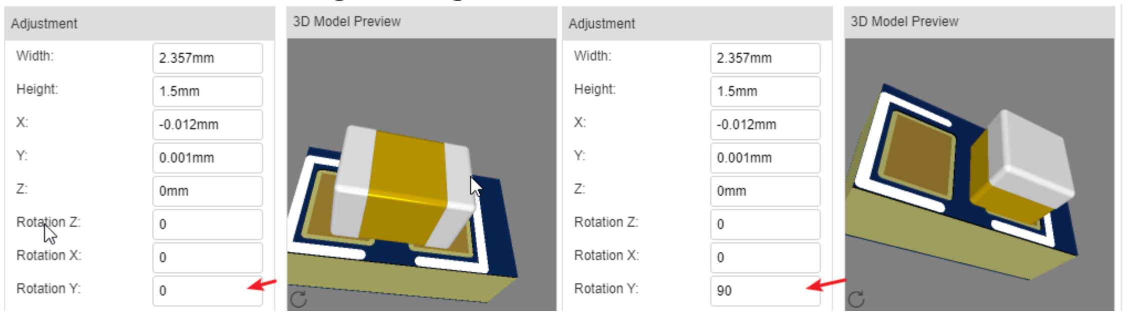
1. Click the footprint in the list of footprints on the left, and the preview of footprints will be displayed in the central area. Support multiple selection: hold down CTRL + Mouse click selection; Hold SHIFT + click select.
2. Select the imported 3D models from the list on the right or directly search the 3D model uploaded by the user, and search by keyword.
3. Select a 3D library from the list of 3D model libraries and display 2D outline of the top view of the 3D model in the middle preview area.
4. Adjust parameters:
   drag the mouse to align the 2D boundary line with the footprint shape; You can also adjust the way you enter parameters below. The right side of the parameters can be previewed directly, and it supports a long left - click drag 3D preview interface.
   width/height: the width/height of the 2D shape of the 3D model
   X and Y: the X and Y coordinates of the 2D shape of the 3D model
   z-axis rotation: in the overhead view, the 3D model rotates anticlockwise. The editor automatically recognizes the width and height of the footprint and automatically sets the z-axis rotation to 90 degrees.



X-axis rotation: in the side view, the 3D model rotates counterclockwise



Y-axis rotation: when facing the image, the 3D model rotates anticlockwise



Adjustment

Width:	2.357mm
Height:	1.5mm
X:	-0.012mm
Y:	0.001mm
Z:	0mm
Rotation Z:	0
Rotation X:	0
Rotation Y:	0

Adjustment

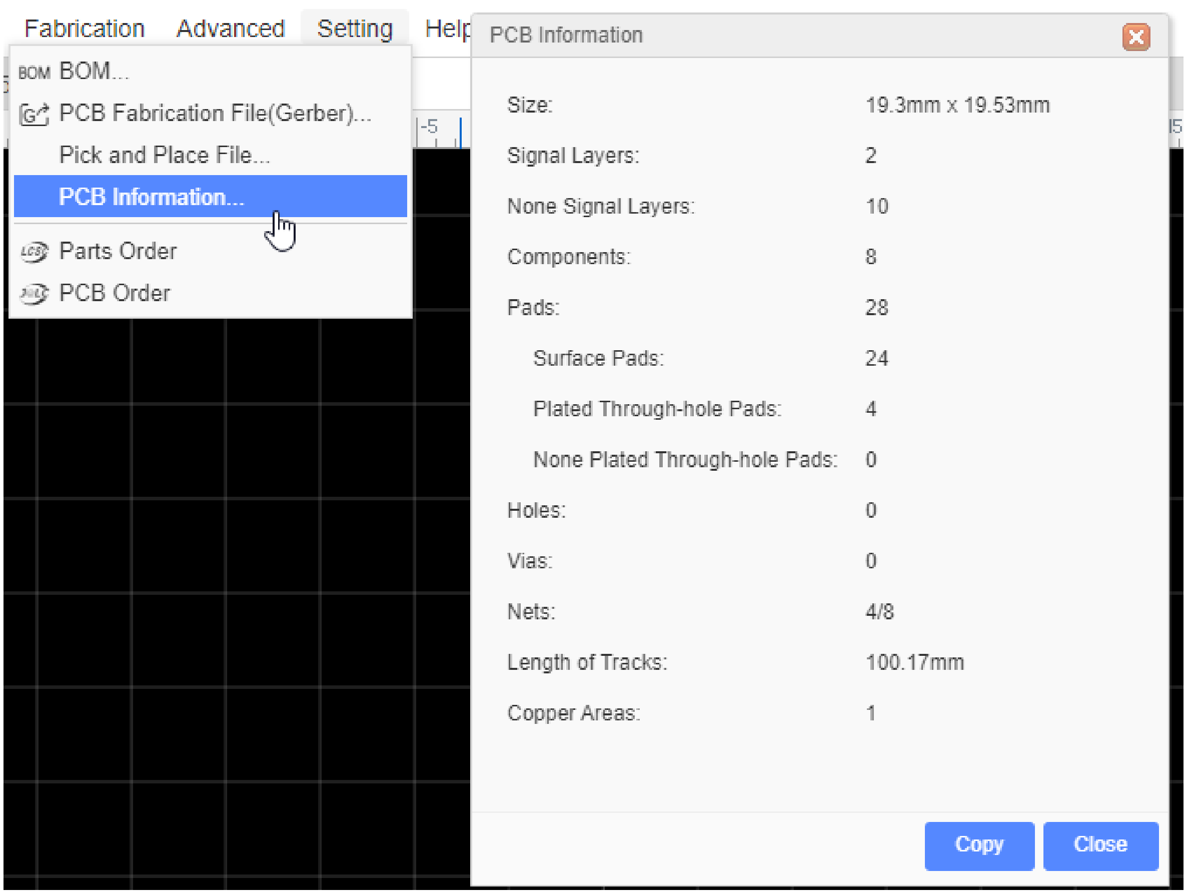
Width:	2.357mm
Height:	1.5mm
X:	-0.012mm
Y:	0.001mm
Z:	0mm
Rotation Z:	0
Rotation X:	0
Rotation Y:	90

5. After the adjustment, click the "update" button to complete the 3D model binding.
6. Click 3D preview from the preview menu at the top to preview the 3D model.

PCB Information

PCB design information can be easily obtained by checking PCB information.

Entry: Top Menu - Fabrication - PCB Information



PCB Information

Size:	19.3mm x 19.53mm
Signal Layers:	2
None Signal Layers:	10
Components:	8
Pads:	28
Surface Pads:	24
Plated Through-hole Pads:	4
None Plated Through-hole Pads:	0
Holes:	0
Vias:	0
Nets:	4/8
Length of Tracks:	100.17mm
Copper Areas:	1

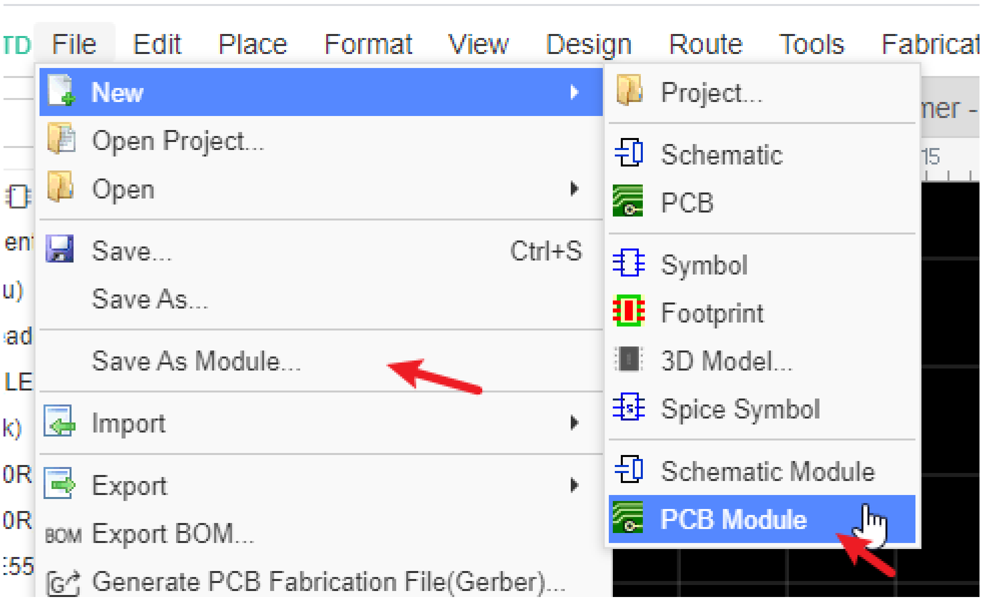
Nets shows: routed nets/total nets.

PCB Module

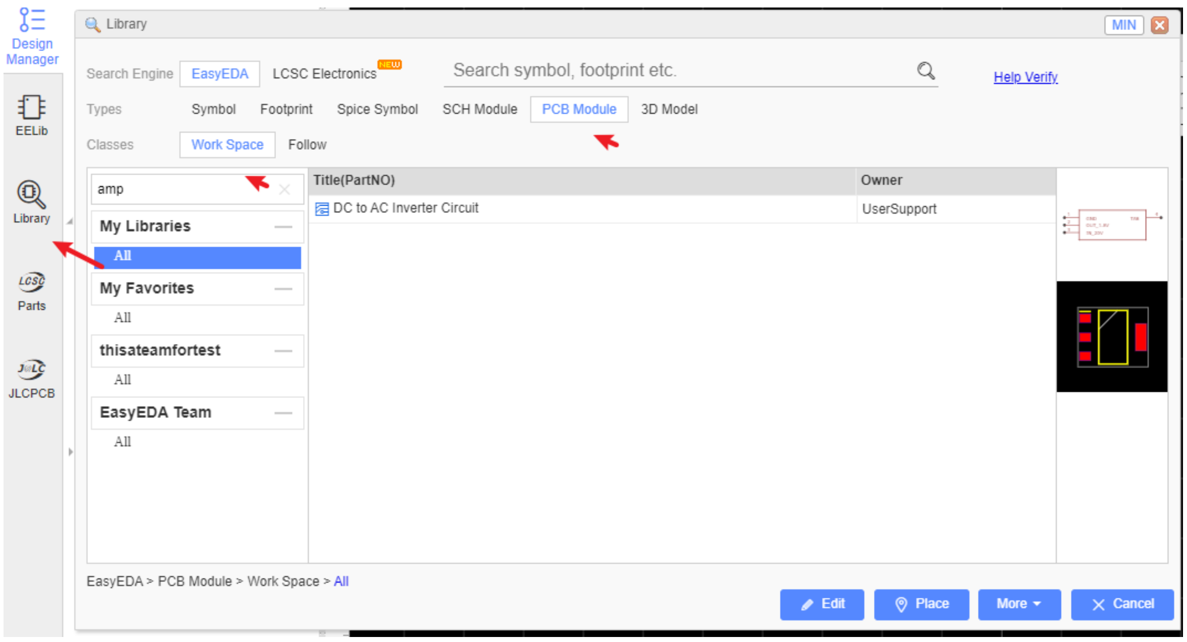
EasyEDA support create the PCB modules, it seems schematic module.

How to Create

Via: **Save as Module** and **File** > **New** > **Schematic/PCB Module**。



PCB module save at **Library** > **Schematic/PCB module** > **Work Space** > **My Libraries**



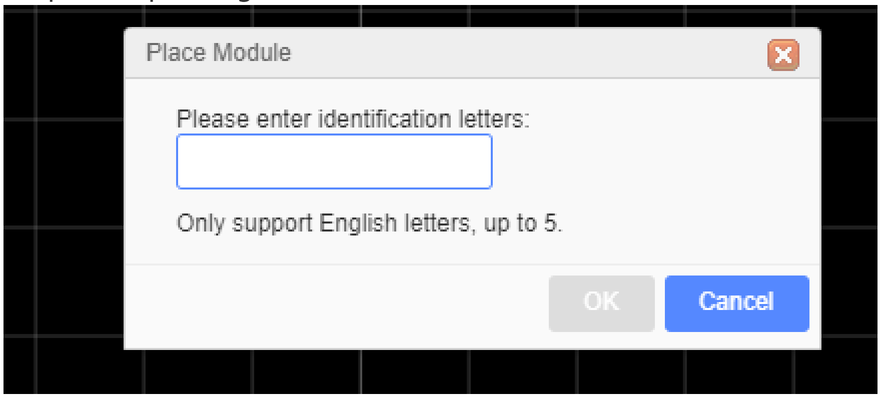
How to use

Since v6.4.3, after placing schematic modules and PCB modules, after Import Changes, supports to keep the layout location.

How to use:

1. Draw schematic modules and PCB modules, and ensure that their component prefix are one to one, and the footprint is also corresponding. The module's component prefix can not have question marks and duplicate prefix, such as U? or two R1.
2. Open schematic and PCB at a same project.
3. Open "Library", select the module.
4. Click the "Place" button to place the previous saved schematic module and PCB module.

5. It will pop up a window to enter English letter. The letter of schematic module should keep corresponding with PCB modules.



Place Module

Please enter identification letters:

Only support English letters, up to 5.

For example: A component at schematic module is U2, enter letter K, press OK to place into canvas, it will be KU2, then PCB module has KU2 too.
Click "OK" and enter the placement mode. After each placement, the pop-up will continue to enter the identification letter. Make sure that the identification letters entered each time are unique.

6. When finish the module place, the PCB component unique ID will same as Schematic component unique ID, then after Import Changes, the component's location will be keep. and you can update the track's net follow the schematic netlabel too.

That implement the multipe chanel placing.

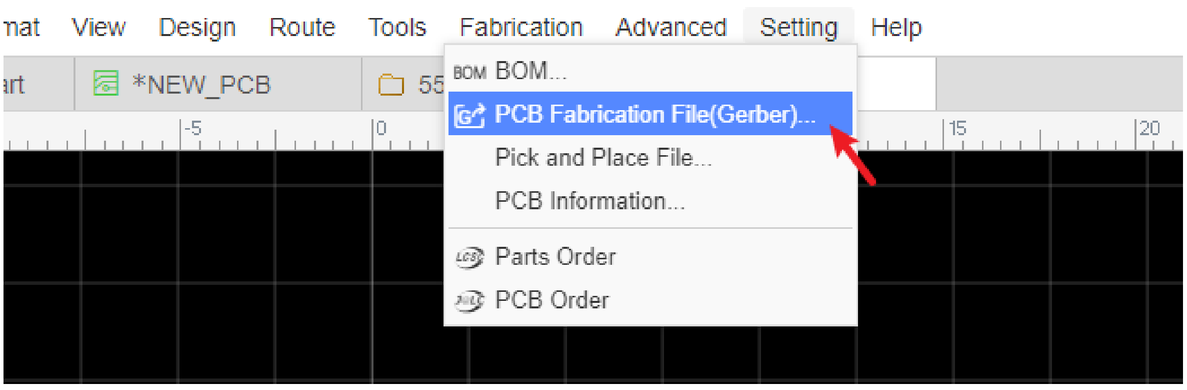
Notice:

- Module composes by tracks and components, it doesn't same as symbol binding footprint, the schematic module can not binding PCB module, after placing, the module will be separated by many objects, only the symbol and footprint can be corresponding via component ID, that is why you need to make the identication letter unique for placing each time to make sure schematic module corresponding with PCB module.

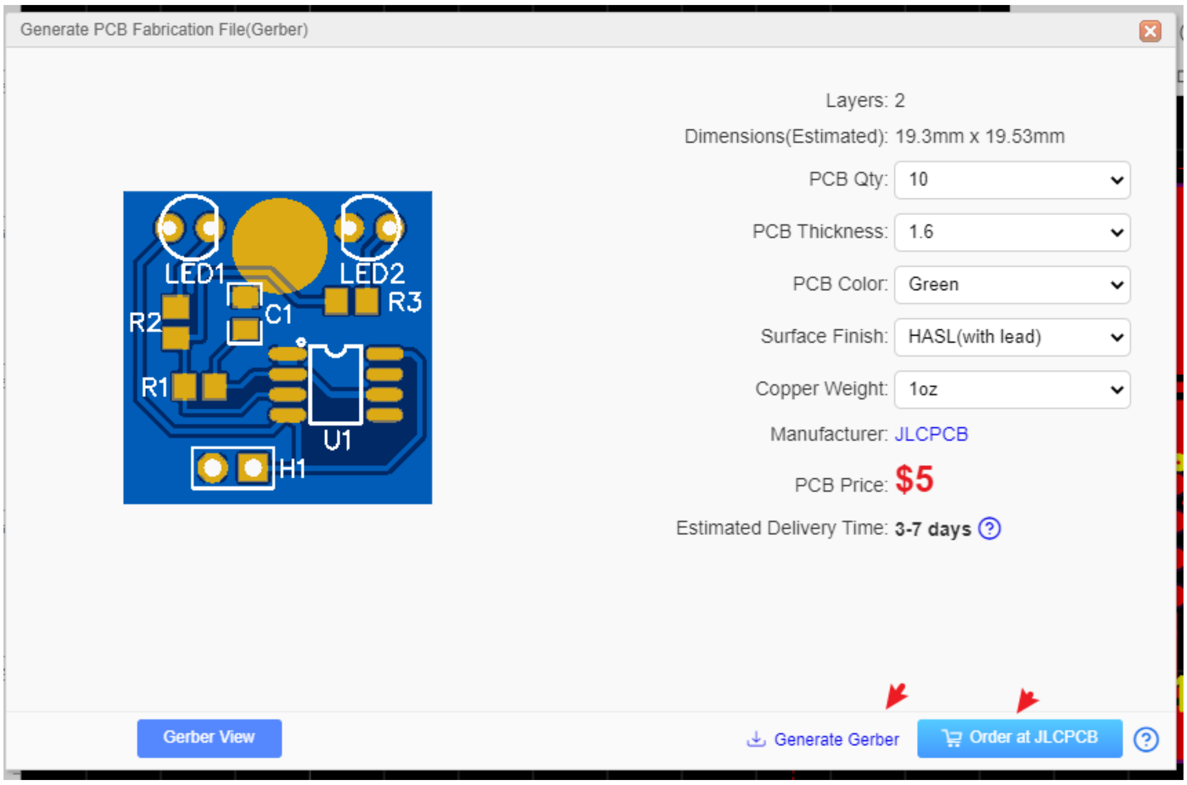
Generate Fabrication File(Gerber)

Generate Fabrication File Gerber

When you finish your PCB, you can output the Fabrication Files(gerber file) via: **File > Generate PCB Fabrication File(Gerber)** , or **Fabrication > PCB Fabrication File(Gerber)**.



After clicking, will open the Gerber generate dialog:


Generate PCB Fabrication File(Gerber)

Layers: 2
Dimensions(Estimated): 19.3mm x 19.53mm
PCB Qty: 10
PCB Thickness: 1.6
PCB Color: Green
Surface Finish: HASL(with lead)
Copper Weight: 1oz
Manufacturer: JLCPCB
PCB Price: $5
Estimated Delivery Time: 3-7 days

You can calculate the price for the PCB order, click SAVE to CART will go to JLCPCB and add your PCB in the cart.

Gerber file name

The generated Gerber file is a compressed zip file. After decompression, you can see the following files:

- **Gerber_BoardOutlineLayer.GKO**: PCB Border file. The PCB board factory cuts the shape of the board according to this document. The groove drawn by the EasyEDA, the solid region(Type: NPTH) is reflected in the border file after the Gerber is generated.
- **Gerber_TopLayer.GTL**: Top side copper layer.
- **Gerber_BottomLayer.GBL**: Bottom side copper layer.
- **Gerber_Inner1.G1**: Inner copper layer, signal type.
- **Gerber_Inner2.GP2**: Inner copper layer, plane type
- **Gerber_TopSilkLayer.GTO**: Top silkscreen.
- **Gerber_BottomSilkLayer.GBO**: Bottom silkscreen.
- **Gerber_TopSolderMaskLayer.GTS**: Top solder mask. The default board is covered with green oil, and the elements drawn on this layer correspond to the top layer's area will not be covered with oil.
- **Gerber_BottomSolderMaskLayer.GBS**: Bottom solder mask. The default board is covered with green oil, and the elements drawn on this layer correspond to the bottom layer's area will not be covered with oil.
- **Drill_PTH_Through.DRL**: Plated drill through hole layer. This document shows the location of the hole where the inner wall needs to be metallized. Old name: Gerber_Drill_PTH.DRL
- **Drill_NPTH_Through.DRL**: Non-Plated drill through hole layer. This document shows the location of the hole where the inner wall don't need to be metallized. Old name: Gerber_Drill_NPTH.DRL
- **Gerber_TopPasteMaskLayer.GTP**: Top Paste Mask, for the stencil.
- **Gerber_BottomPasteMaskLayer.GBP**: Bottom Paste Mask, for the stencil.
- **Gerber_TopAssemblyLayer.GTA**:Top Assembly, read only, doesn't affect the PCB manufacture. Old name: ReadOnly.TopAssembly

- **Gerber_BottomAssemblyLayer.GBA**: Bottom Assembly, read only, doesn't affect the PCB manufacture. Old name: ReadOnly.BottomAssembly
- **Gerber_MechanicalLayer.GML**: Record the information on the mechanical layer in the PCB design, and only use it for information recording. Old name: ReadOnly.Mechanical. By default, the shape of the layer is not manufactured at the time of production. Some board manufacturers use the mechanical layer to make the frame when using Altium file to production. When using Gerber file, it is only used for text identification in JLCPCB. For example: process parameters; V cut path etc. In EasyEDA, this layer does not affect the shape of the border of the board. If the mechanical layer has closed wires, JLCPCB will give priority to using the mechanical layer as the shape of the board when producing the board. If there is no outer frame of the mechanical layer, GKO will be used as the frame (historical influence of Altium file). It is necessary to pay attention to the use of the mechanical layer in the design.

Notice:

- *Before ordering the PCB, please check the gerber at the Gerber view as below.*
- *The Gerber files are generated by browser, please use the browser inner downloader to download!*
- *The coordinates of the Gerber file follow the canvas coordinates*
- *When exporting Gerber, the coordinate format accuracy defaults to 3:3. When the PCB size is out of range, it automatically uses 4:2 format. If you view the Gerber as such as CAM350, found that the Drill hole has been offset the location, you can modify the drill coordinate format to fit the location*

Gerber View

Before sending Gerber to the factory, please use gerber viewer to check the Gerber carefully.

local gerber viewer you can use such as: Gerbv, FlatCAM, CAM350, ViewMate, GerberLogix etc.

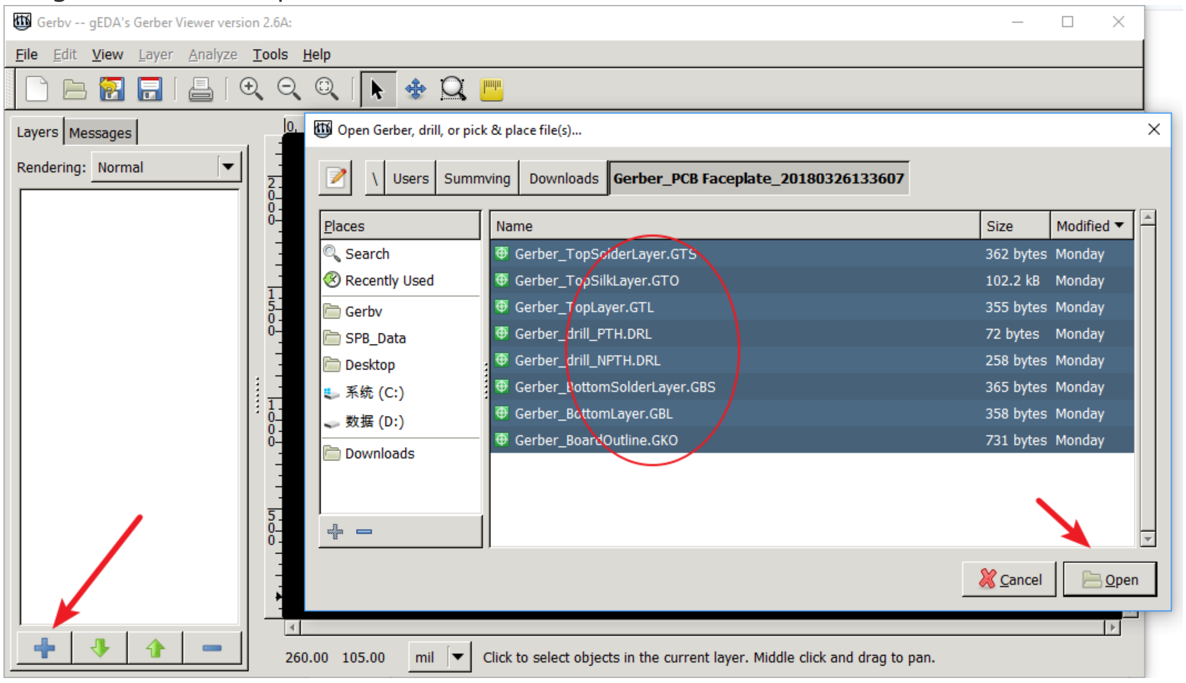
Gerber viewer recommend Gerbv:

- Project page:http://gerbv.geda-project.org/
- Download: https://sourceforge.net/projects/gerbv/files/

How to use Gerbv:

1.Download Gerber zip file, and download Gerbv, unzip Gerber file and run the Gerbv;

2.Click the + button at the Gerbv dialog bottom-left corner, open the gerber folder, select all the gerber files, and open.


Gerbv -- gEDA's Gerber Viewer version 2.6A:
File Edit View Layer Analyze Tools Help
Layers Messages
Rendering: Normal
Open Gerber, drill, or pick & place file(s)...
\ Users Summving Downloads Gerber_PCB Faceplate_20180326133607
Places
Search
Recently Used
Gerbv
SPB_Data
Desktop
系统 (C:)
数据 (D:)
Downloads
Cancel
Open
260.00 105.00 mil
Click to select objects in the current layer. Middle click and drag to pan.

Name	Size	Modified
Gerber_TopSolderLayer.GTS	362 bytes	Monday
Gerber_TopSilkLayer.GTO	102.2 kB	Monday
Gerber_TopLayer.GTL	355 bytes	Monday
Gerber_drill_PTH.DRL	72 bytes	Monday
Gerber_drill_NPTH.DRL	258 bytes	Monday
Gerber_BottomSolderLayer.GBS	365 bytes	Monday
Gerber_BottomLayer.GBL	358 bytes	Monday
Gerber_BoardOutline.GKO	731 bytes	Monday

3.And then zoom, measure, check every layer, check drill holes and location. etc.

FlatCAM is a nice tool too: http://flatcam.org/

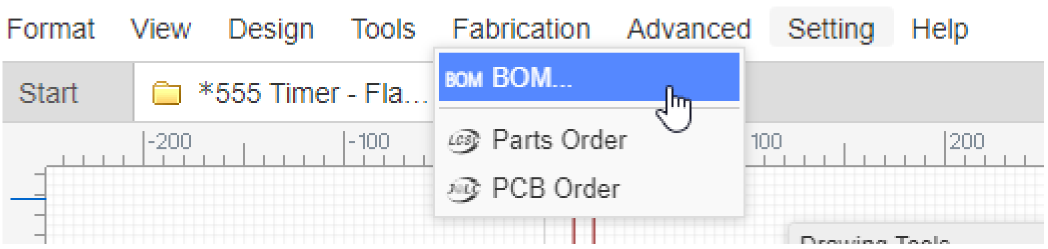
FlatCAM lets you take your designs to a CNC router. You can open Gerber, Excellon or G-code, edit it or create from scatch, and output G-Code. Isolation routing is one of many tasks that FlatCAM is perfect for. It's is open source, written in Python and runs smoothly on most platforms.

Free Online Gerber Viewer:

Recommend：
jlcpcb.com
tracespace.io/view
gerber.ucamco.com

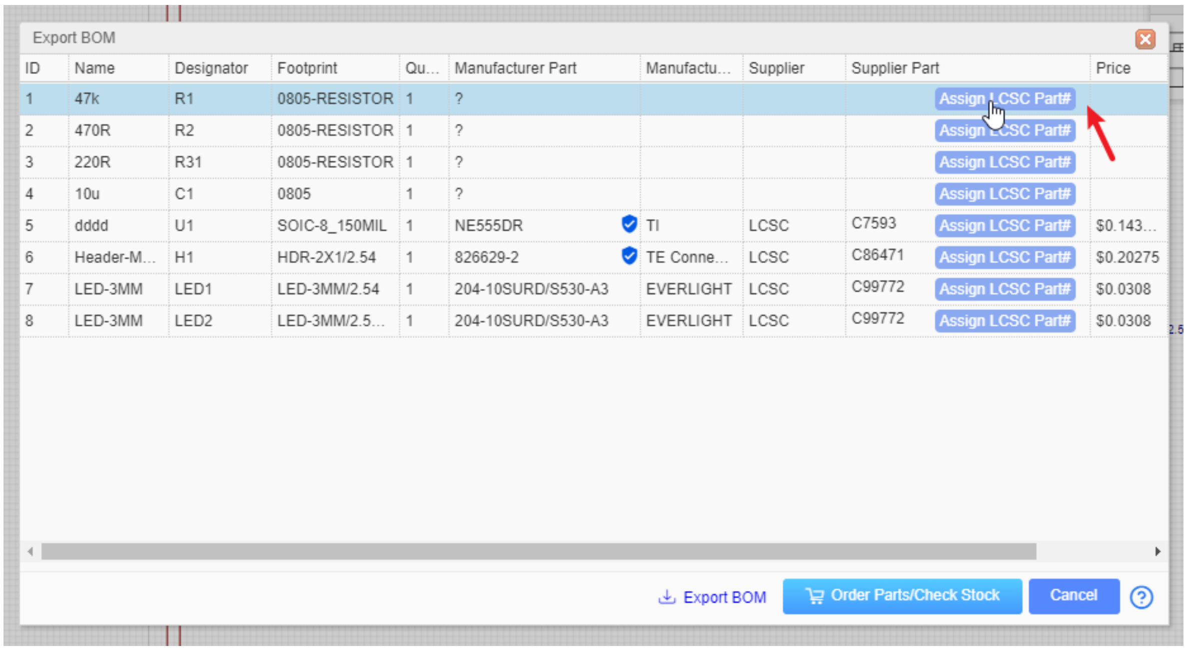
Export BOM

You can export the Bill of Materials (BOM) for the schematic (Document) and PCB, via: "Top Menu - File - Export BOM", or "Top Menu - Fabriaction - BOM".



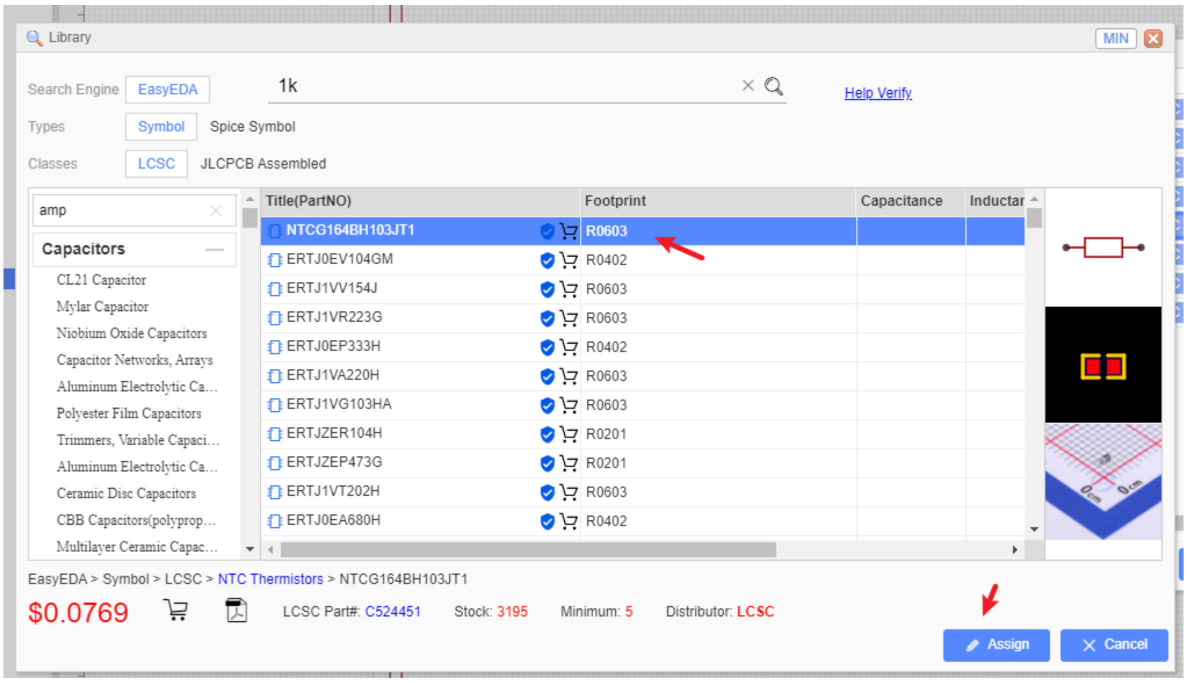
After clicking the BOM export option, the dialog below will open.

In this dialog，you can click the buttom to assign LCSC part's order code for your components.


Export BOM

ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part		Price
1	47k	R1	0805-RESISTOR	1	?				Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?				Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?				Assign LCSC Part#	
4	10u	C1	0805	1	?				Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593	Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471	Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308

After clicking on the assign icon，the components and footprints search dialog will pop up, and you can choose which component you want to assign.


Library

Search Engine: EasyEDA — 1k
Types: Symbol, Spice Symbol
Classes: LCSC, JLCPCB Assembled

amp

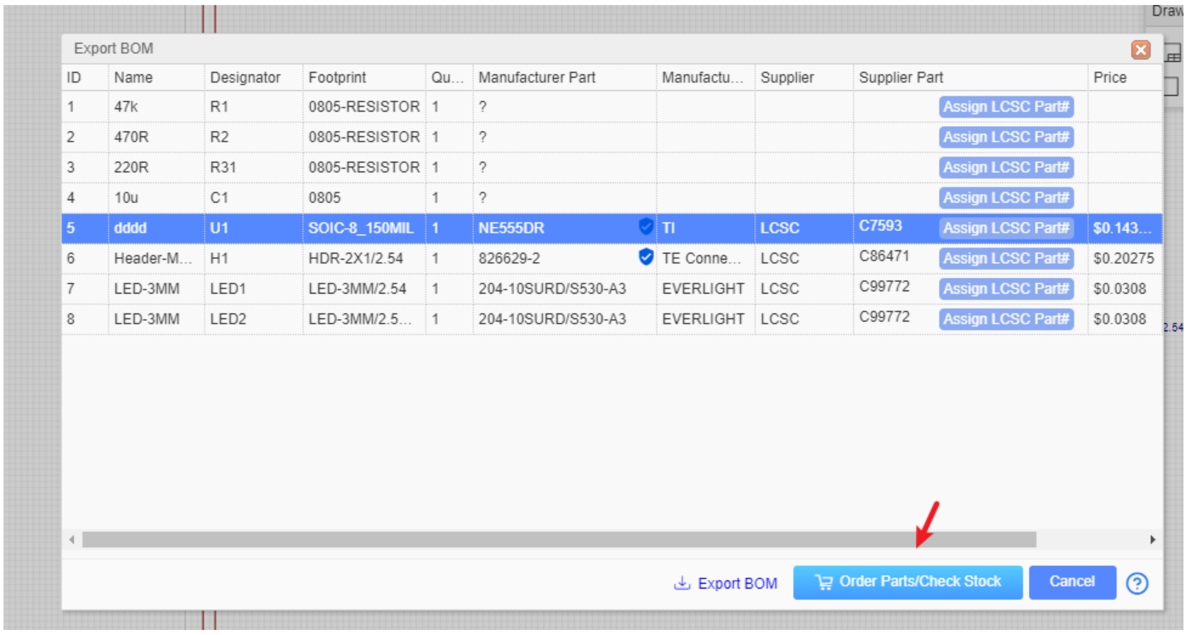
Capacitors
- CL21 Capacitor
- Mylar Capacitor
- Niobium Oxide Capacitors
- Capacitor Networks, Arrays
- Aluminum Electrolytic Ca...
- Polyester Film Capacitors
- Trimmers, Variable Capaci...
- Aluminum Electrolytic Ca...
- Ceramic Disc Capacitors
- CBB Capacitors(polyprop...
- Multilayer Ceramic Capac...

Title(PartNO)	Footprint	Capacitance	Inductar
NTCG164BH103JT1	R0603		
ERTJ0EV104GM	R0402		
ERTJ1VV154J	R0603		
ERTJ1VR223G	R0603		
ERTJ0EP333H	R0402		
ERTJ1VA220H	R0603		
ERTJ1VG103HA	R0603		
ERTJZER104H	R0201		
ERTJZEP473G	R0201		
ERTJ1VT202H	R0603		
ERTJ0EA680H	R0402		

EasyEDA > Symbol > LCSC > NTC Thermistors > NTCG164BH103JT1

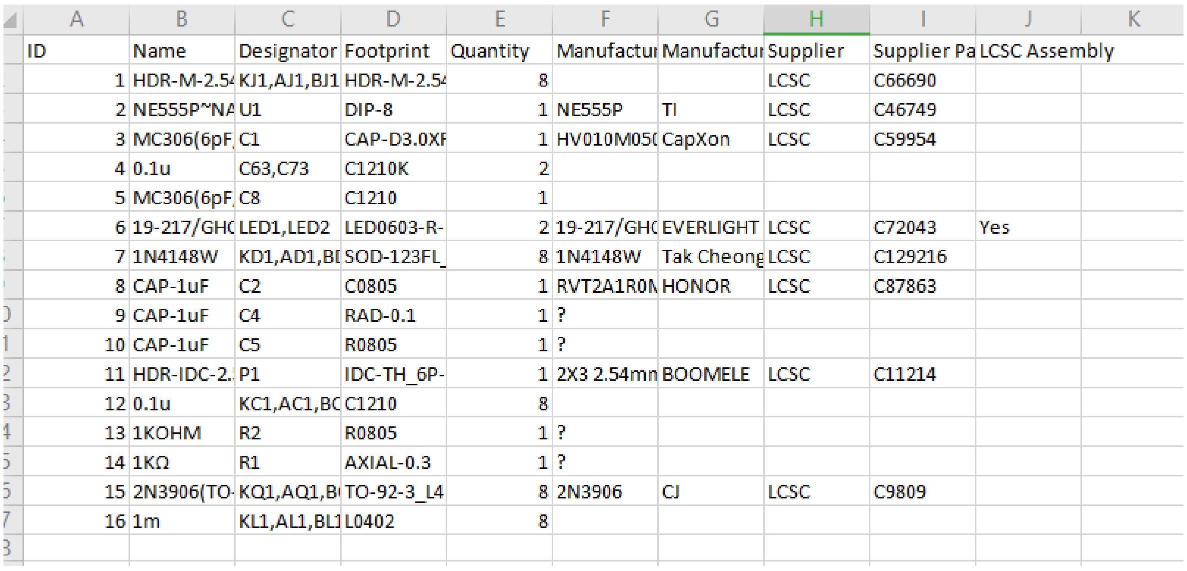
$0.0769 LCSC Part#: C524451 Stock: 3195 Minimum: 5 Distributor: LCSC

When you click the "Order Parts/Check Stock" button, we will help you to list all the components of your BOM at LCSC.com(If you haven't login LCSC, you have to login first). If you want to buy the components form LCSC, and you just need to put them to the cart and check out.



ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part		Price
1	47k	R1	0805-RESISTOR	1	?				Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?				Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?				Assign LCSC Part#	
4	10u	C1	0805	1	?				Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593	Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471	Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308

You can open the BOM in any text editor or spreadsheet.



A	B	C	D	E	F	G	H	I	J	K
ID	Name	Designator	Footprint	Quantity	Manufactur	Manufactur	Supplier	Supplier Pa	LCSC Assembly	
1	HDR-M-2.54	KJ1,AJ1,BJ1	HDR-M-2.54	8			LCSC	C66690		
2	NE555P~NA	U1	DIP-8	1	NE555P	TI	LCSC	C46749		
3	MC306(6pF,	C1	CAP-D3.0XF	1	HV010M050	CapXon	LCSC	C59954		
4	0.1u	C63,C73	C1210K	2						
5	MC306(6pF,	C8	C1210	1						
6	19-217/GHC	LED1,LED2	LED0603-R-	2	19-217/GHC	EVERLIGHT	LCSC	C72043	Yes	
7	1N4148W	KD1,AD1,BD	SOD-123FL_	8	1N4148W	Tak Cheong	LCSC	C129216		
8	CAP-1uF	C2	C0805	1	RVT2A1R0N	HONOR	LCSC	C87863		
9	CAP-1uF	C4	RAD-0.1	1	?					
10	CAP-1uF	C5	R0805	1	?					
11	HDR-IDC-2.	P1	IDC-TH_6P-	1	2X3 2.54mm	BOOMELE	LCSC	C11214		
12	0.1u	KC1,AC1,BC	C1210	8						
13	1KOHM	R2	R0805	1	?					
14	1KΩ	R1	AXIAL-0.3	1	?					
15	2N3906(TO-	KQ1,AQ1,BQ	TO-92-3_L4	8	2N3906	CJ	LCSC	C9809		
16	1m	KL1,AL1,BL1	L0402	8						

Export BOM supports to export LCSC part price, it is the same as LCSC website.

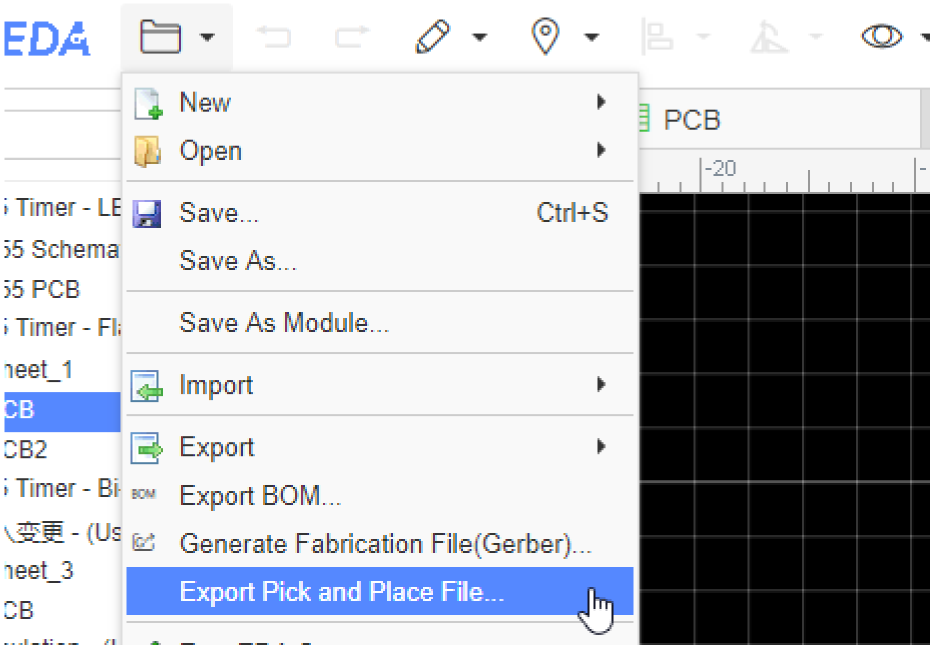
Notice:

- Before v6.4.17, If your project has schematic and PCB, the BOM data will come from schematic; if the project only has PCB, the BOM data will come from PCB.
- Since v6.4.17, the schematic BOM and PCB BOM are separated. If you assign the LCSC part at the PCB, it will not modify the schematic.
- In order to support multiple languages, BOM and coordinate files (CSV file) are UNICODE encoded and tab-based. If the CSV file cannot be read by your components vendor or PCB manufacturer, please convert the encoding and change the delimiter.
- Recommended solution: Save as a new CSV file in Excel or WPS. For example, open a CSV file in Excel, click or select: Save As - Other Formats - CSV (Comma Separated) (*. csv). You can also open the CSV file with any text editor (such as Windows Notepad) and save as ANSI or UTF-8 encoding. If necessary, replace all tabs with commas.

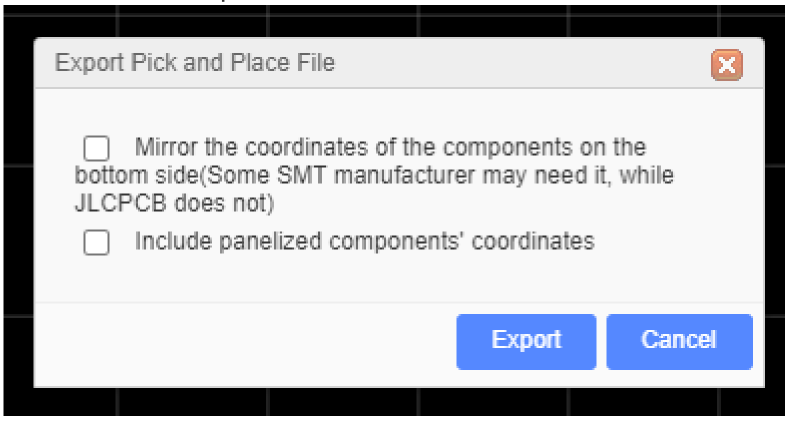
Export Pick and Place File

In PCB editor, if you want to generate Pick And Place as a CSV file, you can via:

File > Export Pick and Place File or **Top Menu - Fabrication - Pick and Place File**.



You can set the options:

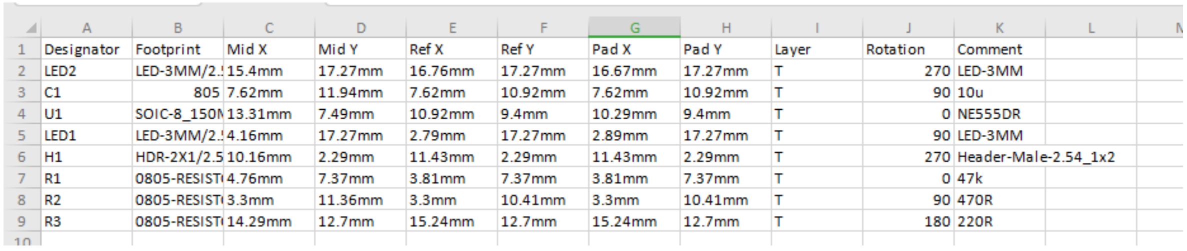

Export Pick and Place File

☐ Mirror the coordinates of the components on the bottom side(Some SMT manufacturer may need it, while JLCPCB does not)

☐ Include panelized components' coordinates

If your PCB has been panelize by the editor, you can enable the "Include panelized components coordinate".

When you open the exported CSV file, you can see:



	A	B	C	D	E	F	G	H	I	J	K
1	Designator	Footprint	Mid X	Mid Y	Ref X	Ref Y	Pad X	Pad Y	Layer	Rotation	Comment
2	LED2	LED-3MM/2.!	15.4mm	17.27mm	16.76mm	17.27mm	16.67mm	17.27mm	T	270	LED-3MM
3	C1	805	7.62mm	11.94mm	7.62mm	10.92mm	7.62mm	10.92mm	T	90	10u
4	U1	SOIC-8_150N	13.31mm	7.49mm	10.92mm	9.4mm	10.29mm	9.4mm	T	0	NE555DR
5	LED1	LED-3MM/2.!	4.16mm	17.27mm	2.79mm	17.27mm	2.89mm	17.27mm	T	90	LED-3MM
6	H1	HDR-2X1/2.5	10.16mm	2.29mm	11.43mm	2.29mm	11.43mm	2.29mm	T	270	Header-Male-2.54_1x2
7	R1	0805-RESIST	4.76mm	7.37mm	3.81mm	7.37mm	3.81mm	7.37mm	T	0	47k
8	R2	0805-RESIST	3.3mm	11.36mm	3.3mm	10.41mm	3.3mm	10.41mm	T	90	470R
9	R3	0805-RESIST	14.29mm	12.7mm	15.24mm	12.7mm	15.24mm	12.7mm	T	180	220R

This file support two units "mm" and "mil", it is following the PCB unit setting.

There is an option "Mirror the coordinates of the components on the bottom side(Some SMT manufacturer may need it, while JLCPCB does not)", you can check with your SMT manufacturer, the mostly SMT manufacturer doesn't need it.

Notice:

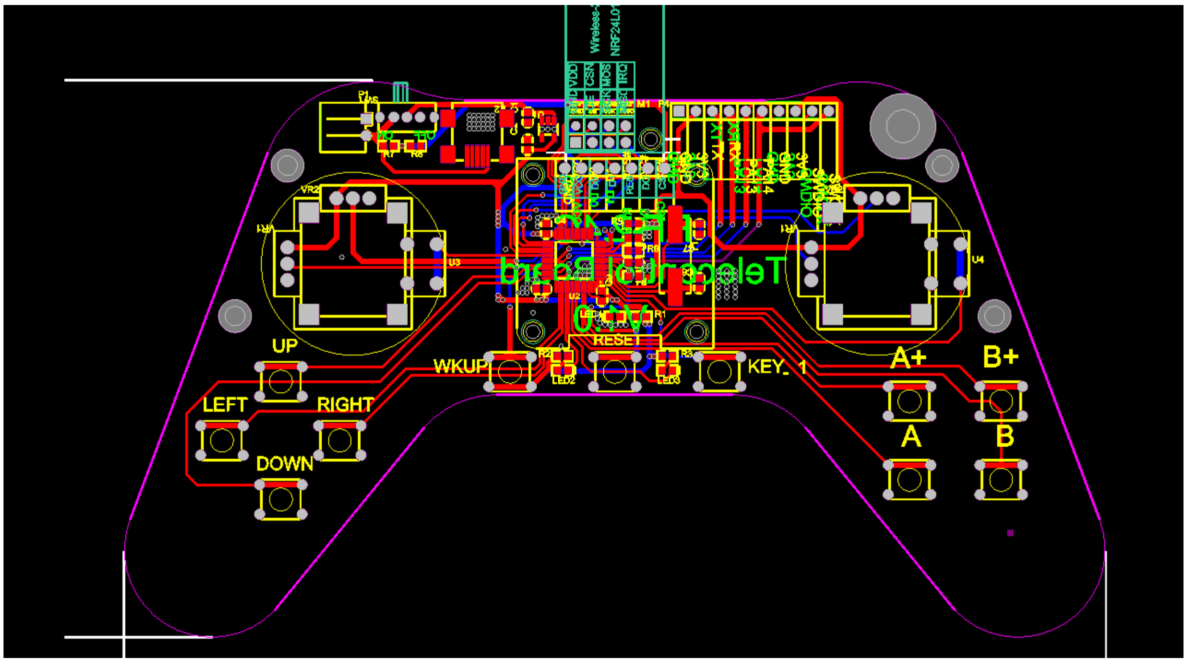
- In order to support multiple languages, BOM and Pick and Place files (CSV file) are UNICODE encoded and tab-based. If the CSV file cannot be read by your components vendor or PCB manufacturer, please convert the encoding and change the delimiter.

- Recommended solution: Save as a new CSV file in Excel or WPS. For example, open a CSV file in Excel, click or select: Save As - Other Formats - CSV (Comma Separated) (*. csv). You can also open the CSV file with any text editor (such as Windows Notepad) and save as ANSI or UTF-8 encoding. If necessary, replace all tabs with commas.

Export DXF

EasyEDA support to export PCB to DXF.

At present EasyEDA supports to export a full layers and objects DXF file:



You can edit it at CAD tools very easy, toggle the layers as you want.

How to Order PCB

Order Parts

1. Finish the schematic and PCB design at EasyEDA.
2. Open schematic, click " - Export BOM" button, the BOM dialog will open, click "Order Parts/Check Stock" button, will open LCSC.com order page. Check Export BOM
3. Add the parts to the cart, and then submit the payment.

Order PCB

1. Open PCB, click " - Generate Fabriaction File(Gerber)". Check Generate Fabrication File(Gerber)
2. Before ordering, check the Gerber first: Gerber Viewer
3. Visit at JLPCB https://jlcpcb.com/quote ,login with EasyEDA accout.

4. Order PCB from EasyEDA editor directly(at Generate) or you can add the Gerber file(compressed file, ZIP) on the page and type the order options.
5. If you want to assembly parts, before enable the SMT option, you need to check all your parts are using "LCSC Assembled" class libs, and then upload the BOM file and Pick and Place file.
   - LCSC Assembled Libraries
   - Export BOM
   - Generate Fabrication File(Gerber)
6. Save to the Cart, and then submit the payment.

Doesn't support to combine the components order with the PCB order.

More information please refer at:

How to place an order

How to order a SMT order

How to order a stencil

Footprint Library

Create The Footprint

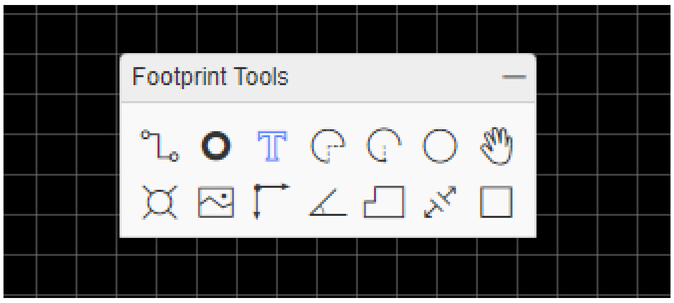
Create The Footprint

There will be times when you will need a PCB footprint that is not already in the EasyEDA libraries.

Footprint Tool

The process of creating your own Footprints is very similar to how you make symbols for your own Schematic Libs.

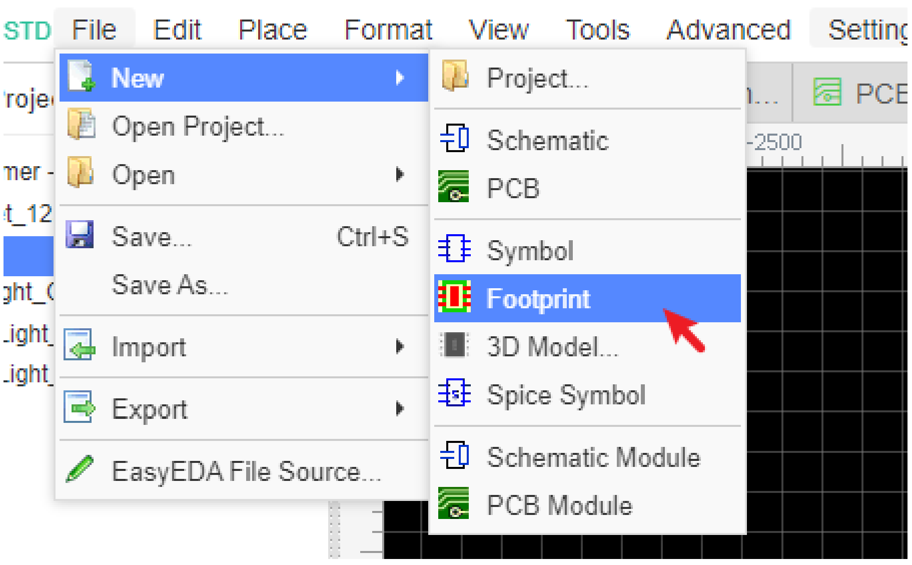
Footprint Tools almost are the same as PCB tools, just lacking some of the functions.



Create Footprint

Start a new Footprint as shown below or by doing:

File > New > Footprint



This opens the New Footprint editor.

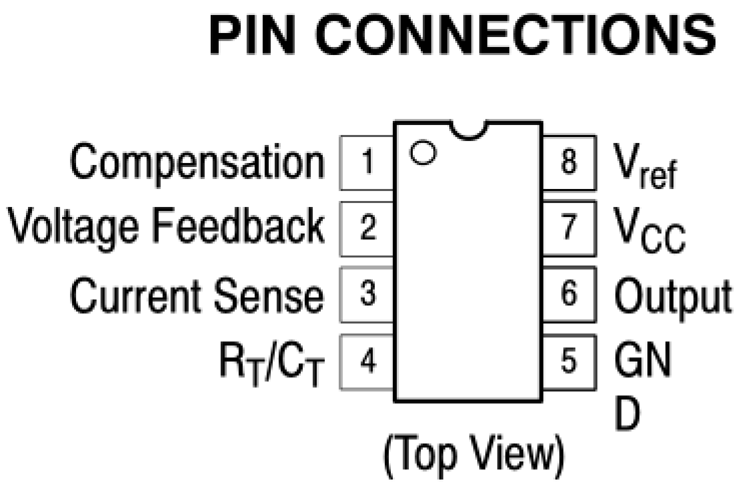
Drawing Steps

1.Downlod the datasheet which you need to draw the Footprint, such as SOIC-8.Such as PDF: UC2844BD1R2G

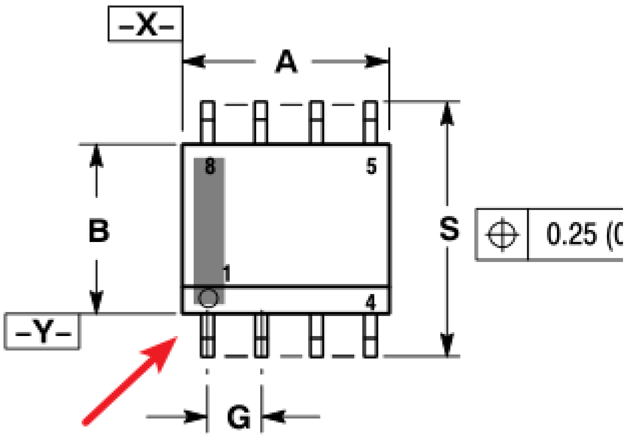
2.Read the datasheet, notice the 0 degree of the Footprint (The 0 degree is the Footprint's direction when you placed it on the PCB without rotation), the right 0 degree will helpful for PCB SMT.

3.Check the footprint size, pad/pin direction and polarity, and then place the Pads on the canvas. You can adjust the pad size base on your real usage situation.

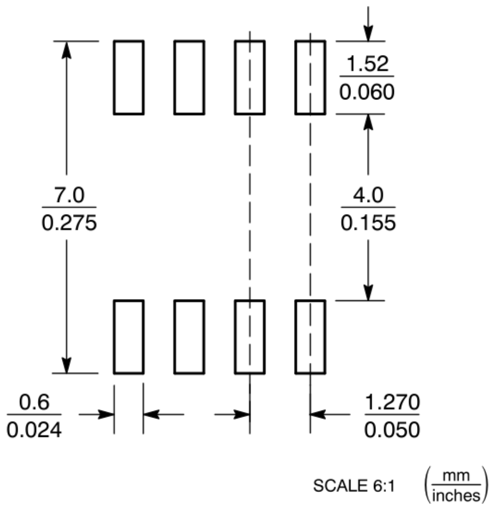
- Component's pin direction, page 1.

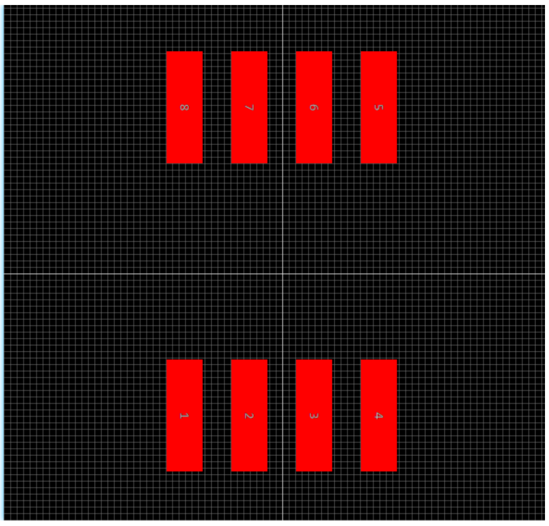


- footprint polarity, page 1 and 18.



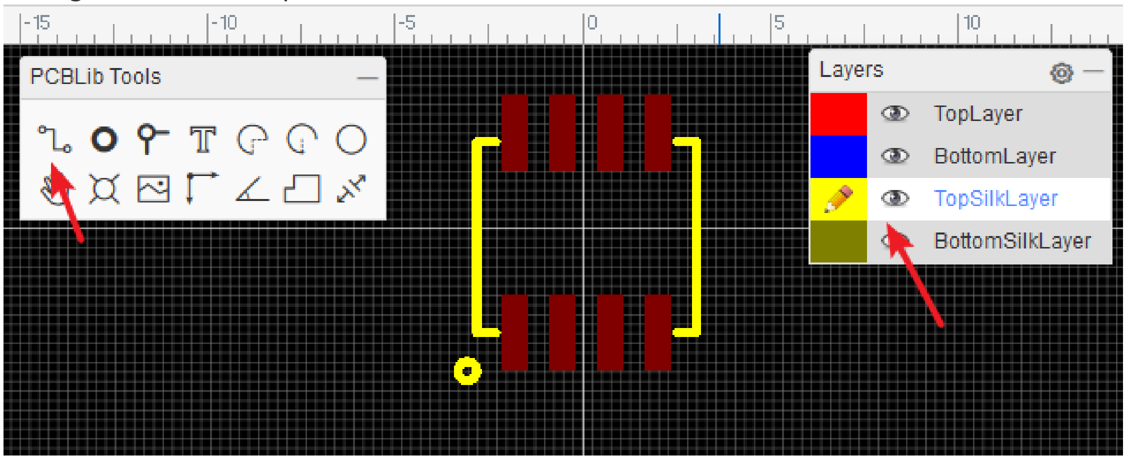
- Depends on page 18, placing one pad on the canvas on the top layer, and then change the pad number, size, shape type etc. And then set the coordinate for it, and place the less pads, you can use the "Top Menu - Align" tools to align the pads to fit the location. If you want to move the pad by mouse or direction key by small steps, you can set a new snap size at the right-hand panel.



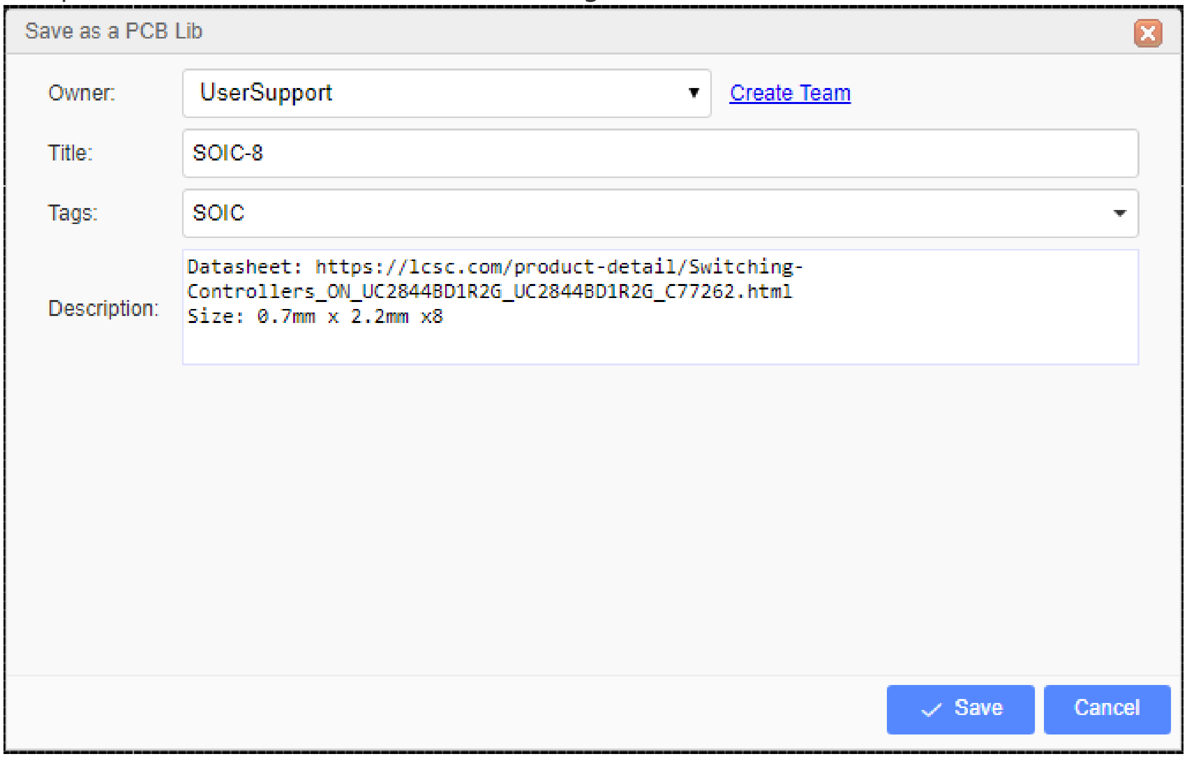


4.Drawing the Footprint silk screen. And sometimes you can add some marking and text on the mechanical or document layer.

- Swicthing layer to TopSlikLayer
- Using the Track and Arc to draw the silk screen. The editor doesn't support draw the retangle silk screen at present.



5.Filling the footprint title and prefix at the right-hand "Custom Attributes", and then Save. When you save it , please fill the tags, description, the description suggesting add the footprint datasheet link and footprint size, that can help you or other people to recognize this Footprint whether if it can be used for the design.



Owner: UserSupport

Title: SOIC-8

Tags: SOIC

Description:
```
Datasheet: https://lcsc.com/product-detail/Switching-Controllers_ON_UC2844BD1R2G_UC2844BD1R2G_C77262.html
Size: 0.7mm x 2.2mm x8
```

6.Use the dimension tool to check the Footprint size, via: Top Menu - Tools - Check Dimension.

7.Set the origin. You can via: "Top Menu - Place - Set Canvas Origin - By Center of Pads" to set the origin.

8.Save.

Then the PCB footprint creating finish .

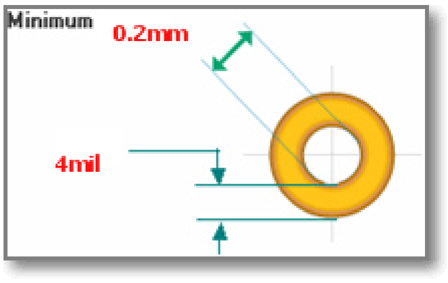
Notice:

- The Origin Point. To simplify rotating your symbols when they are placed into the canvas, make sure all of your symbols are created as near as possible centered around that point. Suggesting the footprint center to be the origin point. That will helpful to rotation when you placing it on the canvas, and help to do the SMT more quickly.
- The pad center suggesting one and more on the grid , avoid when place it on the PCB causing the track hard to connect issue.
- The pad number can be set as number and alphabet, they must match with the SchematicLib's pin number, otherwise the component which was assigned this Footprint will alert the error at the footprint manager, and can' not convert the schematic to PCB.
- The pad number will increase by placing with mouse, if you copy and paste it, the number will not increase.

Others

- It is important to set the right Snap and Grid sizes to ensure that the pads on the finished footprint snap exactly to the grid and so connect the nets. For example, if you are creating a DIP footprint, set the Grid size to 100mil.
- Keep all other shapes such as component outlines and any associated pin identification marks or text on the TopSilkLayer. EasyEDA will automatically take care of the actual layer assignment when you place the footprint on the PCB.

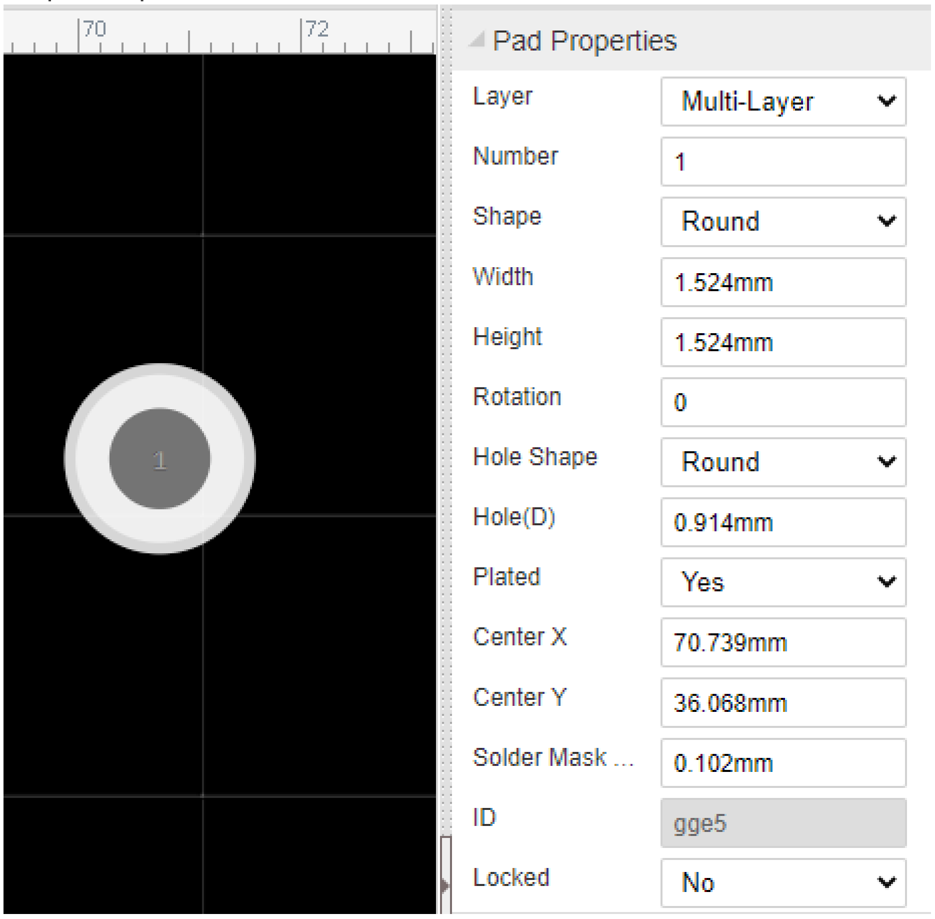
- `CTRL+S` to save your footprint designs and you will find them saved into the **Libraries > Classes: Footprint > Personal > Created** section of the left Navigation panel.
- Annular ring of the pad/via is too small, keep the annular ring >= 4mil. In this case, you can add a `Hole`



Pad attributes

You can add pads using the Pads button from the Footprint Tools palette or using the `P` hotkey.

After selecting one of the pads, you can view and adjust its attributes in the right hand Properties panel.



Pad Properties

Layer	Multi-Layer
Number	1
Shape	Round
Width	1.524mm
Height	1.524mm
Rotation	0
Hole Shape	Round
Hole(D)	0.914mm
Plated	Yes
Center X	70.739mm
Center Y	36.068mm
Solder Mask ...	0.102mm
ID	gge5
Locked	No

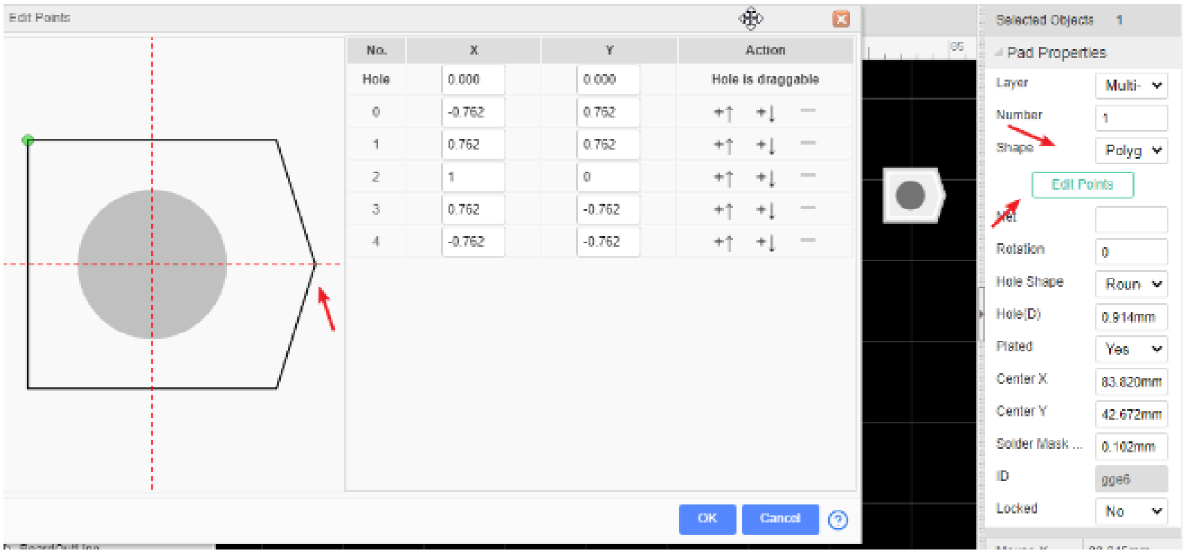
Number: Remembering the pin numbers you set in the schematic symbol in your Schematic symbol: to connect those schematic symbol pins to the pads in your PCB footprint, the pad numbers you set here in the Footprint footprint must be the same.

Shape: Round , Rectangular , Oval and Polygon.

EasyEDA supports four shapes: `Round` , `Rectangular` , `OVAL` and `POLYGON`.

- `OVAL` PAD will give your more space.
- `POLYGON` PAD will let you to create some strange pad.

Like in the image below, you can edit the PADs points when you select a `POLYGON` PAD



No.	X	Y	Action
Hole	0.000	0.000	Hole is draggable
0	-0.762	0.762	+↑ +↓ —
1	0.762	0.762	+↑ +↓ —
2	1	0	+↑ +↓ —
3	0.762	-0.762	+↑ +↓ —
4	-0.762	-0.762	+↑ +↓ —

Selected Objects 1

Pad Properties

Layer	Multi-
Number	1
Shape	Polyg
	Edit Points
Net	
Rotation	0
Hole Shape	Roun
Hole(D)	0.914mm
Plated	Yes
Center X	83.820mm
Center Y	42.672mm
Solder Mask ...	0.102mm
ID	gge6
Locked	No

Layer: If the pads are part of a **SMD** footprint, you can set it to **Top layer** or **Bottom layer**. For through hole components you should set it to **Multi-Layer**.

Net: You don't need to enter anything here because at present this footprint is not connected to anything in a circuit.

Width and Height: When the shape is set to Round, Width will equal Height.

Rotation: Here you can set the Pad's rotation as you want.

Hole(D): This is the drill hole **diameter** for a through hole pad. For a SMD Pad, set this to **zero**.

Center-X and Center-Y: using these two attributes, you can set the pad's position with more precision, compared to using the mouse.

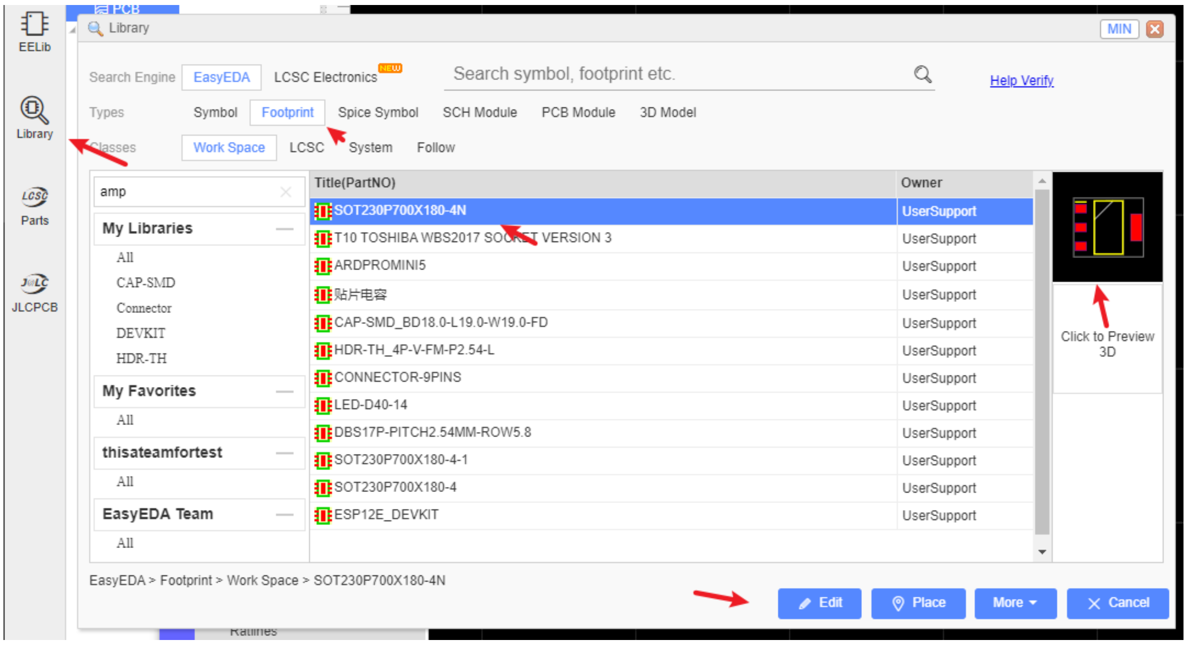
Plated: Yes or No. when the pad is multi-layer pad, if it set the plated as no, this pad top side and bottom side will not be connected together.

Edit Footprints

Edit Footprint in Library

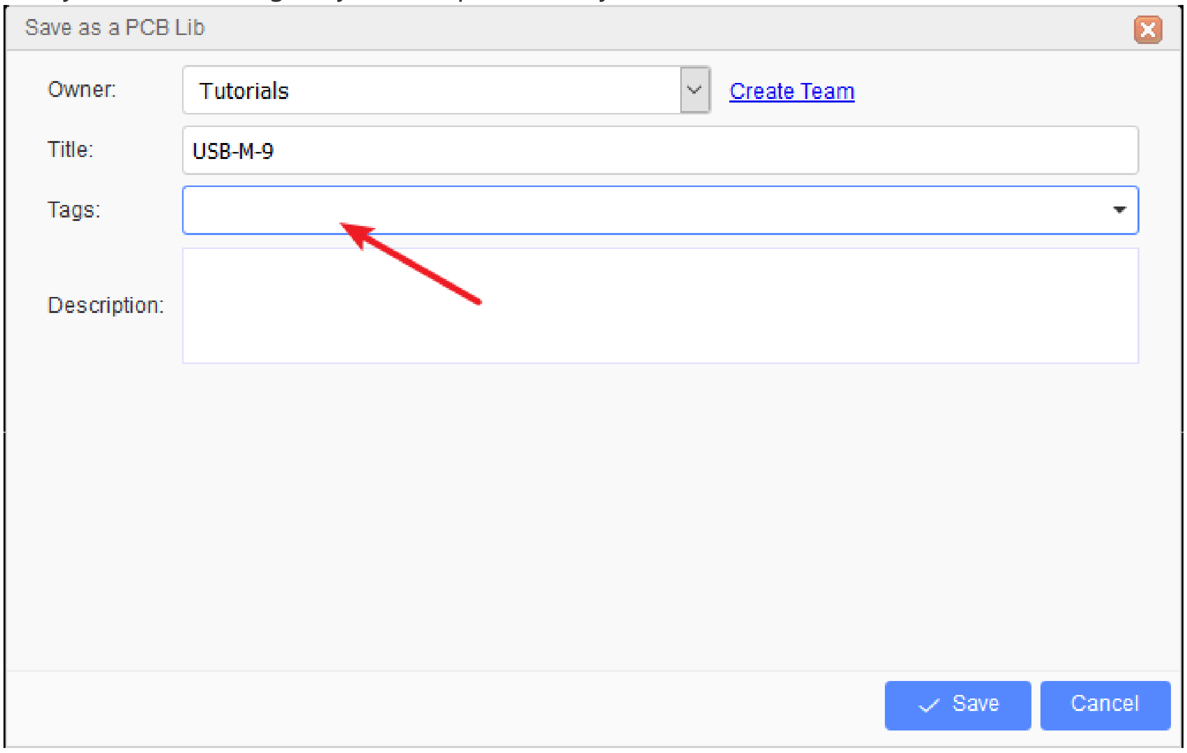
When you found a Footprints(footprint) but it can not be satisfied for your design, you can edit it to be your personal PCB footprint.

Via **Library** > **Footprint** > **Search Component/Personal/LCSC/System** > **Select footprint** > **Edit**

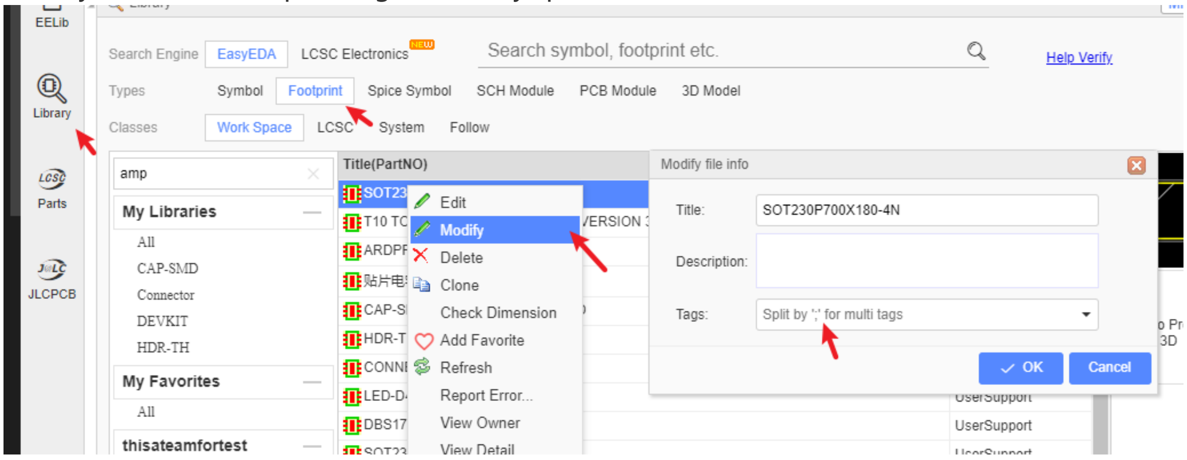


You can edit the pad size, shape outlines, etc. when you finish and save, it will be saved to your personal libraries "Created" and become your personal libraries.

And you can add a tag for your Footprint when you save it:



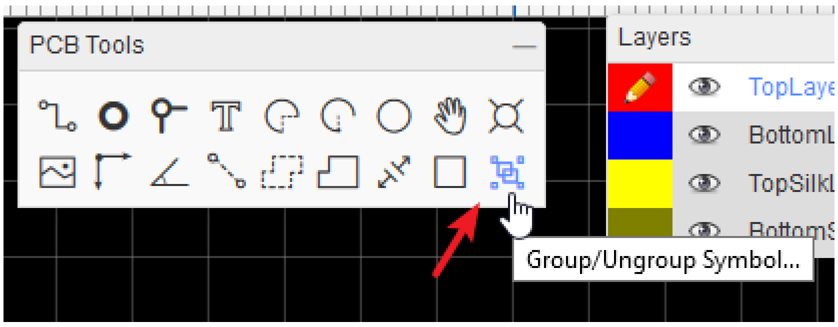
Modify the saved Footprint tag at "Library" part list.



Edit Footprint in PCB

If you want to edit a package(footprint) in the PCB, you can use the Ungroup/Group function same as the schematic.

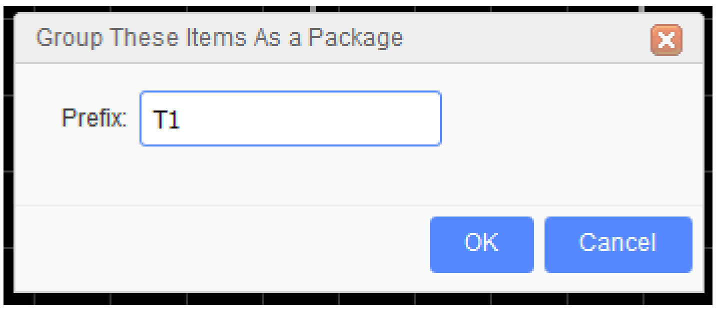
On the **PCB Tools** palette there is the **Group/Ungroup Symbol...** button.



This tool is for you to quickly create or edit library symbols.

1. Select a footprint
2. Click the **Group/Ungroup Symbol...** button
   Up to this point you have a collection of separate pads, a drawn silk layer tracks and some text that are all separate items with no particular association with each other.
3. Edit the shape or pad what you want to change
4. Select all of the items and click the **Group/Ungroup Symbol...** button.

A dialog will be opened:


Group These Items As a Package

Prefix: T1

After you click OK, all those separate elements will be grouped together to form your new symbol directly in the PCB.

Using the group function, you can create/edit any symbol in the Schematic/PCB, easily and quickly.

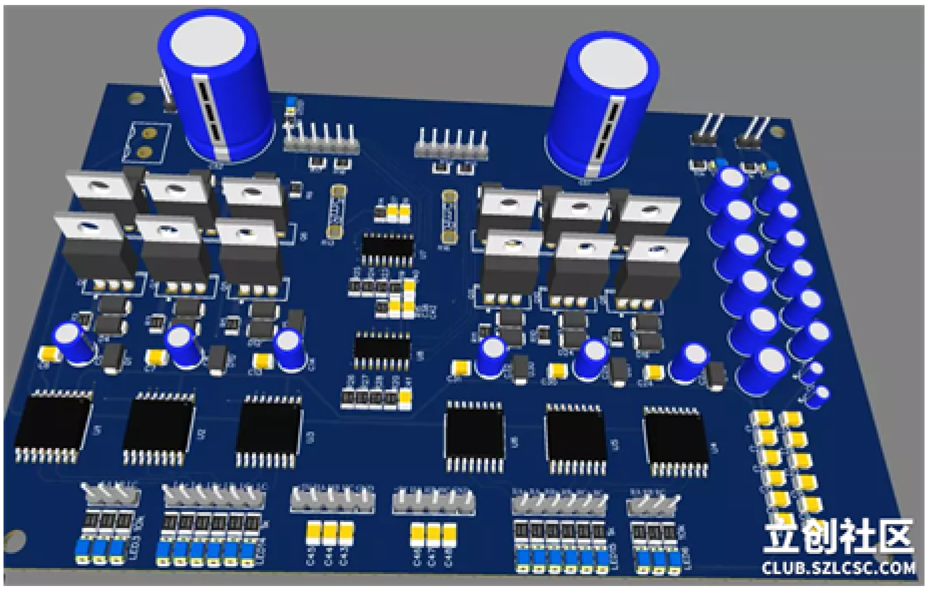
Notice:

- Before ungroup the footprint, please change it's layer to top layer first, because of the footprint after grouping will at top layer.

Import PCB 3DLib

Import 3D File

EasyEDA supports for importing 3D models, PCB can view cool 3D models when doing 3D preview. Exporting PCB 3D model files is not supported yet.



1. Draw or download 3D model

Note: currently only 3D models in "WRL(VRML)" and "obj" are supported. WRL can be imported directly without the need for compression; Obj must be compressed into a zip file with the "mtl" file and then imported, and the "mtl" file is usually taken with you when you download the obj file. Other formats of 3D files wii be supported in the future.

Note that file suffixes cannot be capitalized.

Download address:

https://library.io/explore/3dmodels ("mtl" files are automatically downloaded when obj files are downloaded.)

https://github.com/KiCad/kicad-packages3D

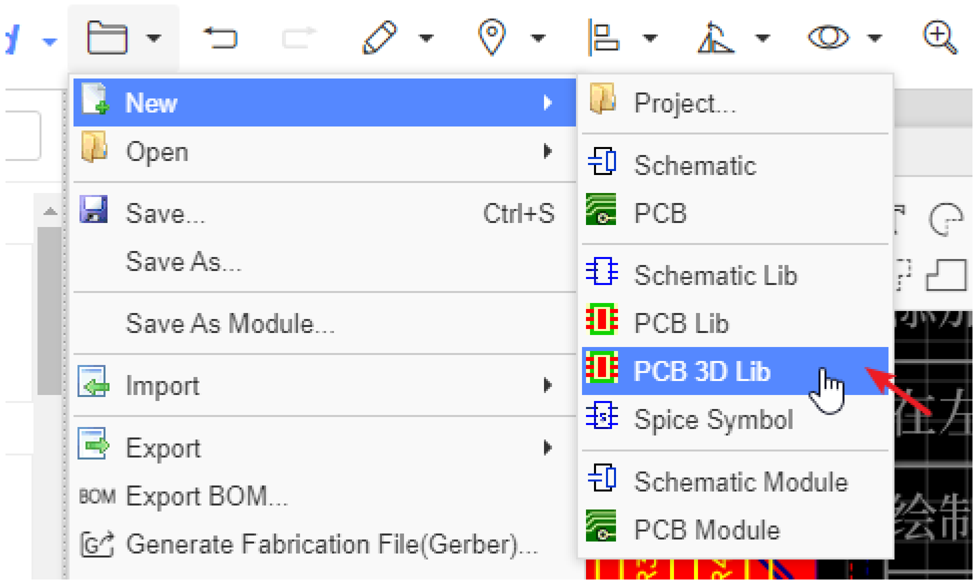
https://www.traceparts.com/zh

https://www.3dcontentcentral.com/

https://grabcad.com/

2. Create a new 3D library

in "Top Meun - New - PCB 3D Lib".



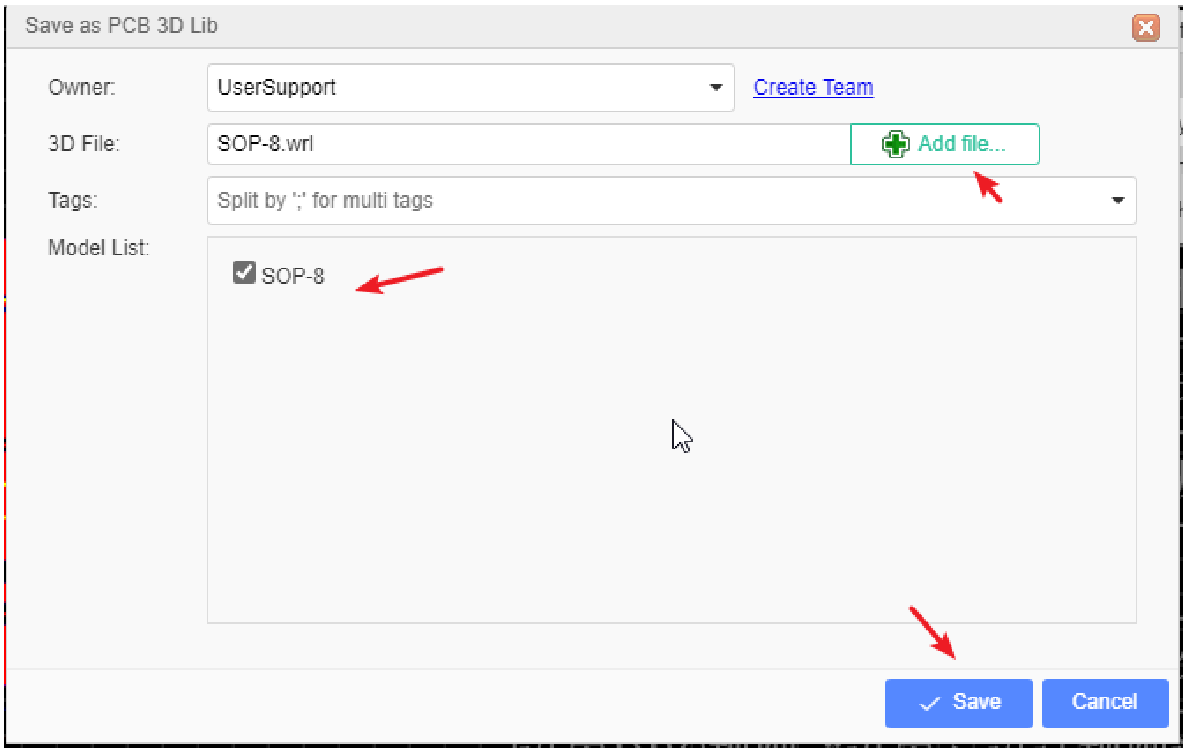
If you have many 3D libraries, you can zip them to a zip file to import, no more than 10 WRL files for one zip file, otherwise it will fail to import.

OBJ format contains many 3D models in one file, you don't need to zip them.

If your WRL file cannot import directly, please use FreeCAD to export a new one and try again.

3. Import 3D model.

You can check which 3D model you want to import.



Save as PCB 3D Lib

Owner: UserSupport

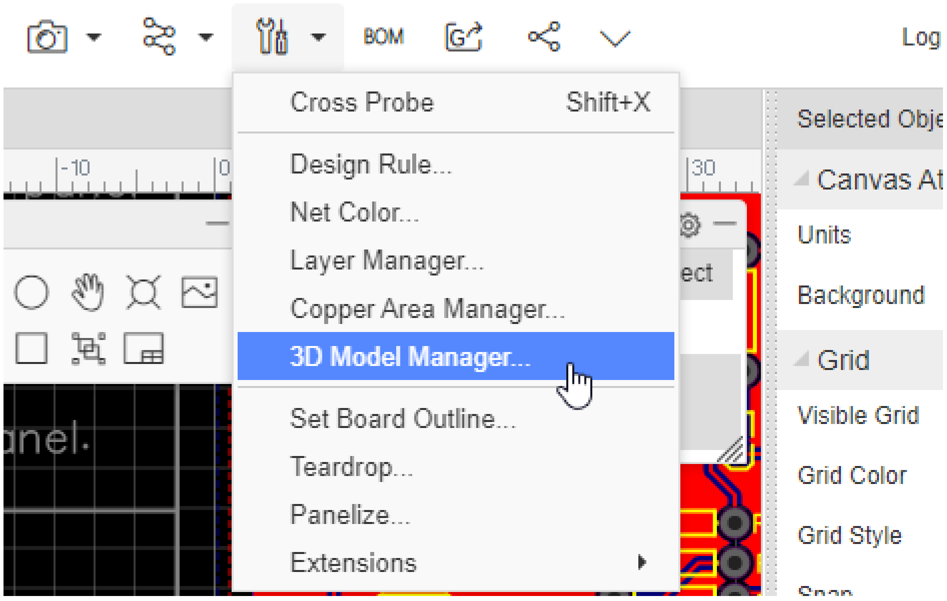
3D File: SOP-8.wrl

Tags: Split by ';' for multi tags

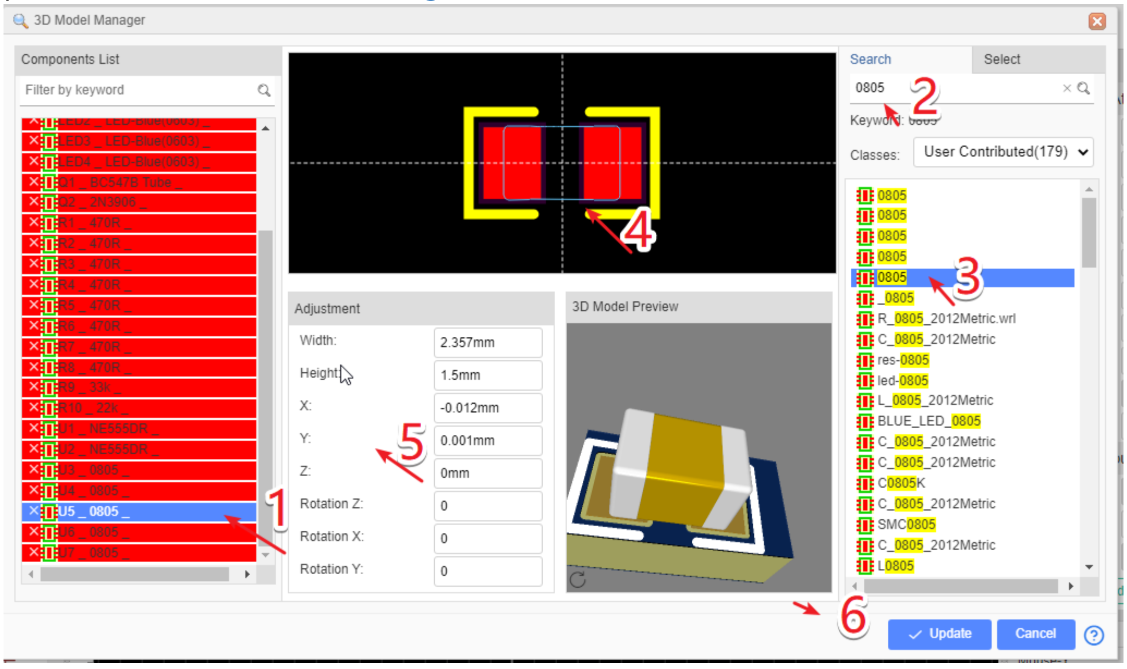
Model List: ☑ SOP-8

4. Specify the 3D model

- Open the PCB or Footprint, and find " - Tools - 3D Model Manager"



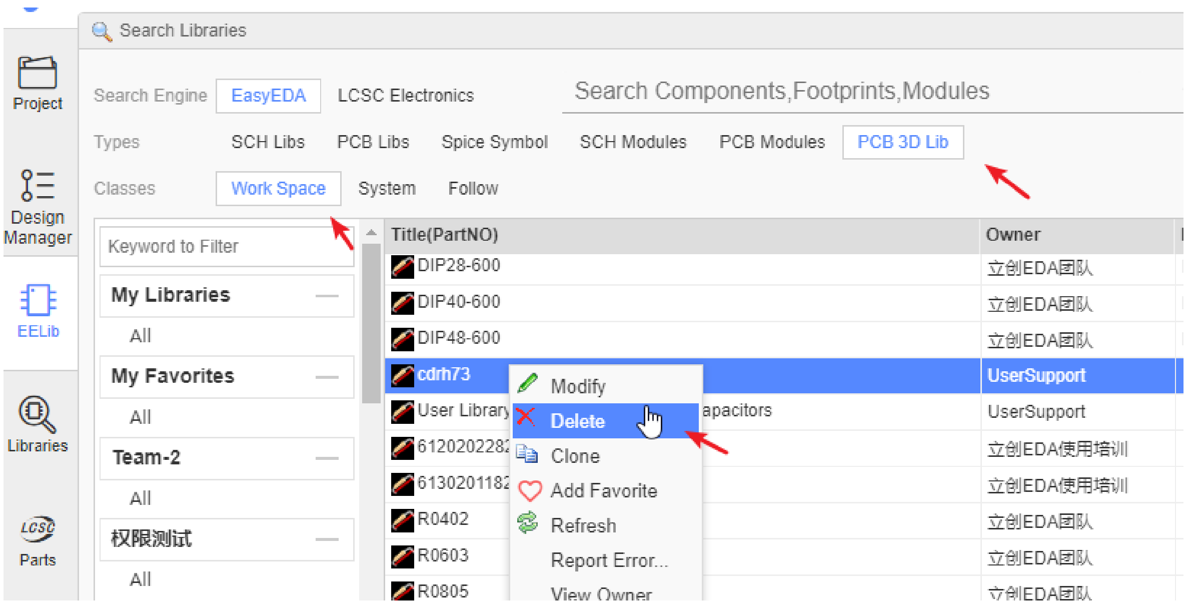
- Specify the imported 3D model for the corresponding footprint, which is basically consistent with the footprint manager operation. For the specific use of the tutorial, please see: [PCB - 3D Model Manager]()



- Adjust the position and parameter relationship between the 3D model and PCB packaging, and click update
- After completing all the specified 3D models, you can start the 3D preview of the whole PCB.

Edit 3D Lib

1. The SHIFT+F shortcut opens the component library dialog box
2. Switch to "PCB 3D library" and "WorkSpace"
3. Right click can edit and delete 3D library



FAQ:

Q：Can the official footprint library specify the 3D model first?
A：Yes, later official libraries will specify 3D models. At present, you need to specify to PCB or PCB footprint.

Q：Can EasyEDA export the whole PCB 3D format for structural design? Step, etc.
A：It will be supported in the future, step by step, and will directly support importing the step format in the future. This format is more complicated and needs time to study.

Q：Will EasyEDA support to draw 3D models in the future?
A：Don't. At present, many 3D rendering tools are very mature (Auto CAD, CAXA, SolidWorks, etc.) or open source free (FreeCAD, LibreCAD). Online 3D design tools (onshape) are also available.

Footprint Naming Rule

EasyEDA Footprint Naming Rule

EasyEDA Footprint Naming Rule Reference

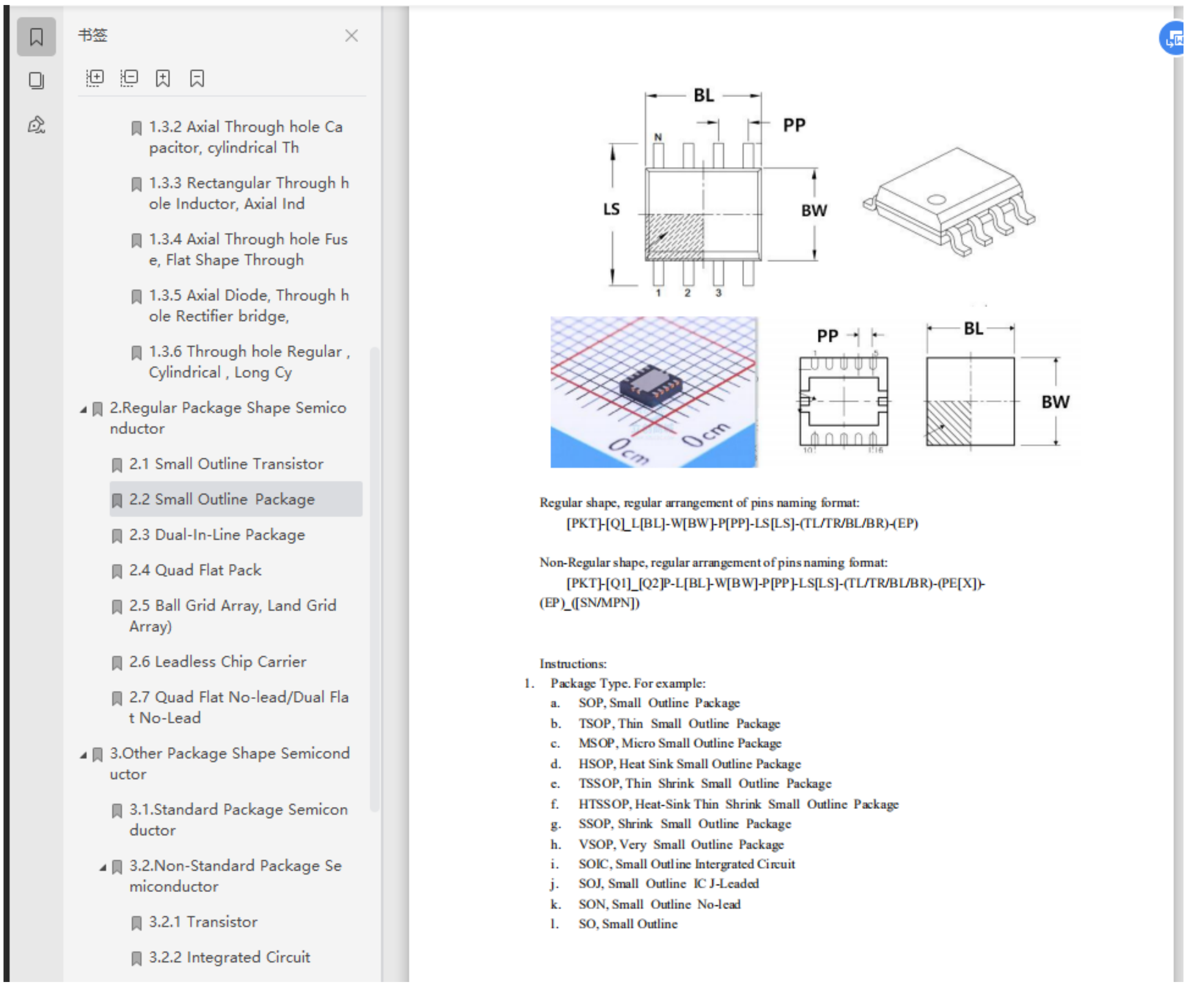
Introduction:

Believe that the vast number of electronic engineers will encounter the problem of footprint name naming, and now EasyEDA to provide everyone with a reference scheme - "EasyEDA Footprint Naming Rule Reference".

Each company should have its own footprint naming specification,EasyEDA is no exception, EasyEDA has more than 180,000 of official library (LCSC library), multiple engineers in the construction of footprint, more need unified library rules and footprint naming rules to ensure library consistency and footprint reuse.

Written by LCSC engineering department and EasyEDA team, after close one year of running in, now we are very happy to release the "EasyEDA Footprint Naming Rule Reference".

EasyEDA has been established according to the new footprint naming specification Footprints for more than half a year, and EasyEDA will continue to draw new library according to this rule.



Regular shape, regular arrangement of pins naming format:

[PKT]-[Q]_L[BL]-W[BW]-P[PP]-LS[LS]-(TL/TR/BL/BR)-(EP)

Non-Regular shape, regular arrangement of pins naming format:

[PKT]-[Q1]_[Q2]P-L[BL]-W[BW]-P[PP]-LS[LS]-(TL/TR/BL/BR)-(PE[X])-(EP)_([SN/MPN])

Instructions:

1. Package Type. For example:
   a. SOP, Small Outline Package
   b. TSOP, Thin Small Outline Package
   c. MSOP, Micro Small Outline Package
   d. HSOP, Heat Sink Small Outline Package
   e. TSSOP, Thin Shrink Small Outline Package
   f. HTSSOP, Heat-Sink Thin Shrink Small Outline Package
   g. SSOP, Shrink Small Outline Package
   h. VSOP, Very Small Outline Package
   i. SOIC, Small Outline Intergrated Circuit
   j. SOJ, Small Outline IC J-Leaded
   k. SON, Small Outline No-lead
   l. SO, Small Outline

The majority of EasyEDA users can also according to this rule:

1. Find the components of the specified package type;
2. Create your own or team's or company's footprint according to this rule;
3. Quickly reuse the official footprint.

Characteristics:

1. The rules of "package type _ feet number - body width - foot distance - body length - foot azimuth - polarity direction _ series name" are adopted in naming, so that users can quickly and clearly footprint most of the information
2. It covers most common component classification and encapsulation types and can quickly locate and query
3. Continuously expand new naming rules according to new components or packaging types, and continuously update and maintain
4. Public distribution, free of charge for both individuals and enterprises

Disadvantages:

Titles of some footprint types are too long

Update record:

2019.12.27 First release

Download:

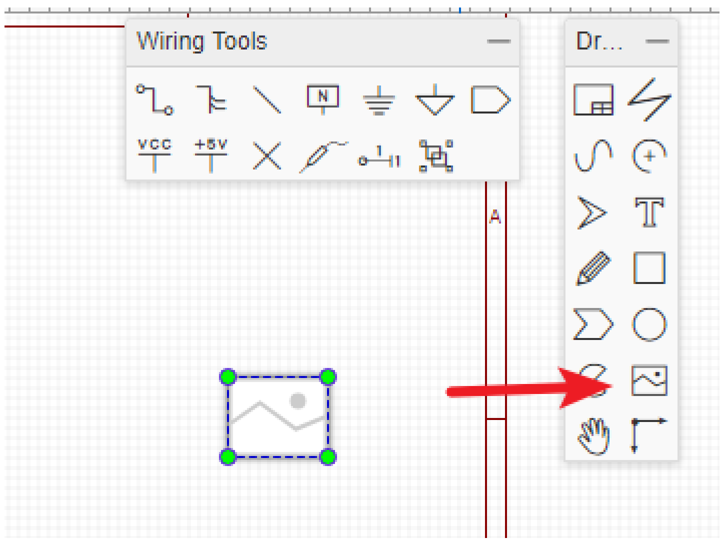
Download: EasyEDA Footprint Naming Rule Reference.pdf

Import

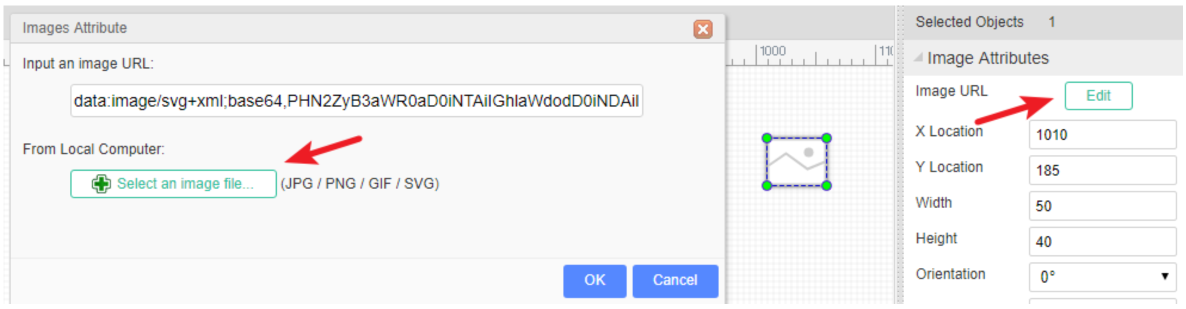
Import Image

Import Image to Schematic

When you select Image from the Drawing Tools palette, an image place holder will be inserted into the canvas:



Select the place holder, so you can see the image's attributes in the right hand Properties panel:


Images Attribute

Input an image URL:

data:image/svg+xml;base64,PHN2ZyB3aWR0aD0iNTAiIGhlaWdodD0iNDAiI

From Local Computer:

Select an image file... (JPG / PNG / GIF / SVG)

OK Cancel

Selected Objects 1

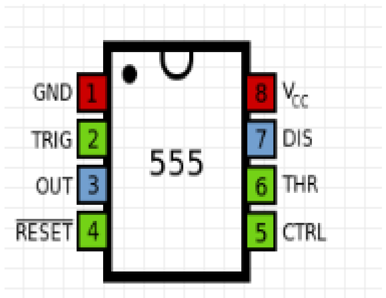
Image Attributes

Image URL	Edit
X Location	1010
Y Location	185
Width	50
Height	40
Orientation	0°

Set the URL of your image. For example, setting the URL to:

http://upload.wikimedia.org/wikipedia/commons/thumb/c/c7/555_Pinout.svg/220px-555_Pinout.svg.png

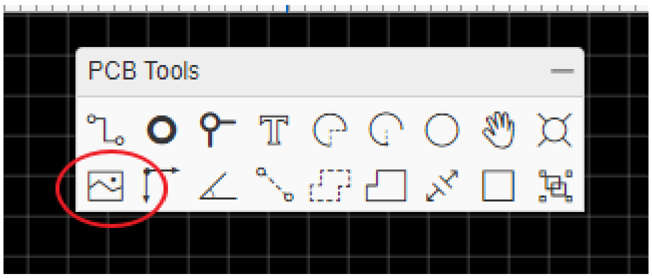
will make your image look like this:



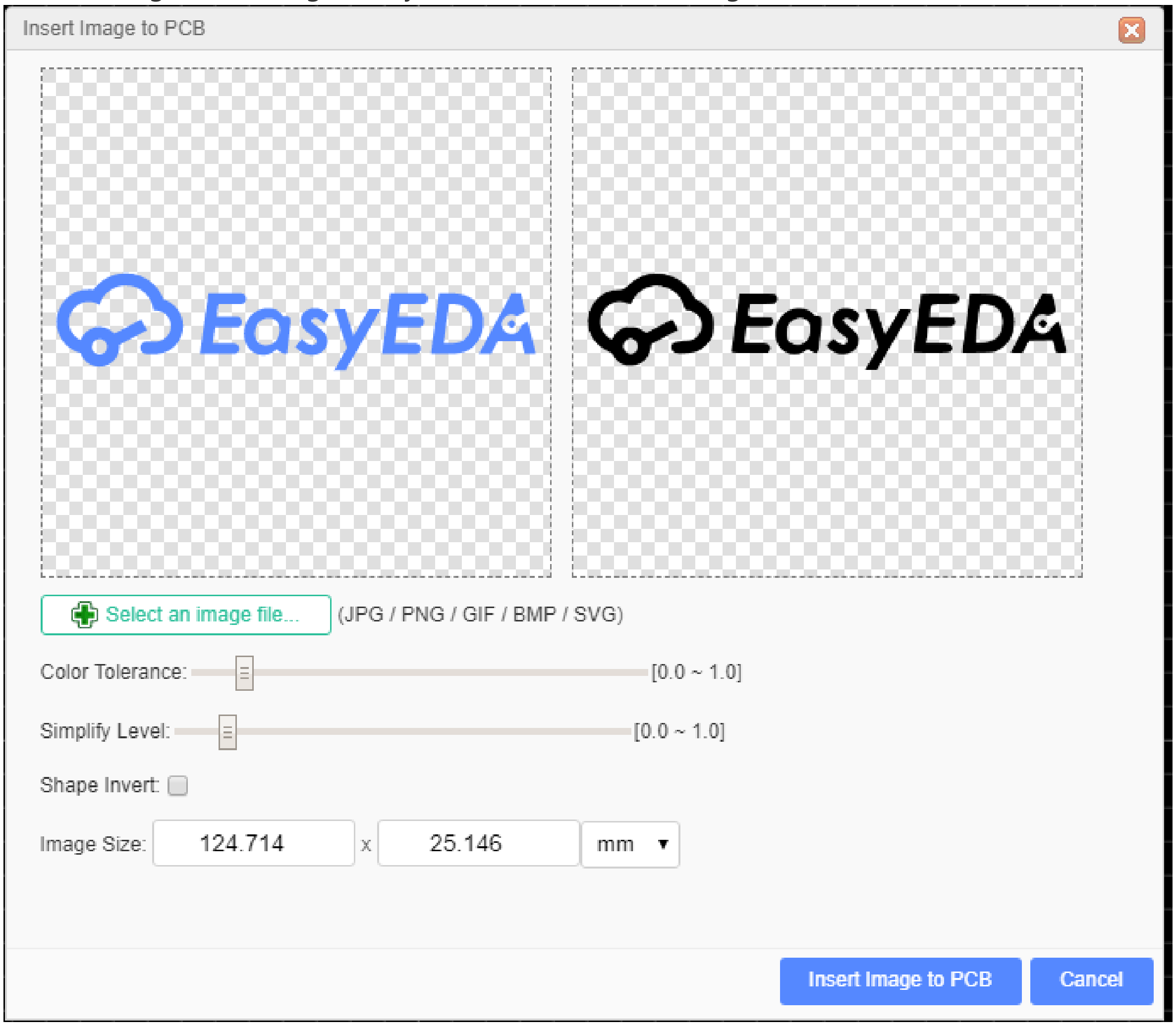
Please note: at present, EasyEDA does not host images, so you need to upload your images to an image sharing site such as http://www.imgur.com.

Import Image to PCB

In the PCB and Footprint editor there is a nice feature on the PCB Tools bar.



After clicking on the image icon you will see the Insert Image window as shown below.



(JPG / PNG / GIF / BMP / SVG)

Color Tolerance: [0.0 ~ 1.0]

Simplify Level: [0.0 ~ 1.0]

Shape Invert: ☐

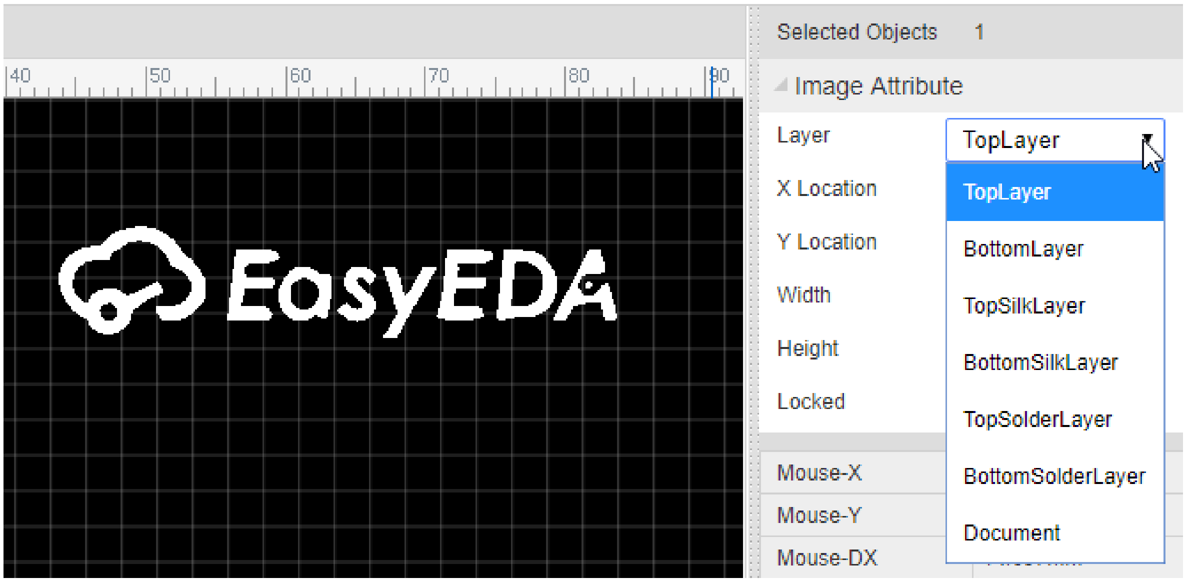
Image Size: 124.714 x 25.146 mm

In this dialog, you can choose your favorite image, EasyEDA support `JPG`, `BMP`, `PNG`, `GIF`, and `SVG`. Unlike some other EDA tools which only support a Monochrome Bitmap image, EasyEDA supports full color, but Monochrome Bitmap is welcome.

You can adjust the color tolerance, simplify levels and reset the image size there.

You can also select shape invert.

The image will be inserted to the active layer, if it is not right, you can change an attribute, such as TopSilkLayer.

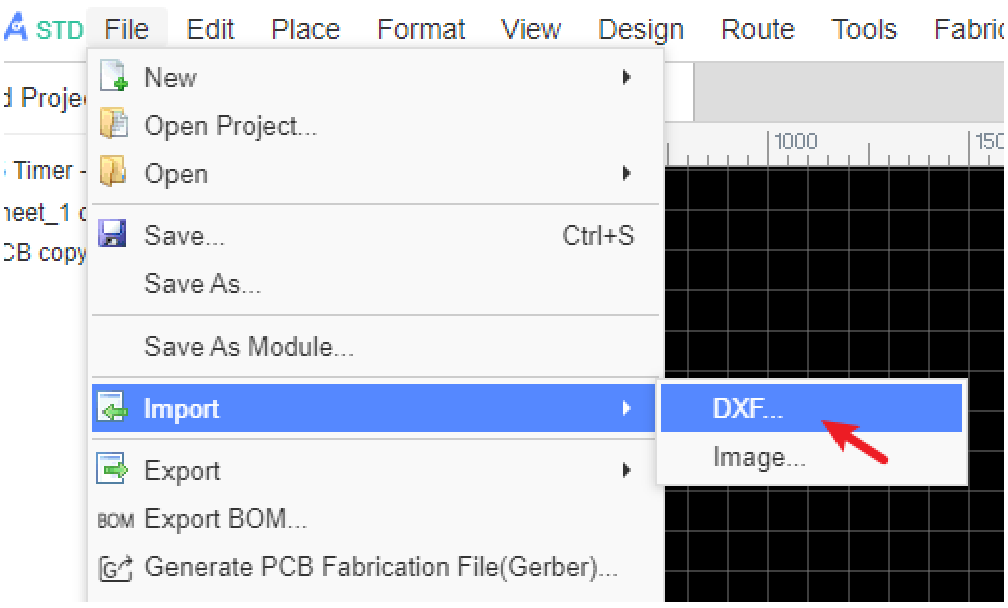


Import DXF File

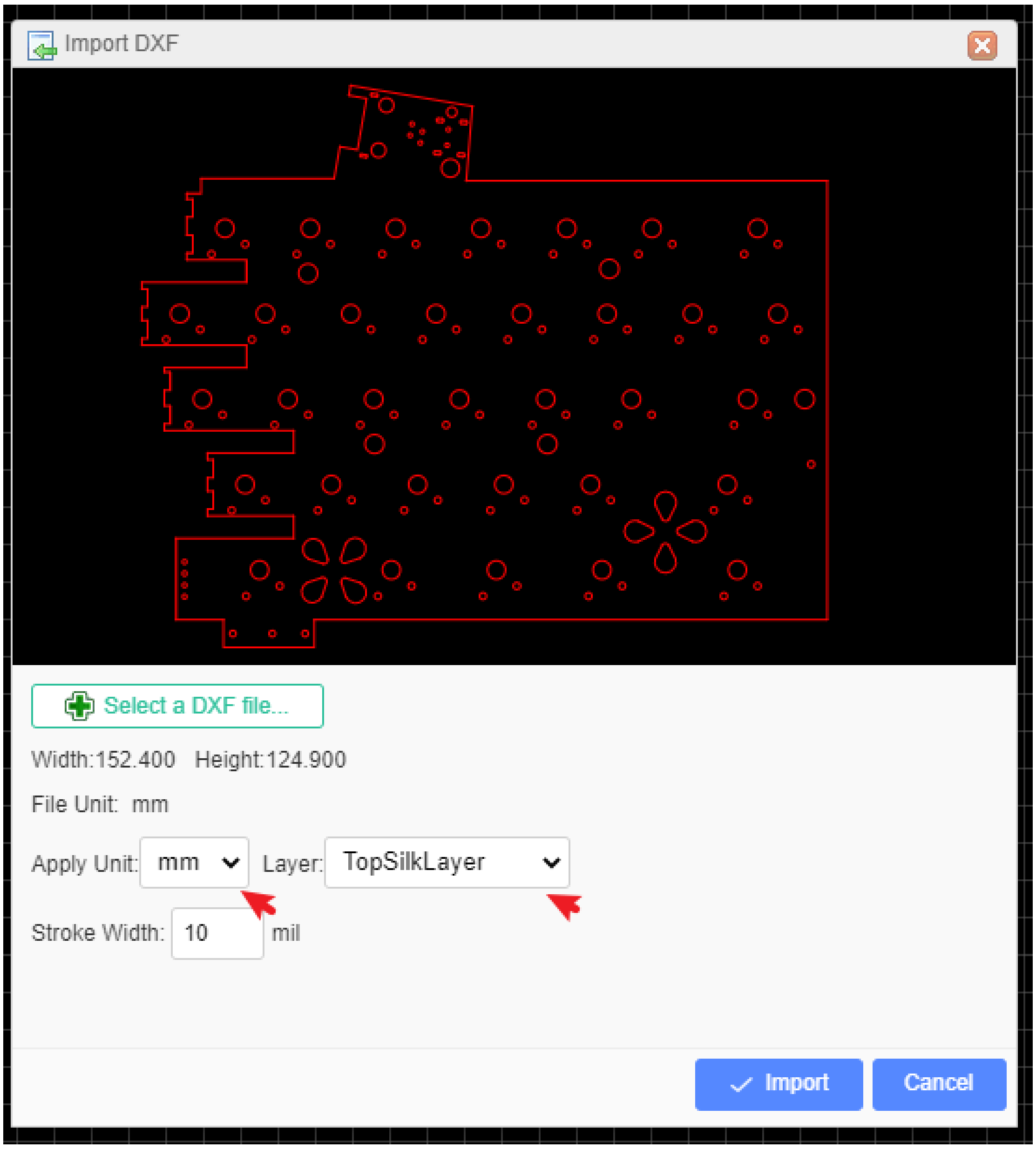
How can you create irregular board outlines or complex board outline in EasyEDA?
This is sometimes needed when you are designing a PCB for an enclosure that may have a curved profile, or other unavoidable mechanical features that must be accounted for in the design.

EasyEDA supports importing DXF into PCB.

Find the import DXF menu under the file menu. Use: File - Import - DXF



After selecting the *.DXF file, you will find a dialog as shown below



Width:152.400 Height:124.900

File Unit: mm

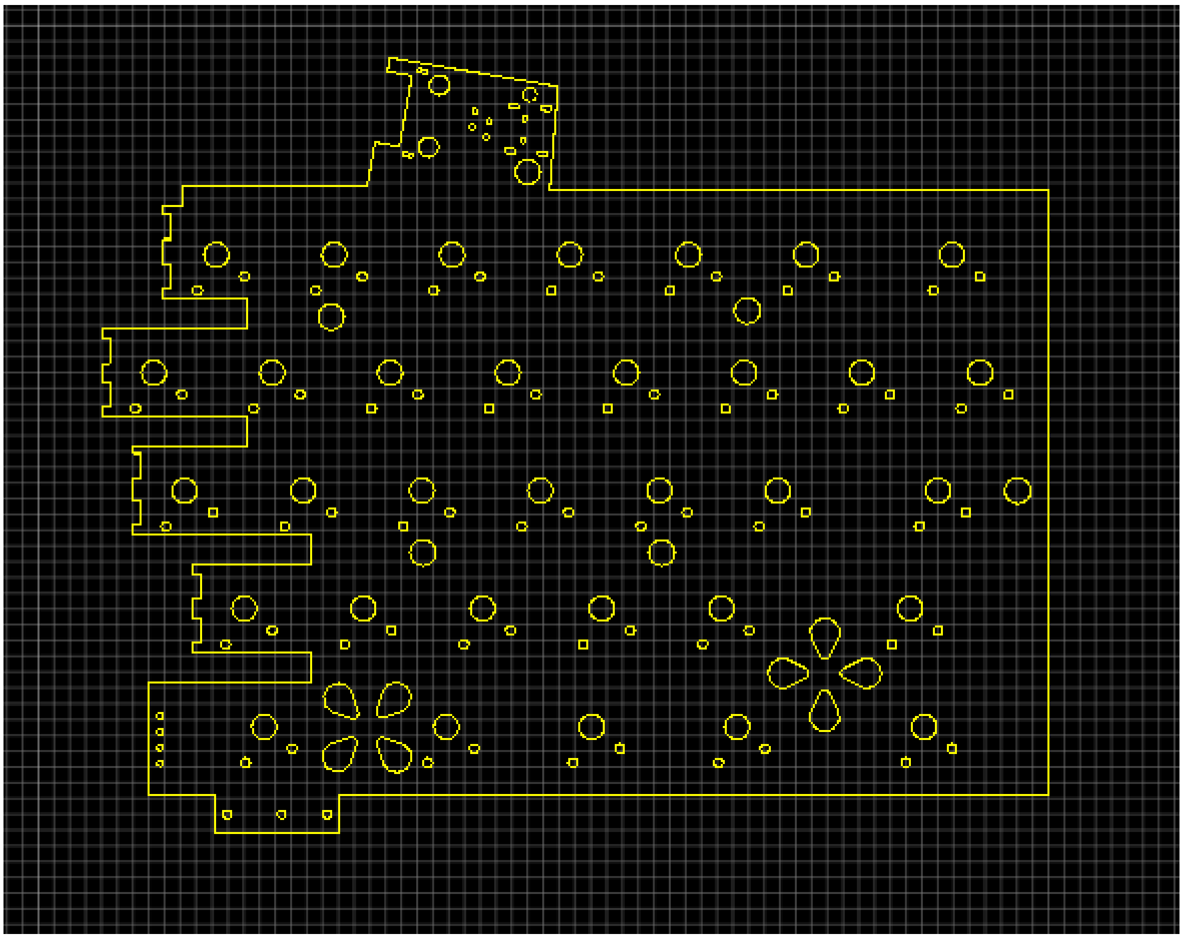
Apply Unit: mm Layer: TopSilkLayer

Stroke Width: 10 mil

EasyEDA provides some options, units(mm, cm, mil, inch), and selecting the layer to which the shapes will be applied.

When importing DXF into schematic or symbols, the units are pixels.

After clicking the import button, you will find them on your PCB canvas.



You can try this to import this example by yourself. [DXF example]

Please note:

1. The file must have a *.dxf filename extension.
2. The circles will be converted to holes if you choose the layer as board outline.
3. There are some items which are not supported, such as Mirror, spiral line etc.
4. If the DXF objects were grouped, please ungroup them before importing.
5. Do not import a large DXF to copper layer directly, it will be very slow and appear to be hung in the editor.

Import Altium Designer

The import function is in beta now, please check your design carefully after importing.

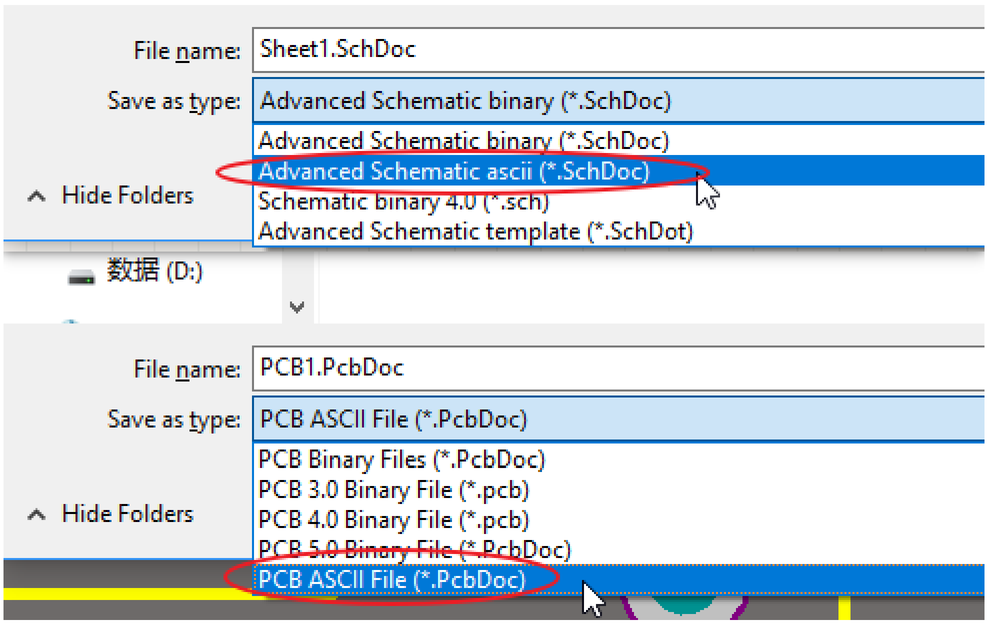
Some design rules are not yet supported by EasyEDA.

Import Schematic and PCB

1. You can import Altium Designer's Schematic and PCB files into EasyEDA but the format must be **ASCII** files, so you need to save the designs as Ascii files like this.

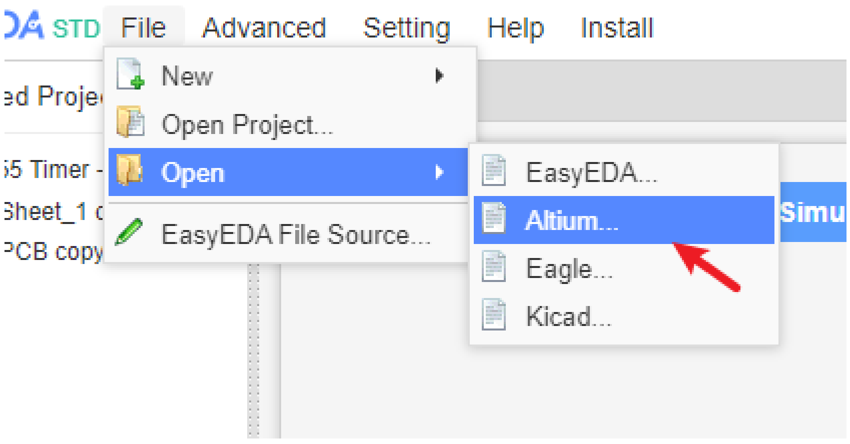
Schematic saved as "Advanced Schematic ascii (*.SchDoc)"

PCB save as "PCB ASCII File (*.PcbDoc)"

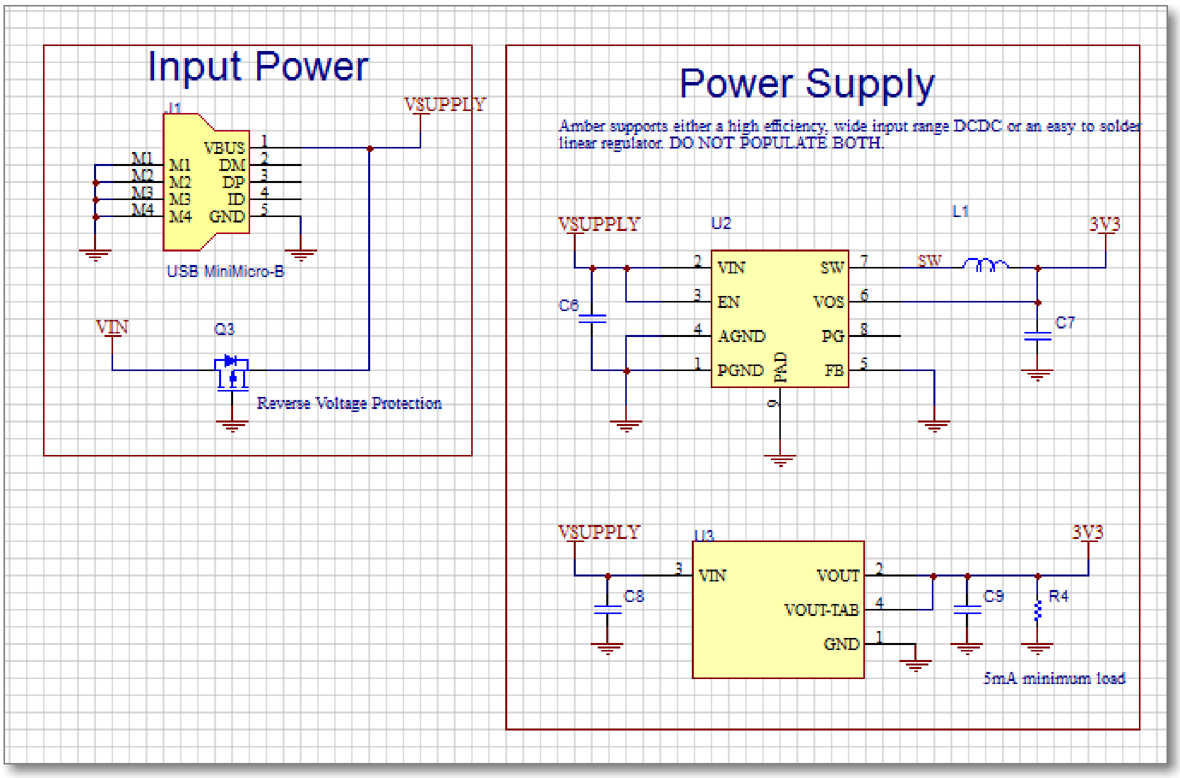

File name: Sheet1.SchDoc
Save as type: Advanced Schematic binary (*.SchDoc)
Advanced Schematic binary (*.SchDoc)
Advanced Schematic ascii (*.SchDoc)
Schematic binary 4.0 (*.sch)
Advanced Schematic template (*.SchDot)

File name: PCB1.PcbDoc
Save as type: PCB ASCII File (*.PcbDoc)
PCB Binary Files (*.PcbDoc)
PCB 3.0 Binary File (*.pcb)
PCB 4.0 Binary File (*.pcb)
PCB 5.0 Binary File (*.PcbDoc)
PCB ASCII File (*.PcbDoc)

Then import it using: File - Open - Altium...



EasyEDA offers an excellent experience in importing Altium Designer's Schematic and PCB as you can see from the image below of a schematic imported from Altium Designer:



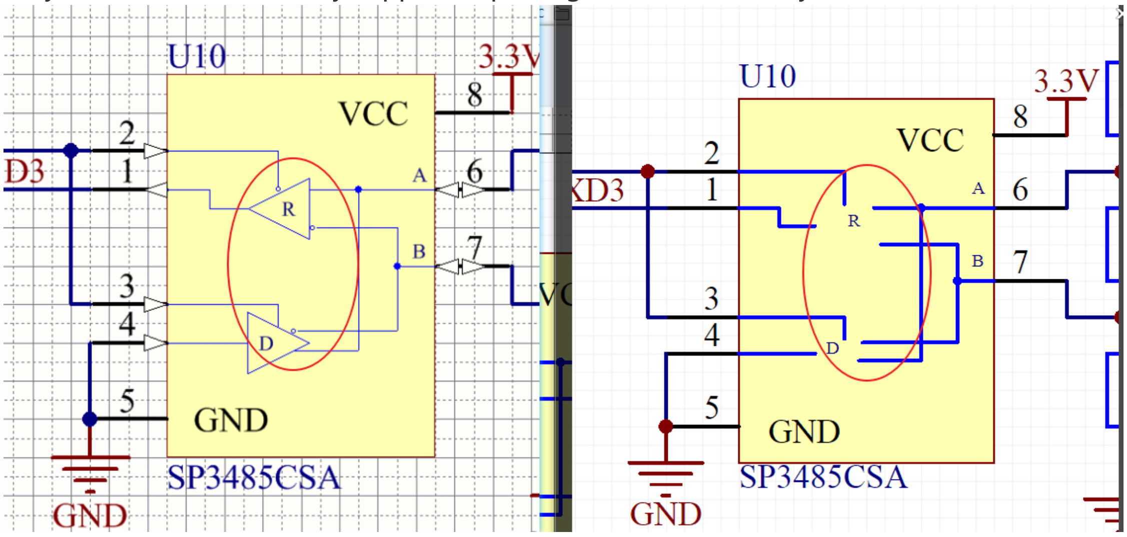
Input Power
Power Supply
Amber supports either a high efficiency, wide input range DCDC or an easy to solder linear regulator. DO NOT POPULATE BOTH.
VSUPPLY
USB MiniMicro-B
VIN
Q3
Reverse Voltage Protection
U2
L1
3V3
U3
5mA minimum load

If your schematic and PCB are Protel 99se format files, please open in Altium Designer and save as ASCII format, and then import them. EasyEDA does not support Protel 99se file format directly.

2. If you import Altium schematic and some text becomes unreadable, please encode your ASCII file with UTF-8.

Notice:

- EasyEDA does not currently support importing Altium PCB rules.
- EasyEDA does not currently support importing Altium PCB inner layers, please modify manually after imported.
- EasyEDA does not currently support importing the Altium IEEE symbol in the schematic.



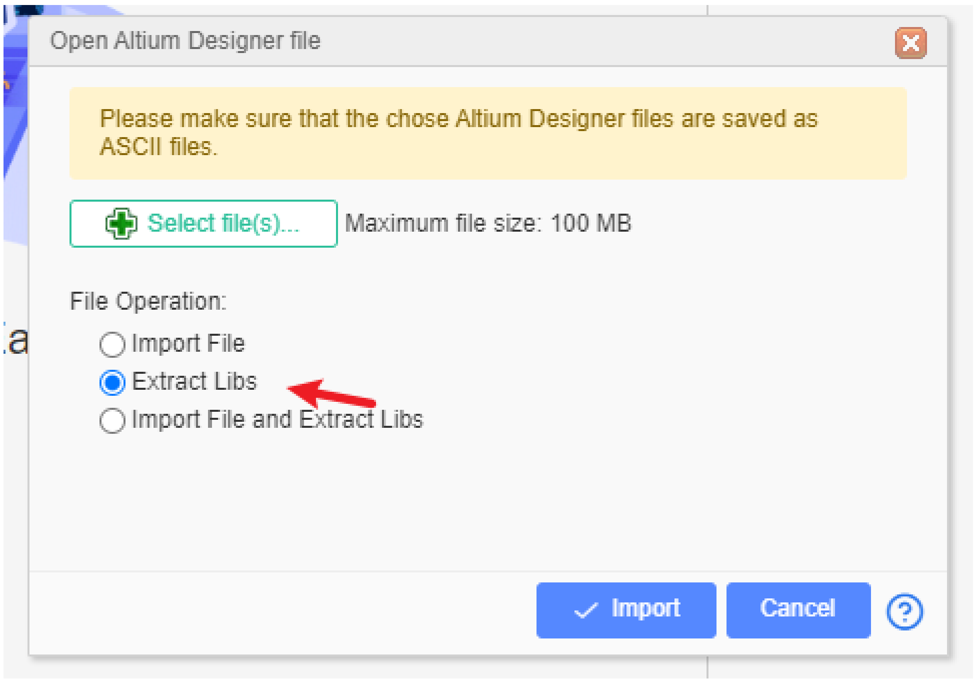
- Please do not export your design to Altium and then import it again as this may cause some details to be lost!!!
- If the Altium file is large it may take a long time to import, removing the copper layer before importing can improve the speed.
- Importing the border processing logic of AD file: The Board Shape turns into a border layer, keep-out turns into a document layer, and Mechanical becomes a mechanical layer.

Import Altium libraries

Altium Designer's Schematic and Footprint libraries are not available as **ASCII** files, EasyEDA can't import them directly, so how can you import them?

In the Import file from your computer dialog to the right of File Operation; tick the `Extract Libs` option and EasyEDA will extract all the libs from the Schematic files or PCB Files.

So, if you want to import Altium Designer's Libs, you can add them to your Altium Designer Schematic or PCB and then extract them again into your EasyEDA library.


Open Altium Designer file

Please make sure that the chose Altium Designer files are saved as ASCII files.

Maximum file size: 100 MB

File Operation:

Import File [✓ Extract Libs] Import File and Extract Libs

Import Eagle

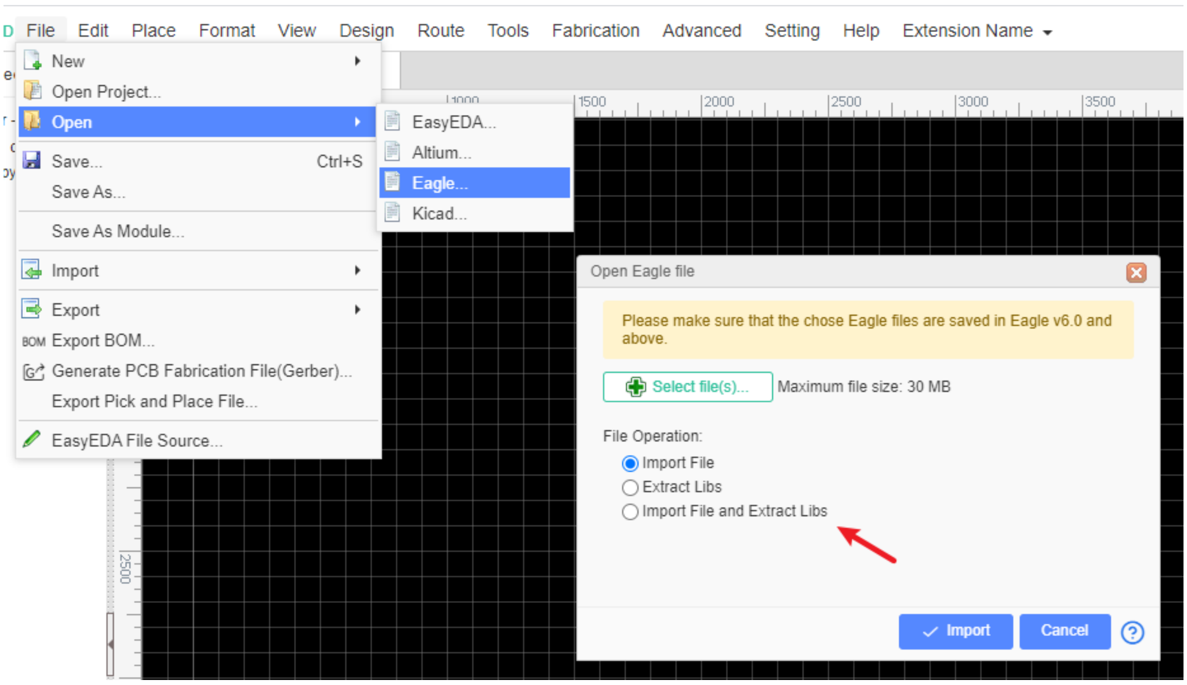
Please refer next section

Import KiCAD

Please refer next section

Import Eagle

Eagle Schematic/PCB/libs can be imported, but EasyEDA only supports version 6 and later (6+) because that was when Version 6 Eagle adopted an **ASCII XML** data structure as their native file format.



Open Eagle file

Please make sure that the chose Eagle files are saved in Eagle v6.0 and above.

Select file(s)... Maximum file size: 30 MB

File Operation:

[✓ Import File]
Extract Libs
Import File and Extract Libs

If your Eagle file can be opened in Eagle, but fails import to EasyEDA, save a copy with the latest Eagle, and then import that one.

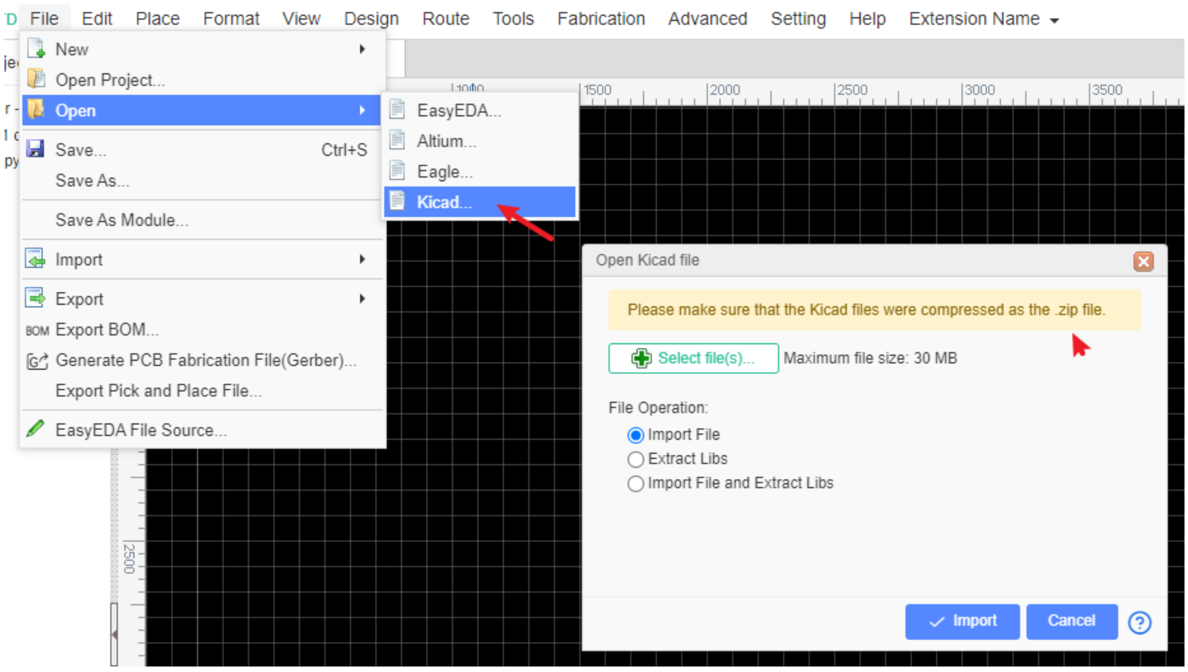
If your file has been saved as a copy from v6.0 or greater, but importing still fails, then edit the Eagle file with a Text Editor. Search the file for garbled characters (something that looks strange), remove them, and try again.

If your schematic needs to update the PCB, please use the "Import File and Extract Libs" option, make sure that all libraries are imported first.

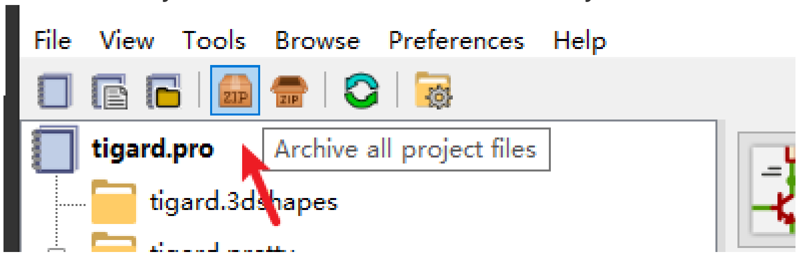
Some rules or primitives are not supported, please check carefully after importing to make sure the design is correct.

Import KiCAD

EasyEDA supports importing KiCAD v4.06 and greater version KiCAD files, if the KiCAD files version is less than v4.06, please open them with the latest KiCAD and save as a new one, and then import them.
The KiCAD project files need to be compressed as zip file before importing.



- If you only want to import the PCB, you just need to ZIP the PCB file and then import it.
- If you want to import the schematic, you must ZIP the schematic and symbols together, we suggest using the KiCAD archive tool when opening the project in KiCAD, it will include the symbols in the ZIP file automatically.



Notice

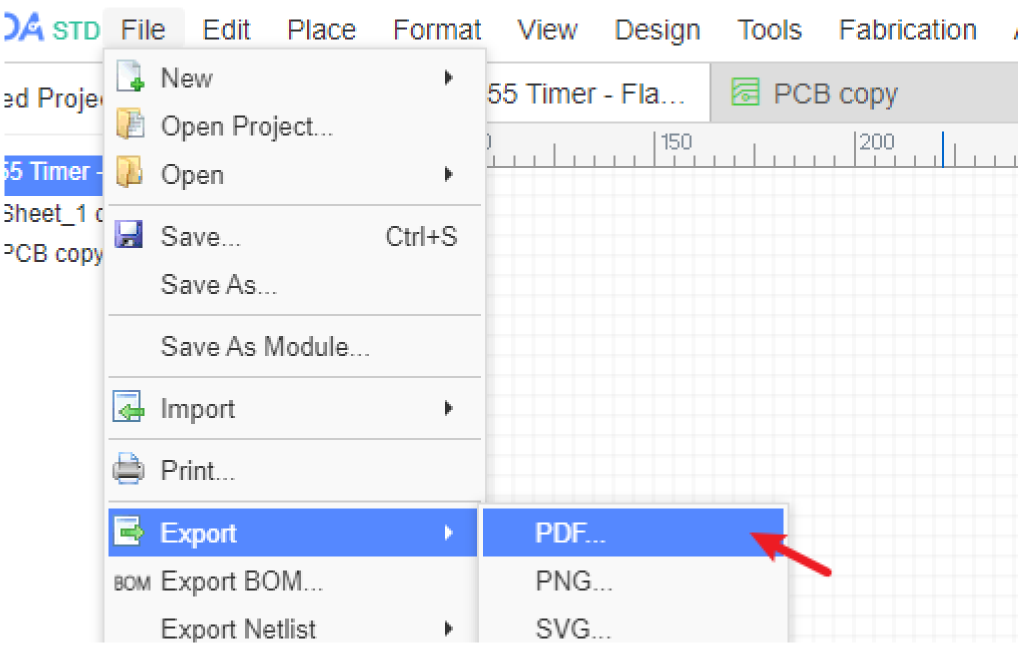
- For the KiCAD special symbols such as Power symbol (Power Flag(PWR_FLAG)), EasyEDA will convert them as the symbol not Netflag, you can delete them if you don't need them.
- The PCB design rule is not supported yet.
- KiCAD has updated the document format since KiCad v5.1.3, if the import fails, please try a previous version. A fix for this is being developed.

Export

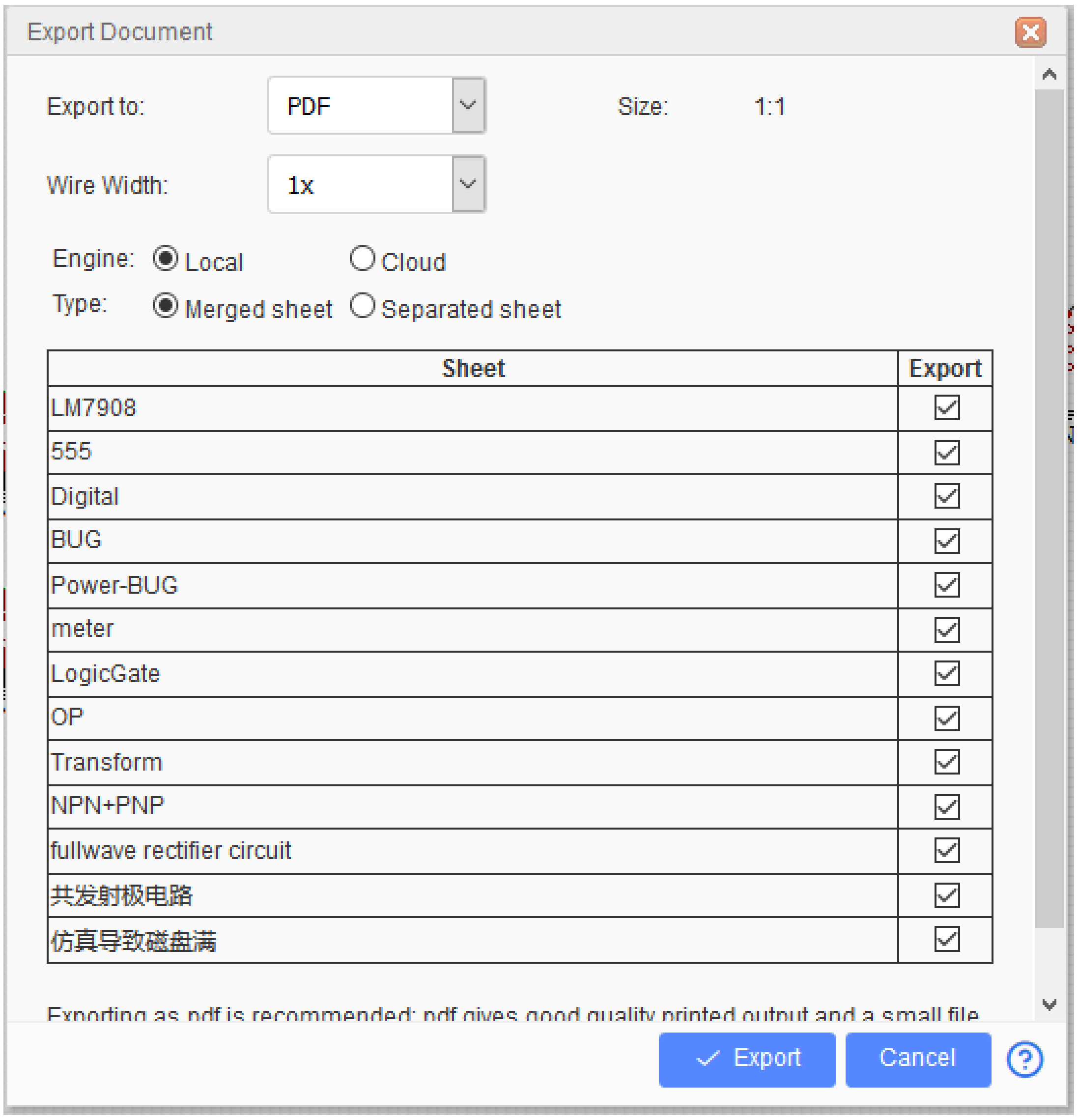
Export Schematics

Export Schematics in PDF/PNG/SVG

Use: ** > File > Export > PDF/PNG/SVG...**



will open this dialog:



Export to: PDF

Size: 1:1

Wire Width: 1x

Engine: [✓ Local] Cloud

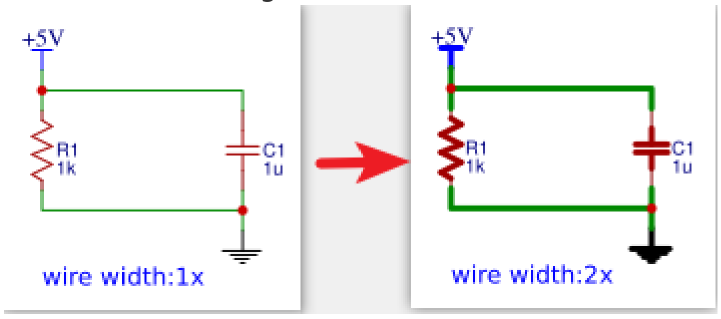
Type: [✓ Merged sheet] Separated sheet

Sheet	Export
LM7908	✓
555	✓
Digital	✓
BUG	✓
Power-BUG	✓
meter	✓
LogicGate	✓
OP	✓
Transform	✓
NPN+PNP	✓
fullwave rectifier circuit	✓
共发射极电路	✓
仿真导致磁盘满	✓

Exporting as pdf is recommended: pdf gives good quality printed output and a small file

- **Export to**: This allows you to export your design to SVG, PNG and PDF file format.
- **Size**: Only for PNG and SVG. This is the width of the image, for example, when you set size as 1x, the exported PNG's width is 600 pixels. If you set the size as 2x the width will be 1200 pixels in the exported PNG.

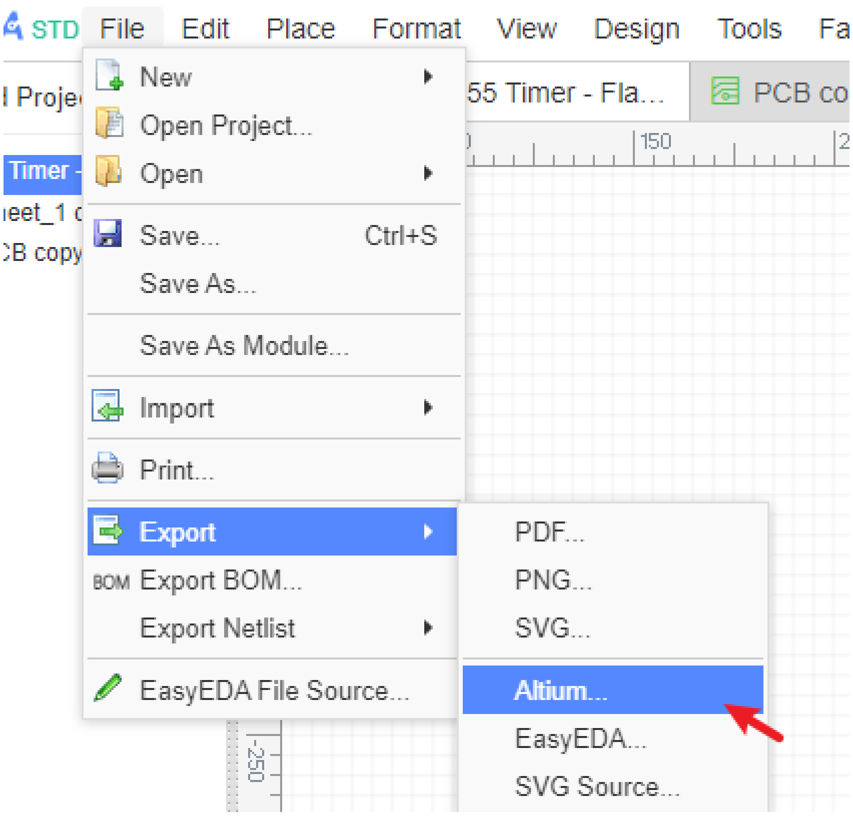
- **Wire Width**: If you set this to 2x, the line width will be enlarged two times. This is illustrated in the image below.



- **Engine**:
    - **Local**: PDF generated by Editor
    - **Cloud**: PDF generated by the Cloud Server, this feature will likely soon be removed from EasyEDA.
- **Type**:
    - **Merged sheet**: If your schematic has multiple sheets it will be merged into one file.
    - **Separated sheet**: If your schematic has multiple sheets each sheet will be created in a separate file.

Export Schematics in Altium Designer Format

EasyEDA supports exporting the schematics in Altium Designer format.
Use **"File > Export > Altium..."**, and click the **"Download now"**. A `.schdoc` file will be created.



The more information please refer to Export Altium

Download Schematics

Please refer to Export EasyEDA Source

Print Schematic

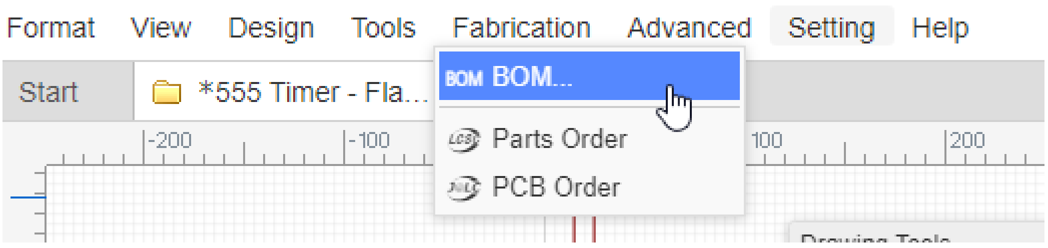
The command **File** > **Print**, will open a web page with the print dialog which is used to print the schematic.

If you are using EasyEDA Client you need to take note of the edge of the sheet, if it is past the page edge it is recommended to export to PDF and print.

If you are using Chrome, you will need to adjust the print options to fit to the page size and disable the option for "Headers and footers". Without this the A4 size will print an extra page.

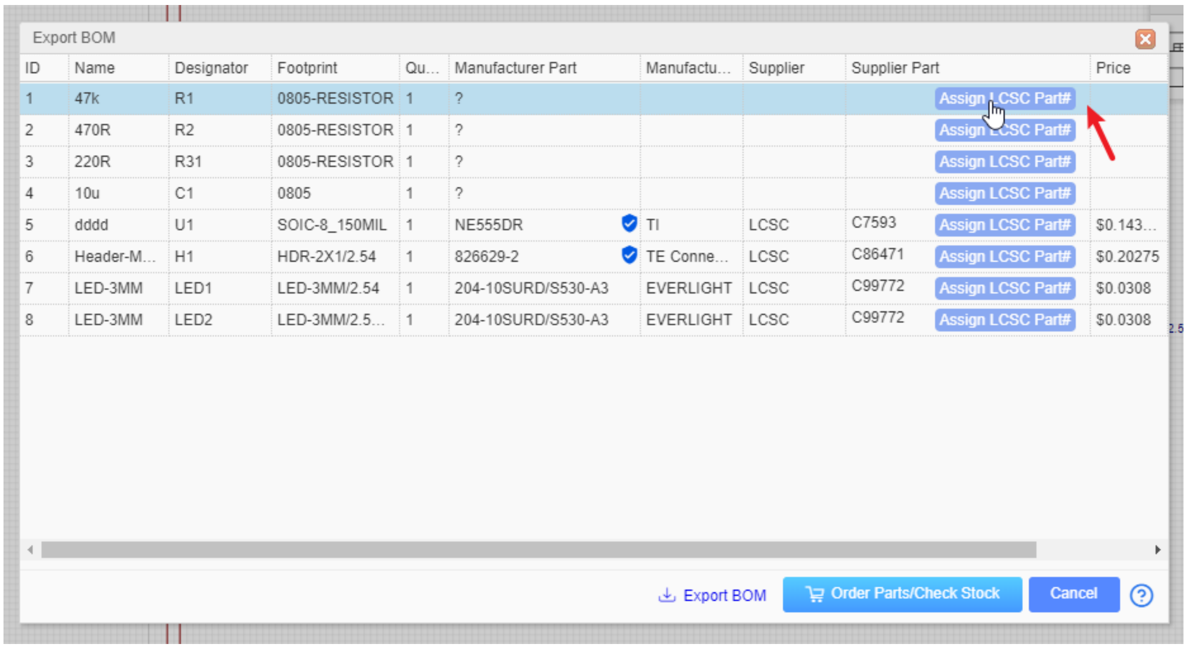
Export BOM

You can export the Bill of Materials (BOM) for the schematic (Document) and PCB, via: "Top Menu - File - Export BOM", or "Top Menu - Fabriaction - BOM".



After clicking the BOM export option, the dialog below will open.

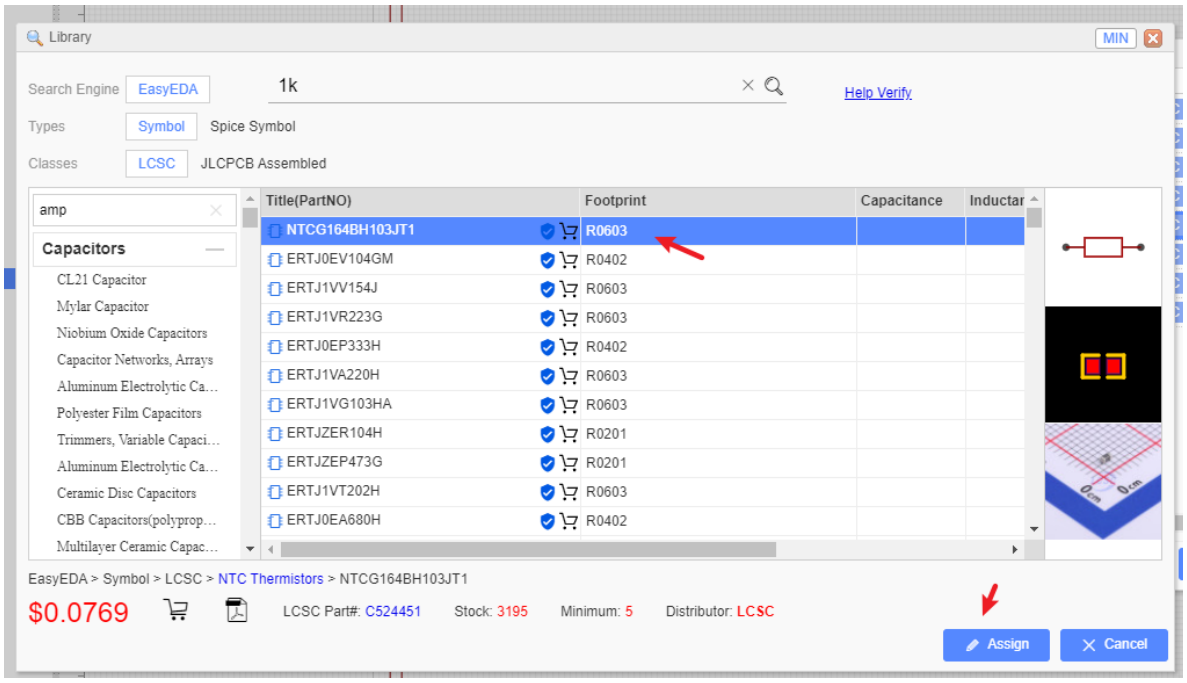
In this dialog，you can click the buttom to assign LCSC part's order code for your components.



Export BOM

ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part	Price
1	47k	R1	0805-RESISTOR	1	?			Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?			Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?			Assign LCSC Part#	
4	10u	C1	0805	1	?			Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593 Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471 Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772 Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772 Assign LCSC Part#	$0.0308

After clicking on the assign icon，the components and footprints search dialog will pop up, and you can choose which component you want to assign.

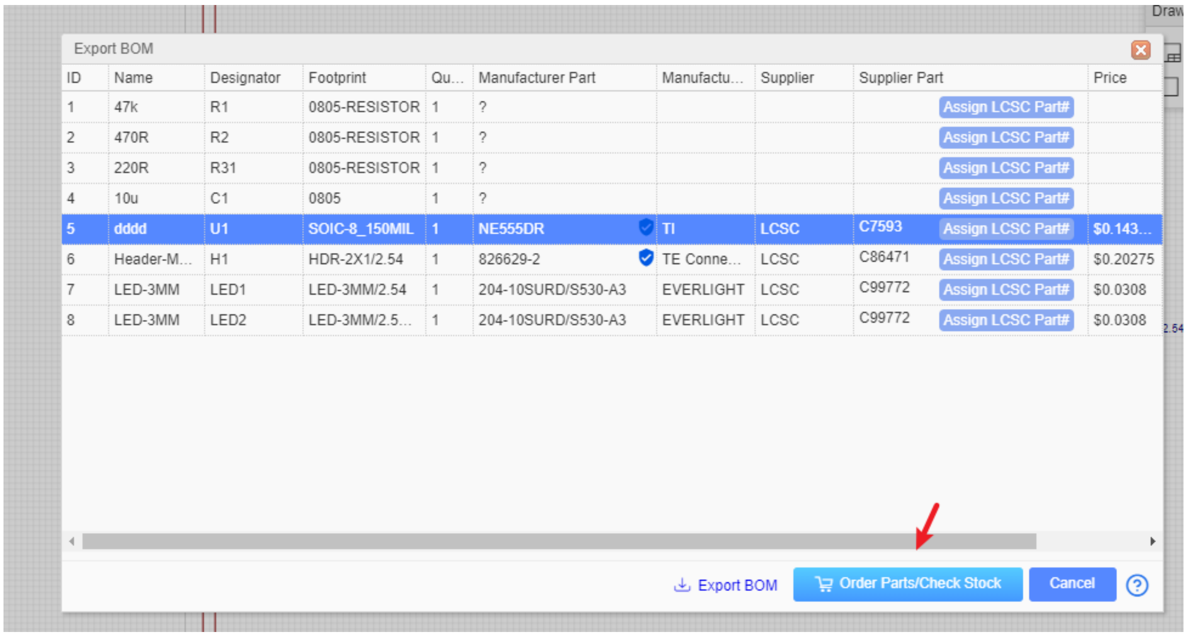

Library
MIN
Search Engine EasyEDA
1k
Help Verify
Types Symbol Spice Symbol
Classes LCSC JLCPCB Assembled
amp
Capacitors
CL21 Capacitor
Mylar Capacitor
Niobium Oxide Capacitors
Capacitor Networks, Arrays
Aluminum Electrolytic Ca...
Polyester Film Capacitors
Trimmers, Variable Capaci...
Aluminum Electrolytic Ca...
Ceramic Disc Capacitors
CBB Capacitors(polyprop...
Multilayer Ceramic Capac...
Assign
Cancel

Title(PartNO)	Footprint	Capacitance	Inductar
NTCG164BH103JT1	R0603		
ERTJ0EV104GM	R0402		
ERTJ1VV154J	R0603		
ERTJ1VR223G	R0603		
ERTJ0EP333H	R0402		
ERTJ1VA220H	R0603		
ERTJ1VG103HA	R0603		
ERTJZER104H	R0201		
ERTJZEP473G	R0201		
ERTJ1VT202H	R0603		
ERTJ0EA680H	R0402		

EasyEDA > Symbol > LCSC > NTC Thermistors > NTCG164BH103JT1

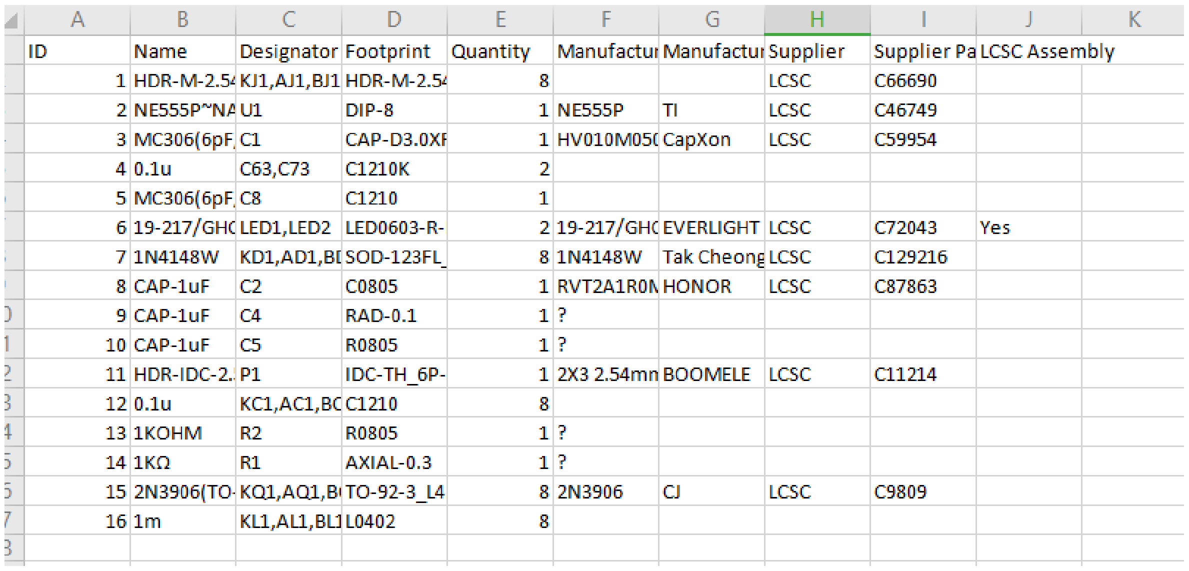
$0.0769 LCSC Part#: C524451 Stock: 3195 Minimum: 5 Distributor: LCSC

When you click the "Order Parts/Check Stock" button, we will help you to list all the components of your BOM at LCSC.com(If you haven't login LCSC, you have to login first). If you want to buy the components form LCSC, and you just need to put them to the cart and check out.



ID	Name	Designator	Footprint	Qu...	Manufacturer Part	Manufactu...	Supplier	Supplier Part		Price
1	47k	R1	0805-RESISTOR	1	?				Assign LCSC Part#	
2	470R	R2	0805-RESISTOR	1	?				Assign LCSC Part#	
3	220R	R31	0805-RESISTOR	1	?				Assign LCSC Part#	
4	10u	C1	0805	1	?				Assign LCSC Part#	
5	dddd	U1	SOIC-8_150MIL	1	NE555DR	TI	LCSC	C7593	Assign LCSC Part#	$0.143...
6	Header-M...	H1	HDR-2X1/2.54	1	826629-2	TE Conne...	LCSC	C86471	Assign LCSC Part#	$0.20275
7	LED-3MM	LED1	LED-3MM/2.54	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308
8	LED-3MM	LED2	LED-3MM/2.5...	1	204-10SURD/S530-A3	EVERLIGHT	LCSC	C99772	Assign LCSC Part#	$0.0308

You can open the BOM in any text editor or spreadsheet.



ID	Name	Designator	Footprint	Quantity	Manufactur	Manufactur	Supplier	Supplier Pa	LCSC Assembly	
1	HDR-M-2.54	KJ1,AJ1,BJ1	HDR-M-2.54	8			LCSC	C66690		
2	NE555P~NA	U1	DIP-8	1	NE555P	TI	LCSC	C46749		
3	MC306(6pF,	C1	CAP-D3.0XF	1	HV010M050	CapXon	LCSC	C59954		
4	0.1u	C63,C73	C1210K	2						
5	MC306(6pF,	C8	C1210	1						
6	19-217/GHC	LED1,LED2	LED0603-R-	2	19-217/GHC	EVERLIGHT	LCSC	C72043	Yes	
7	1N4148W	KD1,AD1,BD	SOD-123FL_	8	1N4148W	Tak Cheong	LCSC	C129216		
8	CAP-1uF	C2	C0805	1	RVT2A1R0M	HONOR	LCSC	C87863		
9	CAP-1uF	C4	RAD-0.1	1	?					
10	CAP-1uF	C5	R0805	1	?					
11	HDR-IDC-2.	P1	IDC-TH_6P-	1	2X3 2.54mm	BOOMELE	LCSC	C11214		
12	0.1u	KC1,AC1,BC	C1210	8						
13	1KOHM	R2	R0805	1	?					
14	1KΩ	R1	AXIAL-0.3	1	?					
15	2N3906(TO-	KQ1,AQ1,BQ	TO-92-3_L4	8	2N3906	CJ	LCSC	C9809		
16	1m	KL1,AL1,BL1	L0402	8						

Export BOM supports to export LCSC part price, it is the same as LCSC website.

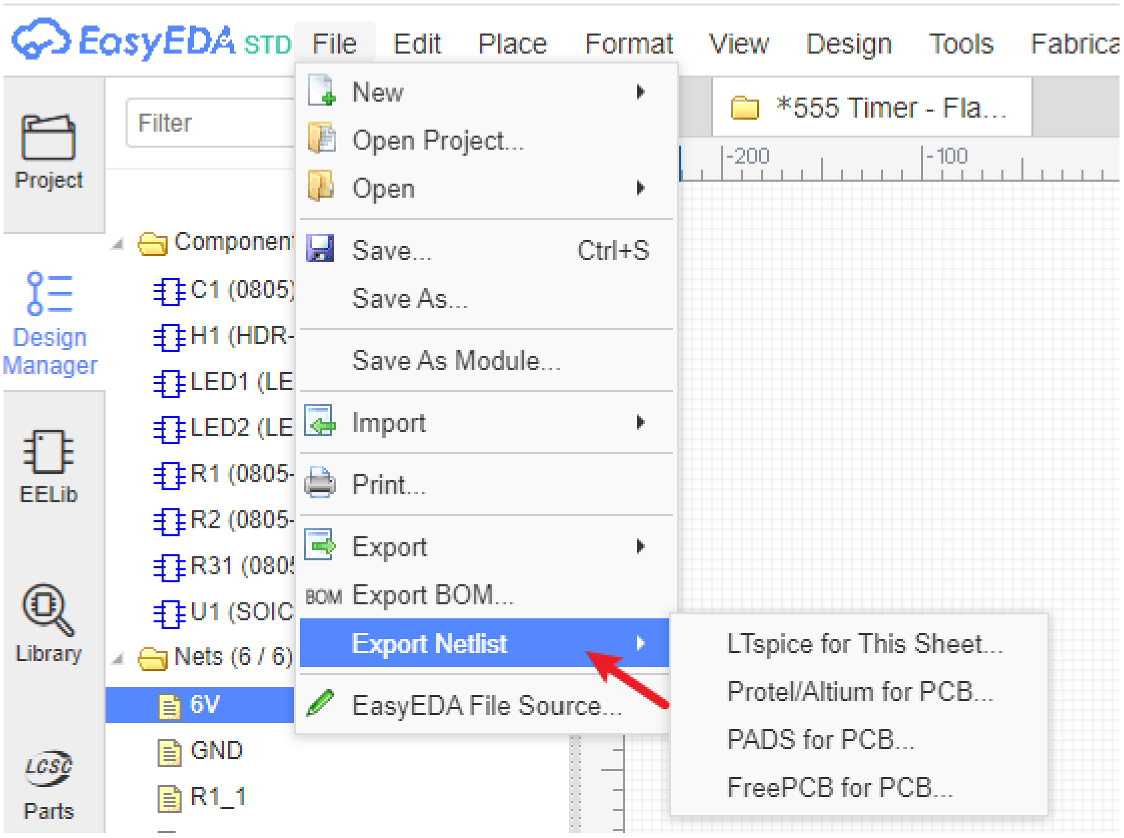
Notice:

- Before v6.4.17, If your project has schematic and PCB, the BOM data will come from schematic; if the project only has PCB, the BOM data will come from PCB.
- Since v6.4.17, the schematic BOM and PCB BOM are separated. If you assign the LCSC part at the PCB, it will not modify the schematic.
- In order to support multiple languages, BOM and coordinate files (CSV file) are UNICODE encoded and tab-based. If the CSV file cannot be read by your components vendor or PCB manufacturer, please convert the encoding and change the delimiter.
- Recommended solution: Save as a new CSV file in Excel or WPS. For example, open a CSV file in Excel, click or select: Save As - Other Formats - CSV (Comma Separated) (*. csv). You can also open the CSV file with any text editor (such as Windows Notepad) and save as ANSI or UTF-8 encoding. If necessary, replace all tabs with commas.

Export NetList

EasyEDA can export the netlist for the whole active project:

File > Export NetList > Spice...



EasyEDA can export a netlist in a variety of formats:

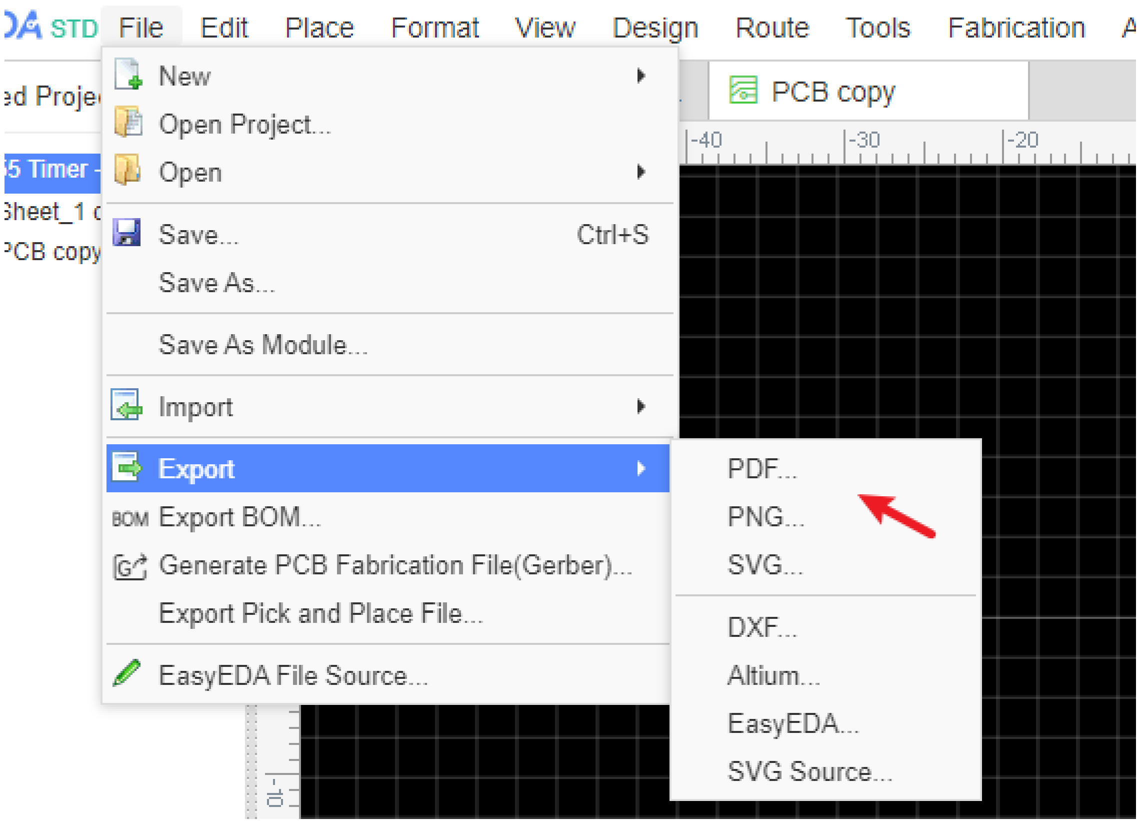
- **LTSpice for this Sheet**: this is a Spice compatible netlist generated by the simulation engine of EasyEDA, It is not normally used as the basis for as a PCB layout.
- **Protel/Altium for PCB**: a PCB netlist in a format that can be imported straight into Altium Designer and it's predecessor, Protel.
- **PADS for PCB**: a PCB netlist in a format that can be imported straight into Pads PCB layout tools.
- **FreePCB for PCB**: a PCB netlist in a format that can be imported straight into FreePCB, a free, open source PCB editor for Windows.

Export PCB

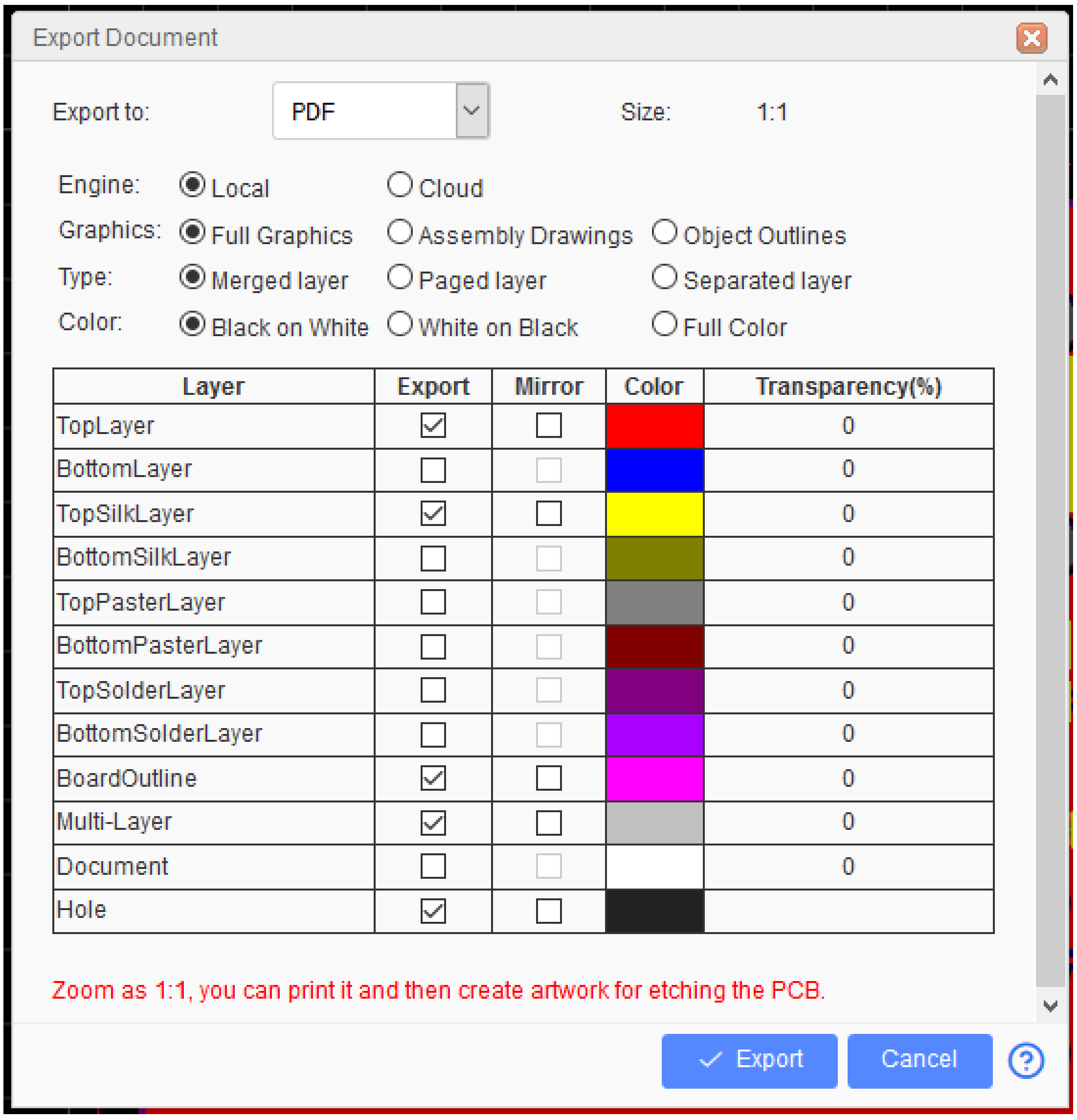
Export PCB in PDF/PNG/SVG

Exporting a PCB design or footprint from EasyEDA is very similar to exporting a Schematic or a Symbol.

Use: **File** > **Export** > **PDF/PNG/SVG...**



This dialog will open:



Export Document

Export to: PDF   Size: 1:1

Engine: [✓ Local] Cloud

Graphics: [✓ Full Graphics] Assembly Drawings Object Outlines

Type: [✓ Merged layer] Paged layer Separated layer

Color: [✓ Black on White] White on Black Full Color

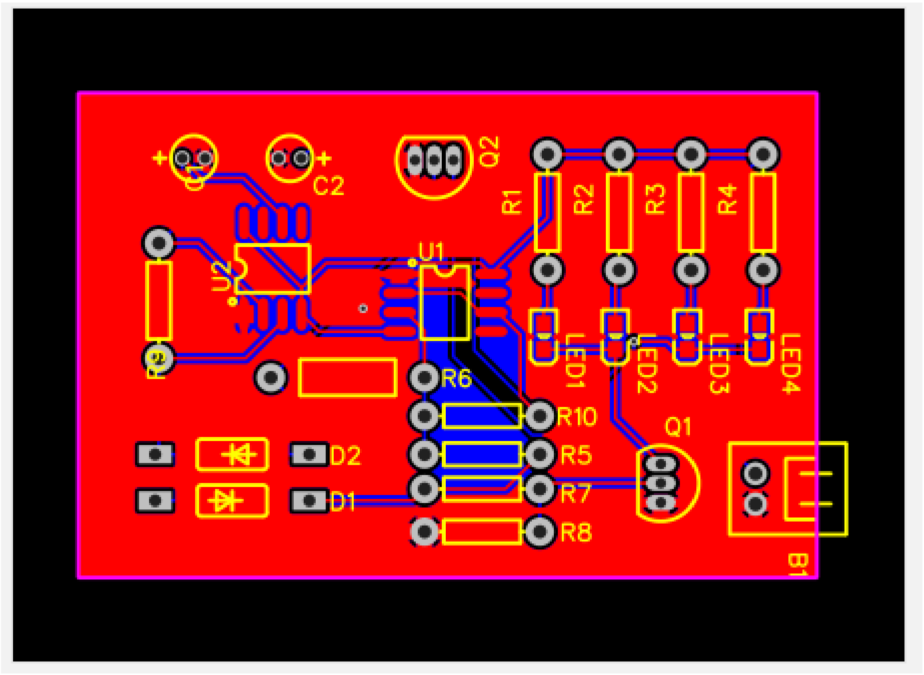
Layer	Export	Mirror	Color	Transparency(%)
TopLayer	☑	☐	🟥	0
BottomLayer	☐	☐	🟦	0
TopSilkLayer	☑	☐	🟨	0
BottomSilkLayer	☐	☐	🟫	0
TopPasterLayer	☐	☐	⬜	0
BottomPasterLayer	☐	☐	🟫	0
TopSolderLayer	☐	☐	🟪	0
BottomSolderLayer	☐	☐	🟪	0
BoardOutline	☑	☐	🟪	0
Multi-Layer	☑	☐	⬜	0
Document	☐	☐	⬜	0
Hole	☑	☐	⬛	

Zoom as 1:1, you can print it and then create artwork for etching the PCB.

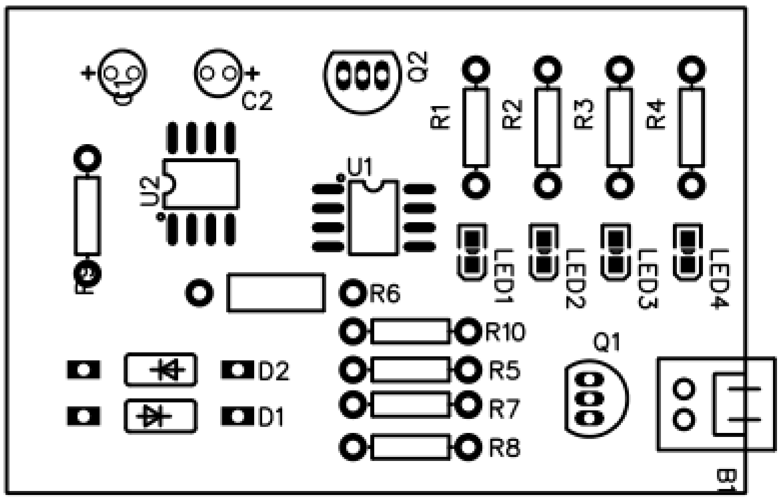
You can choose to export in PDF, PNG or SVG format.

Note: *The PDF size is zoom as 1:1 with PCB. *

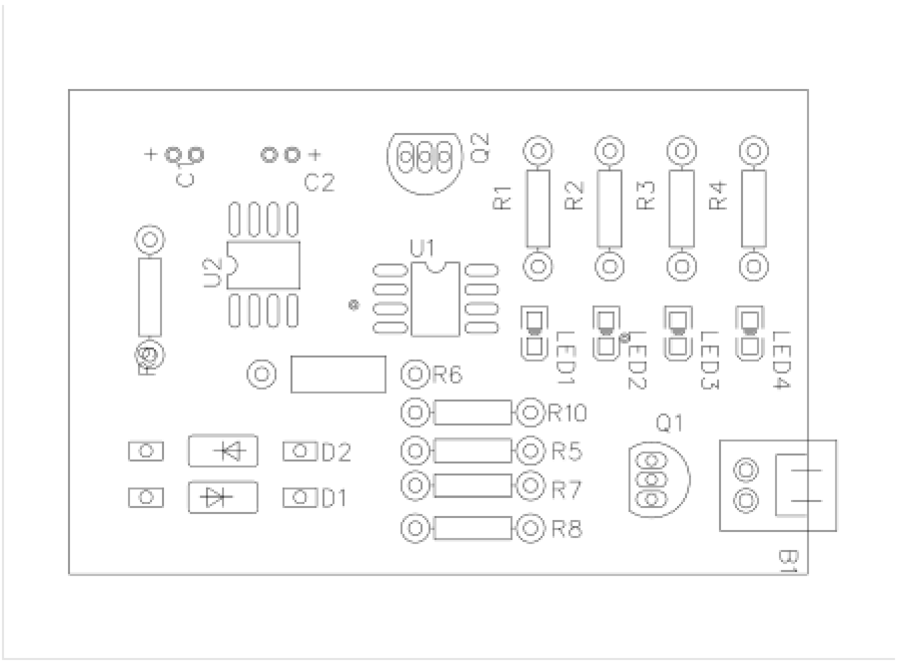
- **Export to**: Support export to PDF, PNG, SVG. To print the PCB as 1:1 you should choose PDF.
- **Engine**:
  - **Local**: PDF generated by Editor
  - **Cloud**: PDF generated by Cloud Server, this feature is planned to be removed in the future.
- **Graphics**:
  - **Full Graphics**: All graphics and objects will be exported.



  - **Assembly Drawings**: This exports only component prefixes, locations, holes, etc. This is used for parts assembly.



  - **Object Outline**: This only exports the objects outlines, such as Pad and silkscreen outlines.



- **Type**:
    - **Merged layer**: All selected layers will be merged into one page.
    - **Paged layer**: All selected layers will be separate pages in a single file.
    - **Separated layer**: All selected layers will be exported into separate files contained in a ZIP file.
- **Color**: You can choose "Black on White", "White on Black", "Full Color".
- **Layer**: You can select to print individual layers or selected layers merged into a single file.
- **Mirror**: It is also possible to mirror selected layers so that the bottom layers are shown in easily readable orientation. This is recommended when all your selected layers are bottom ones.

If EasyEDA PDF does not satisfy your requirements, please let us know.

support@easyeda.com

If you generated the Gerber file, you can use the Gerbv application to easily export the PDF file.

Use: Gerbv

Export PCB in Altium Designer Format

The more information please refer at Export Altium

Download PCB

Please refer to Export EasyEDA Source

Print PCB and Etching

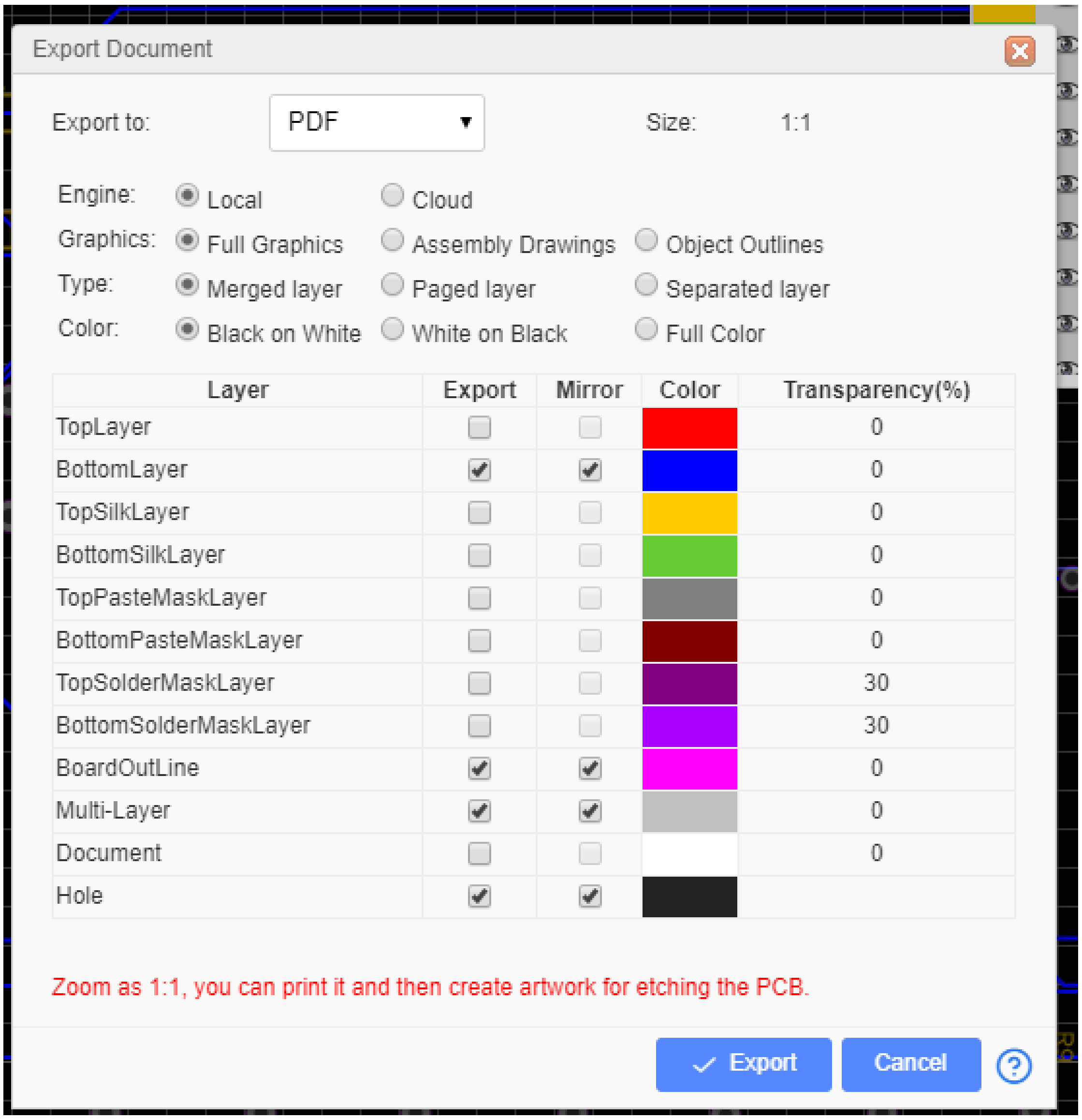
EasyEDA doesn't support directly printing the PCB, instead export to PDF and then print.

If you don't want to order your PCBs from EasyEDA then maybe - for single and double sided PCB designs - you can try using some homemade PCB tech:

http://hackaday.com/2012/12/10/10-ways-to-etch-pcbs-at-home/

So, here's how you can print your PCB layer by layer and then etch it onto a PCB.

Step 1) Export it to PDF, Using: **File > Export > PDF...**



Export to: PDF   Size: 1:1

Engine: [✓ Local] Cloud

Graphics: [✓ Full Graphics] Assembly Drawings Object Outlines

Type: [✓ Merged layer] Paged layer Separated layer

Color: [✓ Black on White] White on Black Full Color

Layer	Export	Mirror	Color	Transparency(%)
TopLayer	☐	☐	🟥	0
BottomLayer	✓	✓	🟦	0
TopSilkLayer	☐	☐	🟨	0
BottomSilkLayer	☐	☐	🟩	0
TopPasteMaskLayer	☐	☐	⬛	0
BottomPasteMaskLayer	☐	☐	🟫	0
TopSolderMaskLayer	☐	☐	🟪	30
BottomSolderMaskLayer	☐	☐	🟪	30
BoardOutLine	✓	✓	🟪	0
Multi-Layer	✓	✓	⬜	0
Document	☐	☐	⬜	0
Hole	✓	✓	⬛	

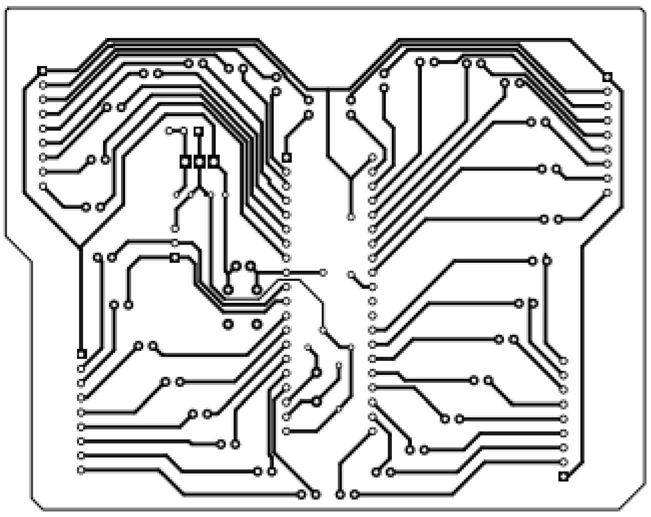
Zoom as 1:1, you can print it and then create artwork for etching the PCB.

Note: *Make sure the Colour is Black on White Background.*

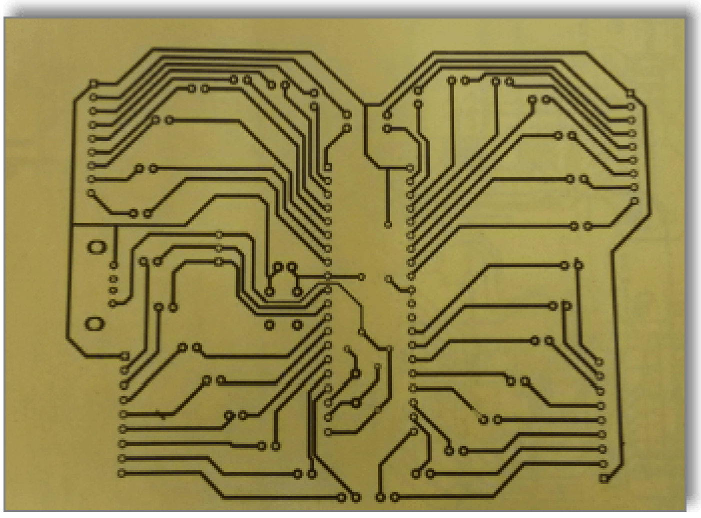
For the bottom layer, select the mirror option on export if needed.

If you have routed PCB tracks on the top layer you will also need to export that layer. Etched PCBs generally need to use the mirror export printing option.

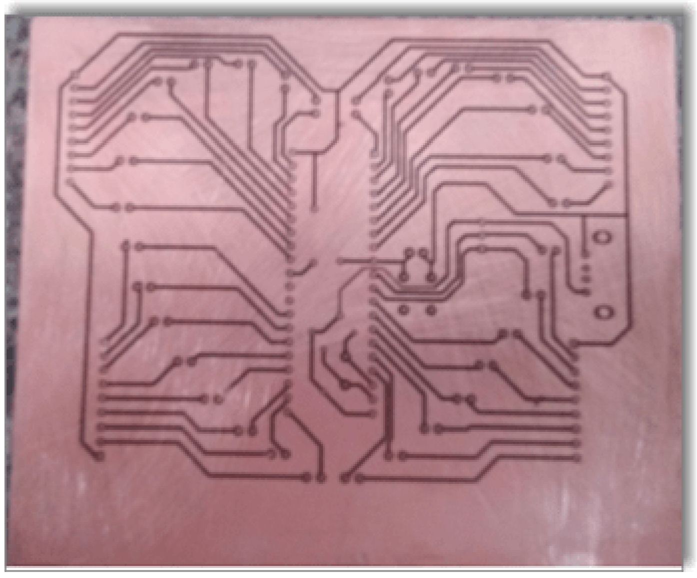
Step 2) Open the pdf file in a viewer



Step 3) Print it to paper

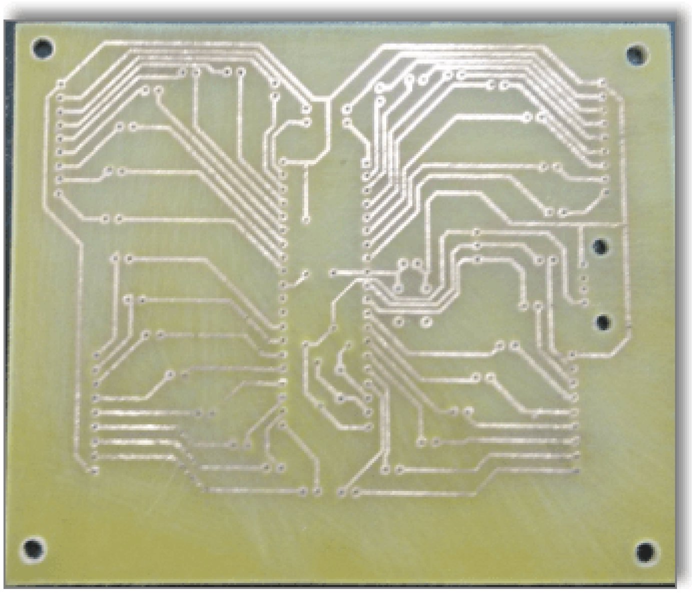


Step 4) Copy it to the copper

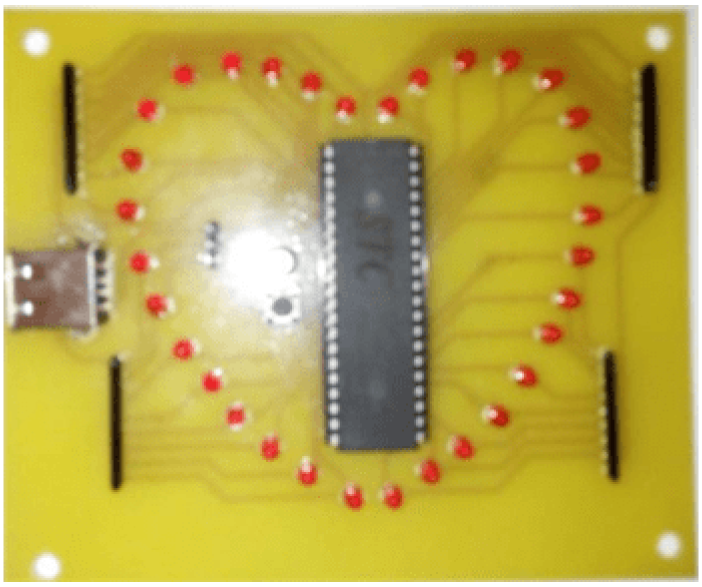


Step 5) Etch it

Step 6) Drill it



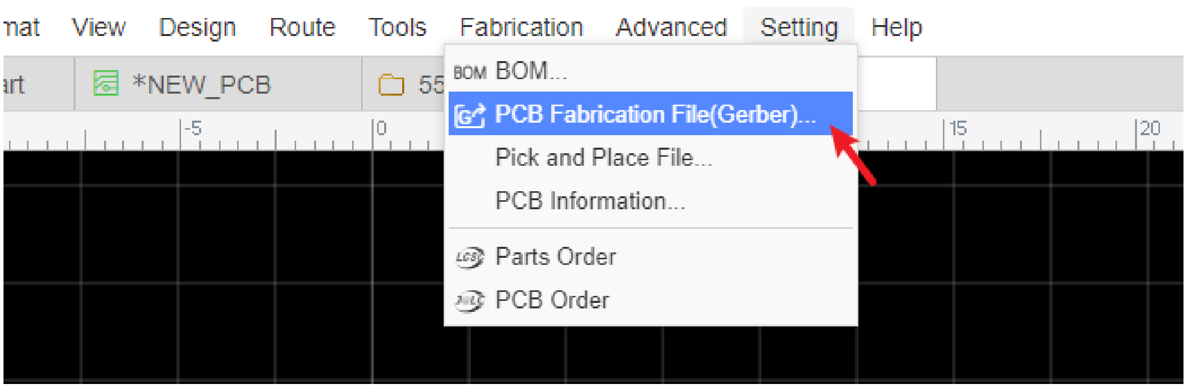
Step 7) Get your soldering iron out!



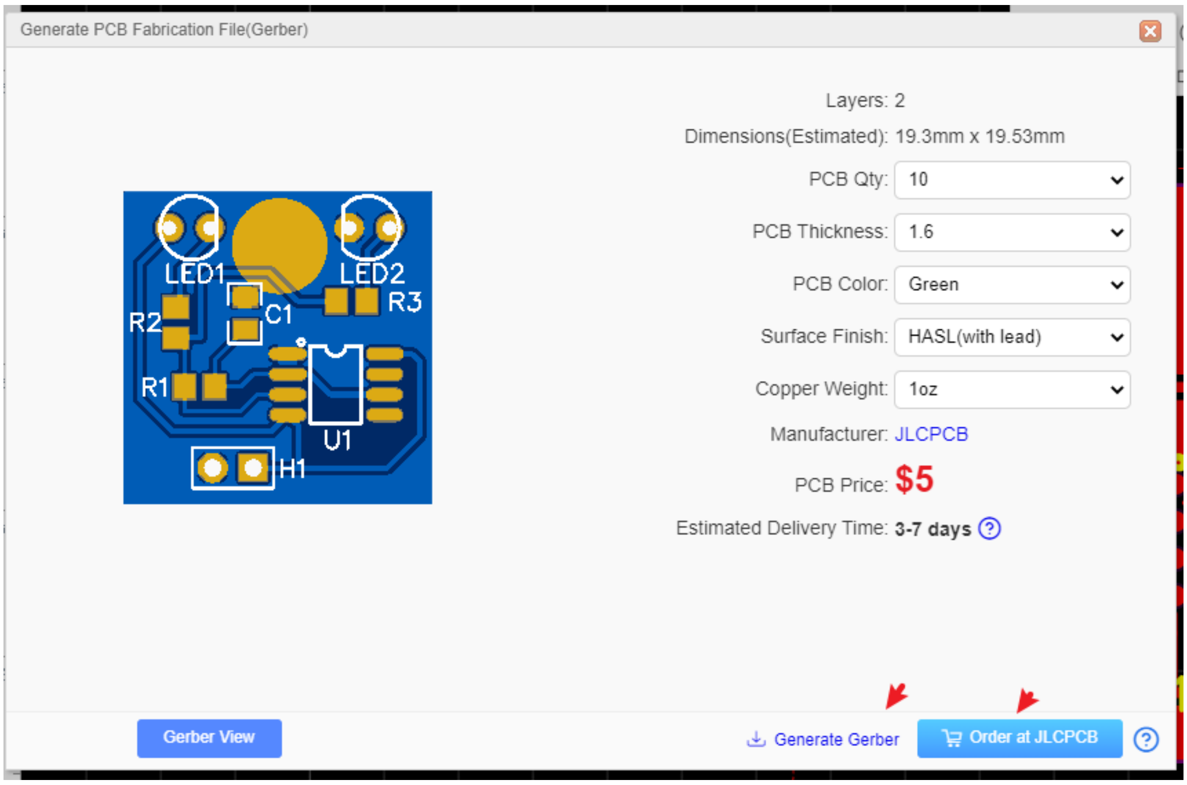
Generate Fabrication File(Gerber)

Generate Fabrication File Gerber

When you finish your PCB, you can output the Fabrication Files(gerber file) via: **File > Generate PCB Fabrication File(Gerber)** , or **Fabrication > PCB Fabrication File(Gerber)**.



After clicking, will open the Gerber generate dialog:


Generate PCB Fabrication File(Gerber)

Layers: 2

Dimensions(Estimated): 19.3mm x 19.53mm

PCB Qty: 10

PCB Thickness: 1.6

PCB Color: Green

Surface Finish: HASL(with lead)

Copper Weight: 1oz

Manufacturer: JLCPCB

PCB Price: $5

Estimated Delivery Time: 3-7 days

You can calculate the price for the PCB order, click SAVE to CART will go to JLCPCB and add your PCB in the cart.

Gerber file name

The generated Gerber file is a compressed zip file. After decompression, you can see the following files:

- **Gerber_BoardOutlineLayer.GKO**: PCB Border file. The PCB board factory cuts the shape of the board according to this document. The groove drawn by the EasyEDA, the solid region(Type: NPTH) is reflected in the border file after the Gerber is generated.
- **Gerber_TopLayer.GTL**: Top side copper layer.
- **Gerber_BottomLayer.GBL**: Bottom side copper layer.
- **Gerber_Inner1.G1**: Inner copper layer, signal type.
- **Gerber_Inner2.GP2**: Inner copper layer, plane type
- **Gerber_TopSilkLayer.GTO**: Top silkscreen.
- **Gerber_BottomSilkLayer.GBO**: Bottom silkscreen.
- **Gerber_TopSolderMaskLayer.GTS**: Top solder mask. The default board is covered with green oil, and the elements drawn on this layer correspond to the top layer's area will not be covered with oil.
- **Gerber_BottomSolderMaskLayer.GBS**: Bottom solder mask. The default board is covered with green oil, and the elements drawn on this layer correspond to the bottom layer's area will not be covered with oil.
- **Drill_PTH_Through.DRL**: Plated drill through hole layer. This document shows the location of the hole where the inner wall needs to be metallized. Old name: Gerber_Drill_PTH.DRL
- **Drill_NPTH_Through.DRL**: Non-Plated drill through hole layer. This document shows the location of the hole where the inner wall don't need to be metallized. Old name: Gerber_Drill_NPTH.DRL
- **Gerber_TopPasteMaskLayer.GTP**: Top Paste Mask, for the stencil.
- **Gerber_BottomPasteMaskLayer.GBP**: Bottom Paste Mask, for the stencil.

- **Gerber_TopAssemblyLayer.GTA**:Top Assembly, read only, doesn't affect the PCB manufacture. Old name: ReadOnly.TopAssembly
- **Gerber_BottomAssemblyLayer.GBA**: Bottom Assembly, read only, doesn't affect the PCB manufacture. Old name: ReadOnly.BottomAssembly
- **Gerber_MechanicalLayer.GML**: Record the information on the mechanical layer in the PCB design, and only use it for information recording. Old name: ReadOnly.Mechanical. By default, the shape of the layer is not manufactured at the time of production. Some board manufacturers use the mechanical layer to make the frame when using Altium file to production. When using Gerber file, it is only used for text identification in JLCPCB. For example: process parameters; V cut path etc. In EasyEDA, this layer does not affect the shape of the border of the board. If the mechanical layer has closed wires, JLCPCB will give priority to using the mechanical layer as the shape of the board when producing the board. If there is no outer frame of the mechanical layer, GKO will be used as the frame (historical influence of Altium file). It is necessary to pay attention to the use of the mechanical layer in the design.

Notice:

- *Before ordering the PCB, please check the gerber at the Gerber view as below.*
- *The Gerber files are generated by browser, please use the browser inner downloader to download!*
- *The coordinates of the Gerber file follow the canvas coordinates*
- *When exporting Gerber, the coordinate format accuracy defaults to 3:3. When the PCB size is out of range, it automatically uses 4:2 format. If you view the Gerber as such as CAM350, found that the Drill hole has been offset the location, you can modify the drill coordinate format to fit the location*

Gerber View

Before sending Gerber to the factory, please use gerber viewer to check the Gerber carefully.

local gerber viewer you can use such as: Gerbv, FlatCAM, CAM350, ViewMate, GerberLogix etc.

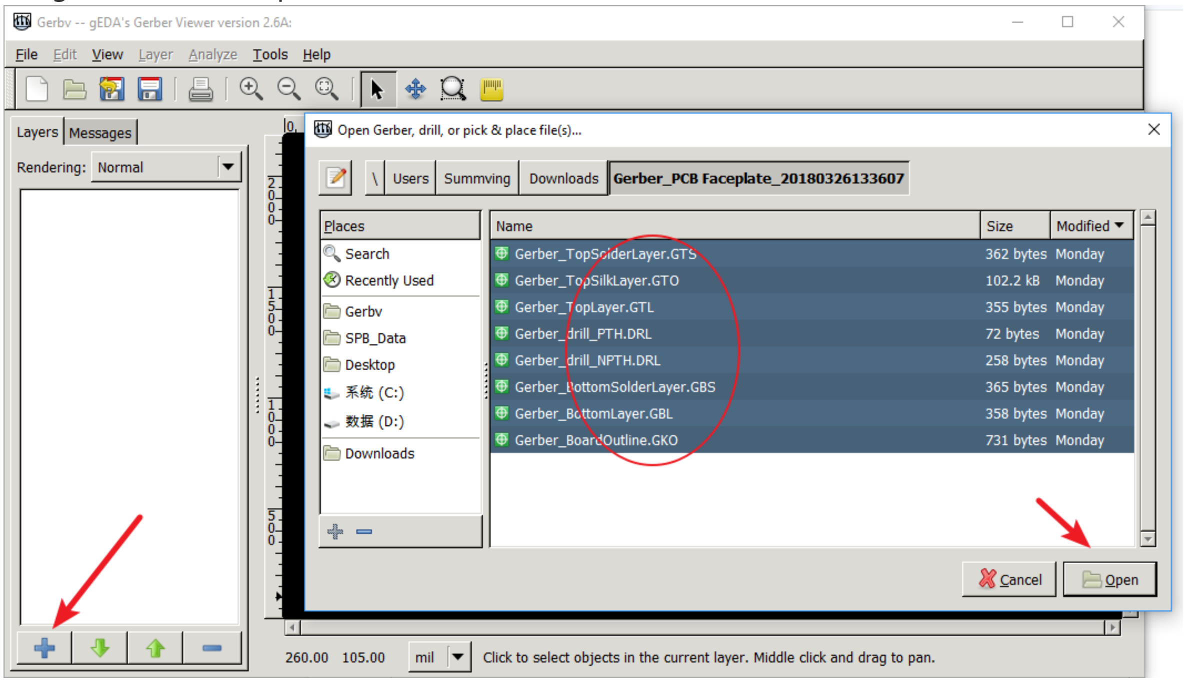
Gerber viewer recommend Gerbv:

- Project page:http://gerbv.geda-project.org/
- Download: https://sourceforge.net/projects/gerbv/files/

How to use Gerbv:

1.Download Gerber zip file, and download Gerbv, unzip Gerber file and run the Gerbv;

2.Click the + button at the Gerbv dialog bottom-left corner, open the gerber folder, select all the gerber files, and open.


Gerbv -- gEDA's Gerber Viewer version 2.6A:
File Edit View Layer Analyze Tools Help
Layers Messages
Rendering: Normal
Open Gerber, drill, or pick & place file(s)...
\ Users Summving Downloads Gerber_PCB Faceplate_20180326133607
Places
Search
Recently Used
Gerbv
SPB_Data
Desktop
系統 (C:)
数据 (D:)
Downloads
Cancel
Open
260.00 105.00 mil
Click to select objects in the current layer. Middle click and drag to pan.

Name	Size	Modified
Gerber_TopSolderLayer.GTS	362 bytes	Monday
Gerber_TopSilkLayer.GTO	102.2 kB	Monday
Gerber_TopLayer.GTL	355 bytes	Monday
Gerber_drill_PTH.DRL	72 bytes	Monday
Gerber_drill_NPTH.DRL	258 bytes	Monday
Gerber_BottomSolderLayer.GBS	365 bytes	Monday
Gerber_BottomLayer.GBL	358 bytes	Monday
Gerber_BoardOutline.GKO	731 bytes	Monday

3.And then zoom, measure, check every layer, check drill holes and location. etc.

FlatCAM is a nice tool too: http://flatcam.org/

FlatCAM lets you take your designs to a CNC router. You can open Gerber, Excellon or G-code, edit it or create from scatch, and output G-Code. Isolation routing is one of many tasks that FlatCAM is perfect for. It's is open source, written in Python and runs smoothly on most platforms.

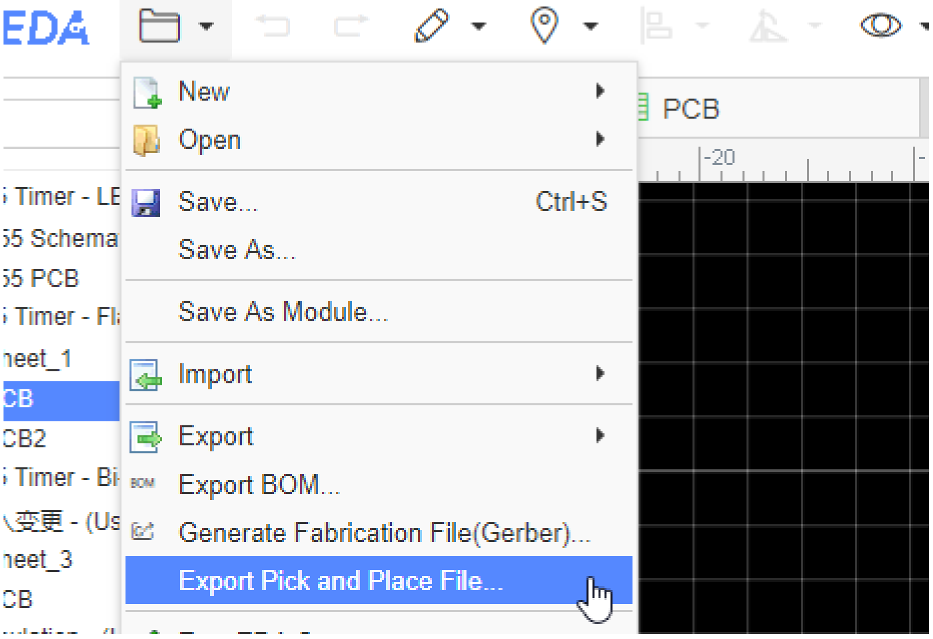
Free Online Gerber Viewer:

Recommend：
jlcpcb.com
tracespace.io/view
gerber.ucamco.com

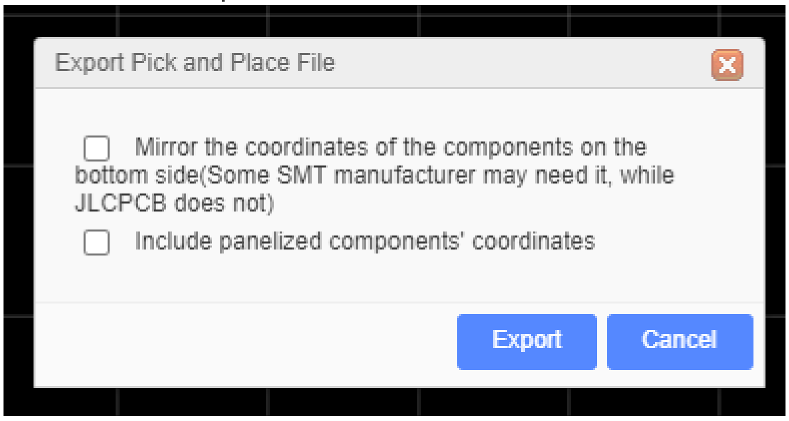
Export Pick and Place File

In PCB editor, if you want to generate Pick And Place as a CSV file, you can via:

File > **Export Pick and Place File** or **Top Menu - Fabrication - Pick and Place File**.



You can set the options:



Export Pick and Place File

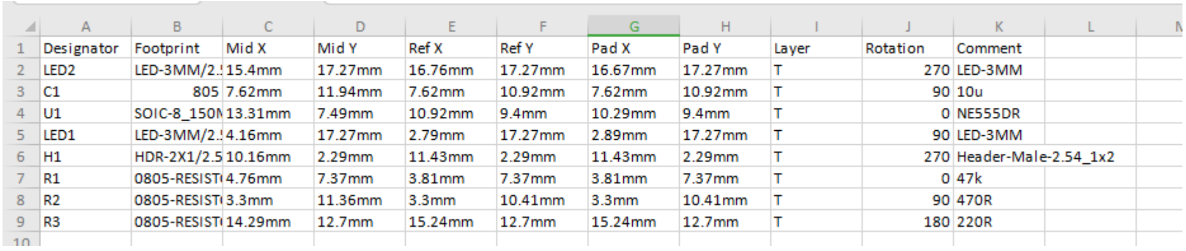
☐ Mirror the coordinates of the components on the bottom side(Some SMT manufacturer may need it, while JLCPCB does not)

☐ Include panelized components' coordinates

Export Cancel

If your PCB has been panelize by the editor, you can enable the "Include panelized components coordinate".

When you open the exported CSV file, you can see:



	A	B	C	D	E	F	G	H	I	J	K	L
1	Designator	Footprint	Mid X	Mid Y	Ref X	Ref Y	Pad X	Pad Y	Layer	Rotation	Comment	
2	LED2	LED-3MM/2.!	15.4mm	17.27mm	16.76mm	17.27mm	16.67mm	17.27mm	T	270	LED-3MM	
3	C1	805	7.62mm	11.94mm	7.62mm	10.92mm	7.62mm	10.92mm	T	90	10u	
4	U1	SOIC-8_150M	13.31mm	7.49mm	10.92mm	9.4mm	10.29mm	9.4mm	T	0	NE555DR	
5	LED1	LED-3MM/2.!	4.16mm	17.27mm	2.79mm	17.27mm	2.89mm	17.27mm	T	90	LED-3MM	
6	H1	HDR-2X1/2.5	10.16mm	2.29mm	11.43mm	2.29mm	11.43mm	2.29mm	T	270	Header-Male-2.54_1x2	
7	R1	0805-RESIST	4.76mm	7.37mm	3.81mm	7.37mm	3.81mm	7.37mm	T	0	47k	
8	R2	0805-RESIST	3.3mm	11.36mm	3.3mm	10.41mm	3.3mm	10.41mm	T	90	470R	
9	R3	0805-RESIST	14.29mm	12.7mm	15.24mm	12.7mm	15.24mm	12.7mm	T	180	220R	

This file support two units "mm" and "mil", it is following the PCB unit setting.

There is an option "Mirror the coordinates of the components on the bottom side(Some SMT manufacturer may need it, while JLCPCB does not)", you can check with your SMT manufacturer, the mostly SMT manufacturer doesn't need it.

Notice:

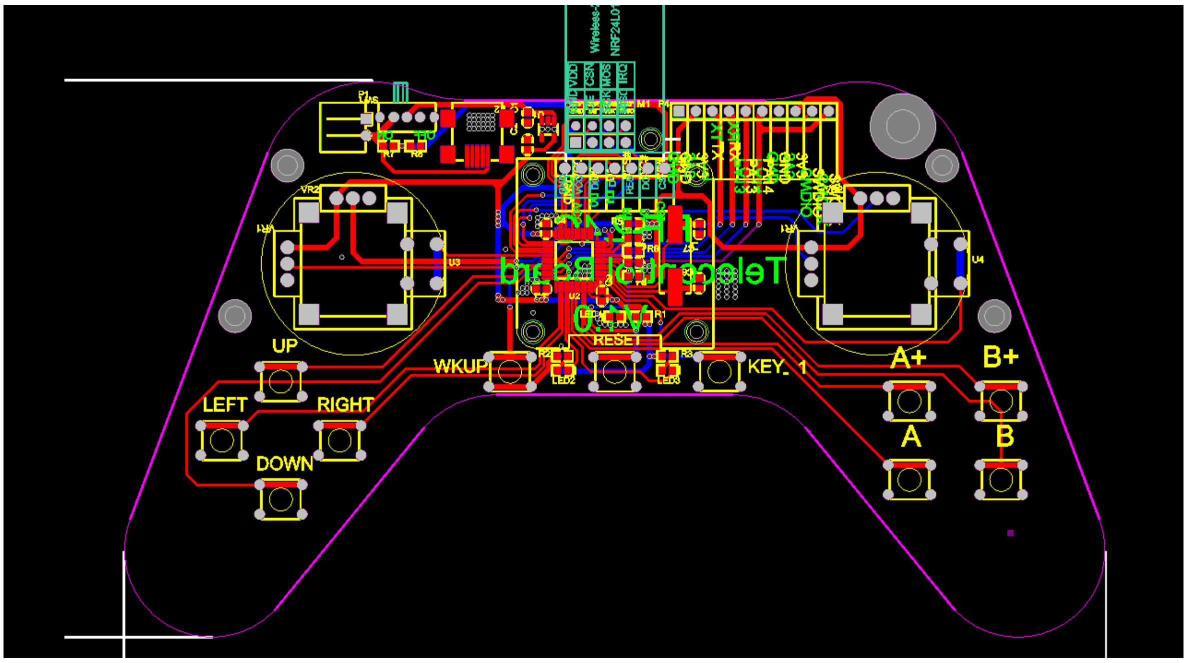
- In order to support multiple languages, BOM and Pick and Place files (CSV file) are UNICODE encoded and tab-based. If the CSV file cannot be read by your components vendor or PCB manufacturer, please convert the encoding and change the delimiter.
- Recommended solution: Save as a new CSV file in Excel or WPS. For example, open a CSV file in Excel, click or select: Save As - Other Formats - CSV (Comma Separated) (*. csv).

You can also open the CSV file with any text editor (such as Windows Notepad) and save as ANSI or UTF-8 encoding. If necessary, replace all tabs with commas.

Export DXF

EasyEDA support to export PCB to DXF.

At present EasyEDA supports to export a full layers and objects DXF file:



You can edit it at CAD tools very easy, toggle the layers as you want.

Export Altium Designer Format

EasyEDA supports exporting the schematics and PCB in Altium Designer format.

The "export to Altium" function is now in beta, Please check carefully after exporting the design to Altium, EasyEDA cannot guarantee that there are no errors!!! EasyEDA is not responsible for any loss due to library errors or format conversion!!! If you do not agree, please do not use Altium export!!!

If you want to order a PCB please generate a Gerber file instead of exporting to Altium! Please do not export your design to Altium and import it more than once as this may cause some details to be lost!!!
Altium version 19 is not supported yet, please open exported file with Altium version 18 or before, version 17 is recommended

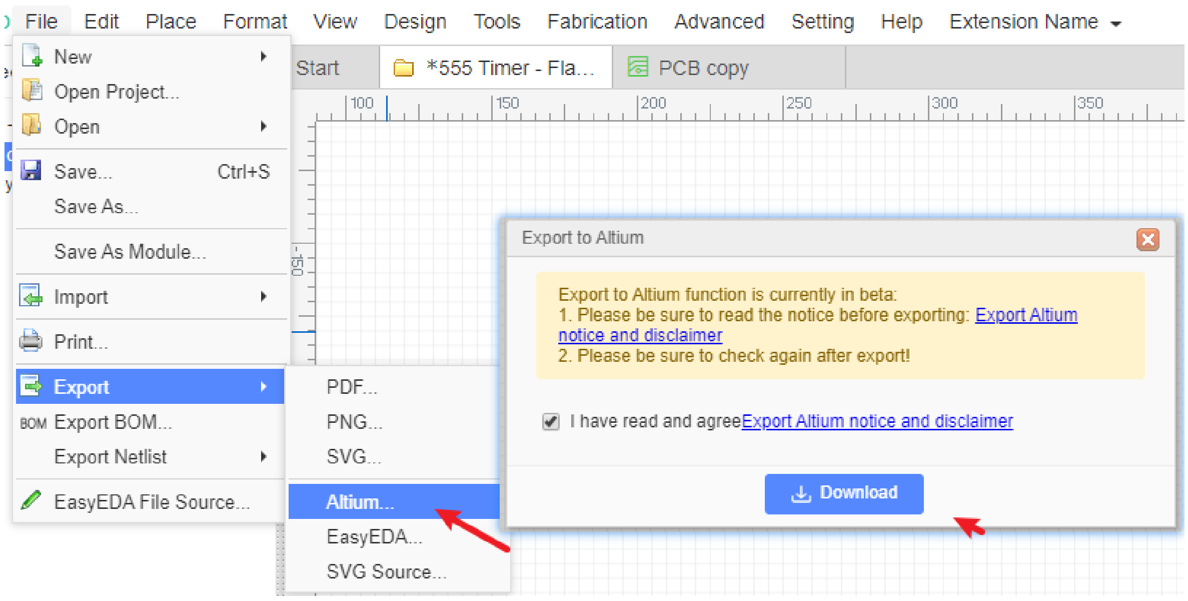
If you find any errors in the export details, please contact us so we can fix it, to help us please include the problem details and files.

support@easyeda.com

When exporting, you must log in first before exporting.

Exporting Schematics In Altium Designer Format

EasyEDA supports exporting the schematics in Altium Designer format.
Use **"File > Export > Altium..."**, clicking **"Download"** will create a `.schdoc` file.

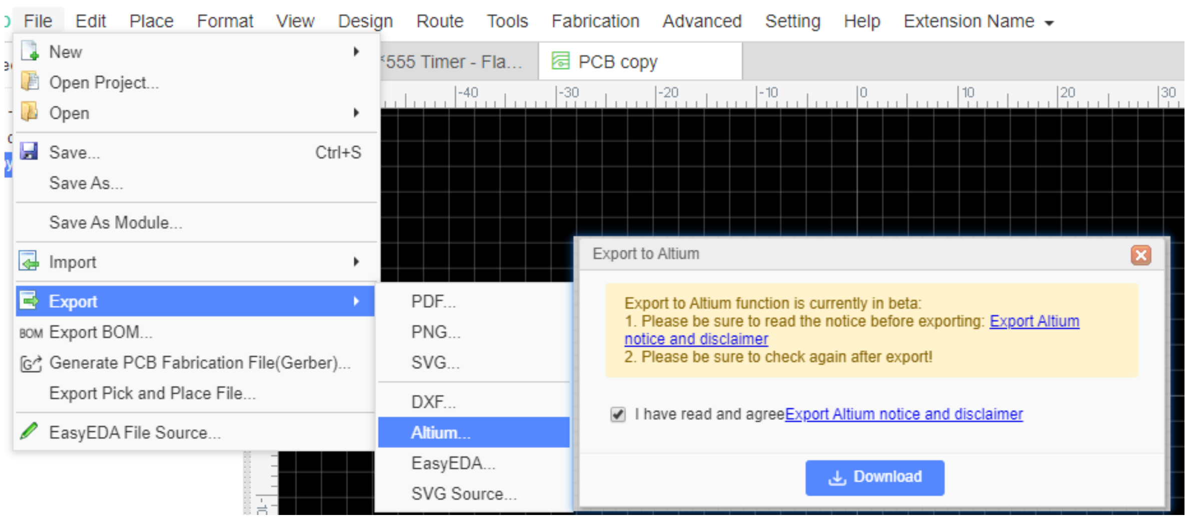


Export to Altium function is currently in beta:
1. Please be sure to read the notice before exporting: Export Altium notice and disclaimer
2. Please be sure to check again after export!

☑ I have read and agreeExport Altium notice and disclaimer

Exporting PCB in Altium Designer Format

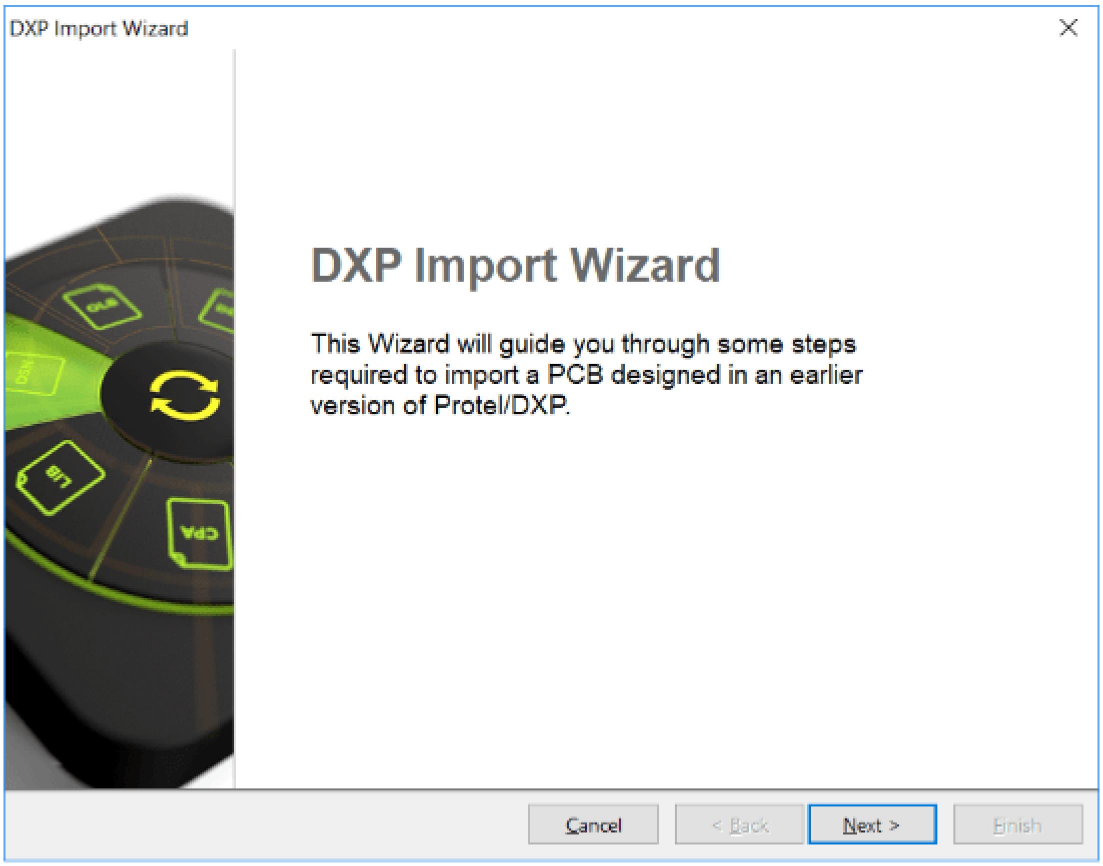
EasyEDA supports exporting the PCB in Altium Designer format.
Use **"File > Export > Altium..."** to create a `.pcbdoc` file.



Export to Altium function is currently in beta:
1. Please be sure to read the notice before exporting: Export Altium notice and disclaimer
2. Please be sure to check again after export!

☑ I have read and agreeExport Altium notice and disclaimer

When opening the exported PCB file in Altium Designer there will be a dialog "DXP Import Wizard", don't worry, just cancel it to continue.

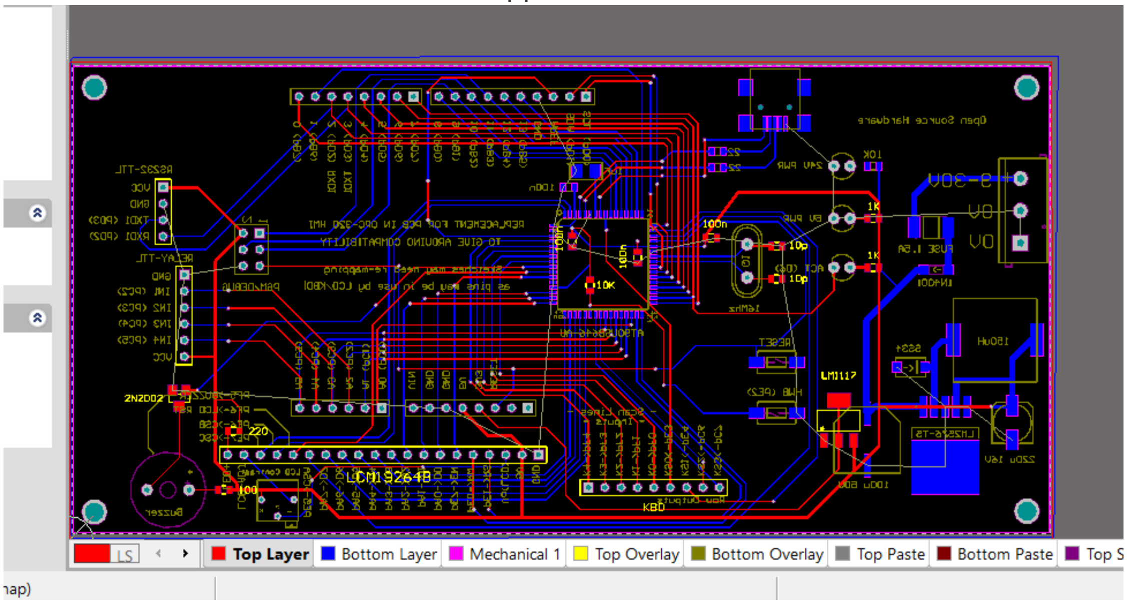

DXP Import Wizard

DXP Import Wizard

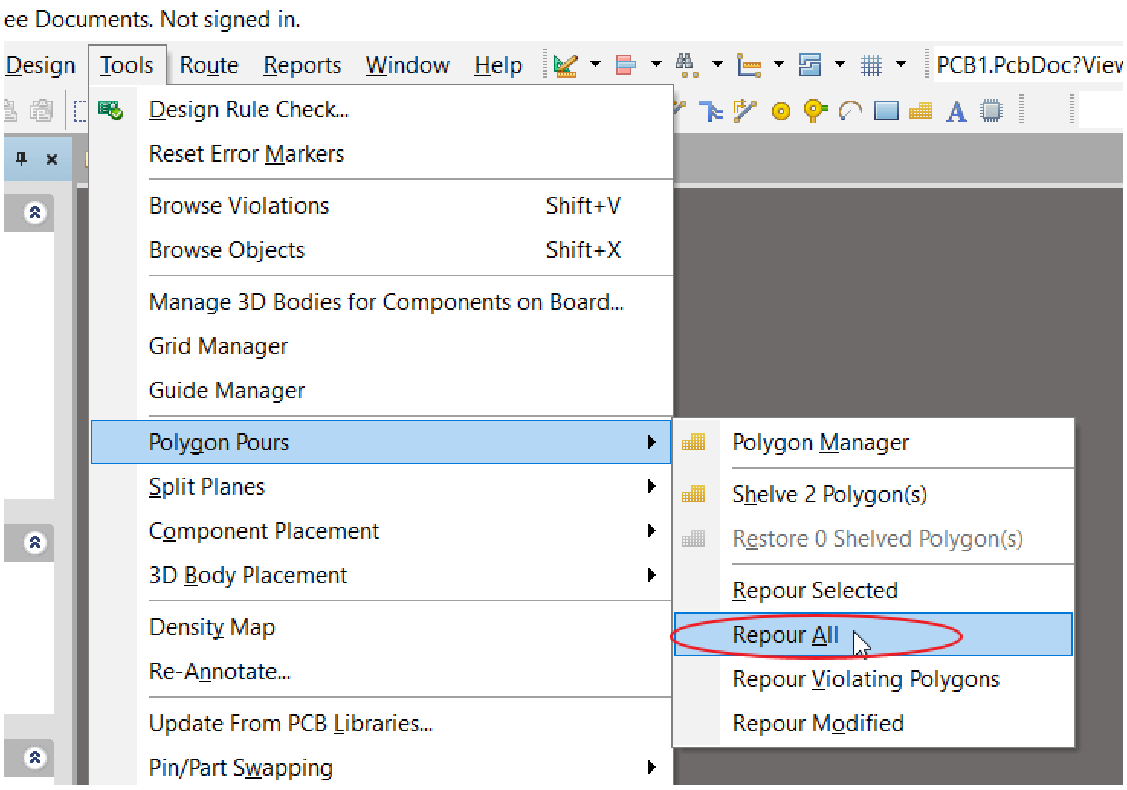
This Wizard will guide you through some steps required to import a PCB designed in an earlier version of Protel/DXP.

Known Issues:

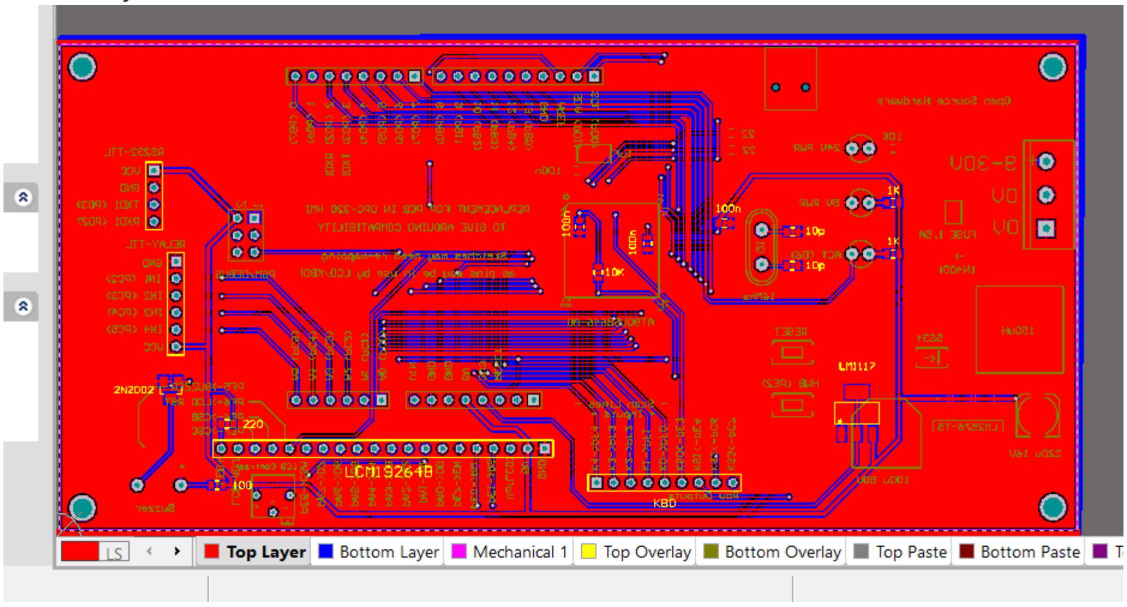
- **1. No Copper Area fill data.**
  The PCB file will look like this without copper area:



You need to repour all polygons in Altium Designer using: **Tools > Polygon Pours > Repour All**:

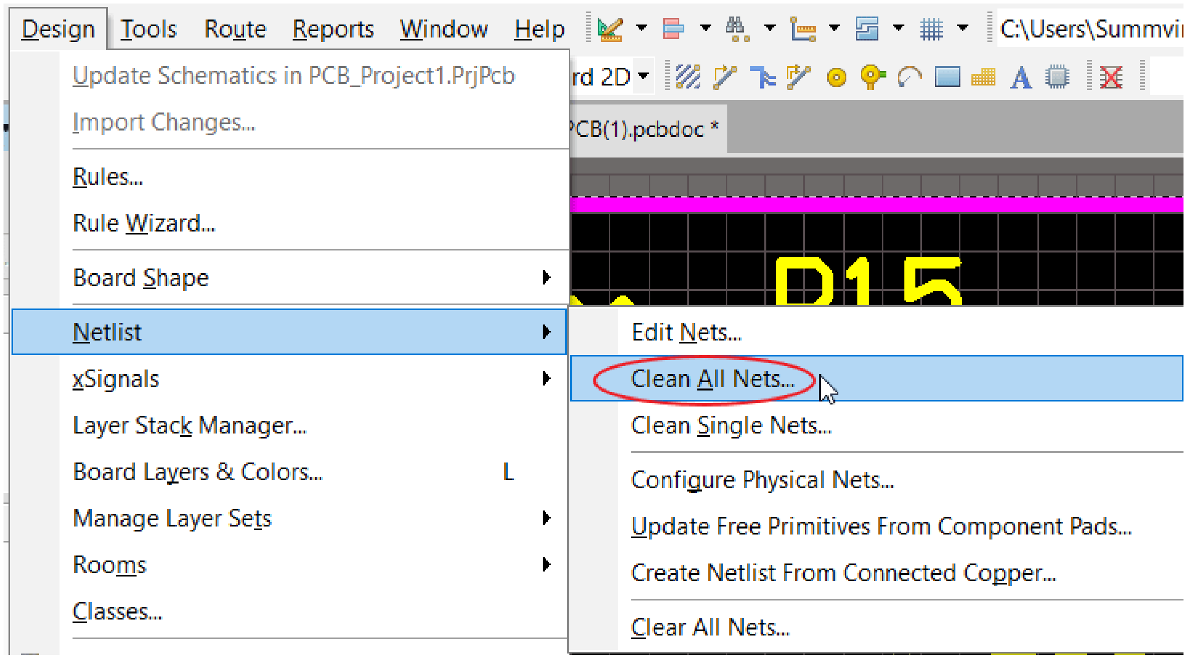


And finally, save it.

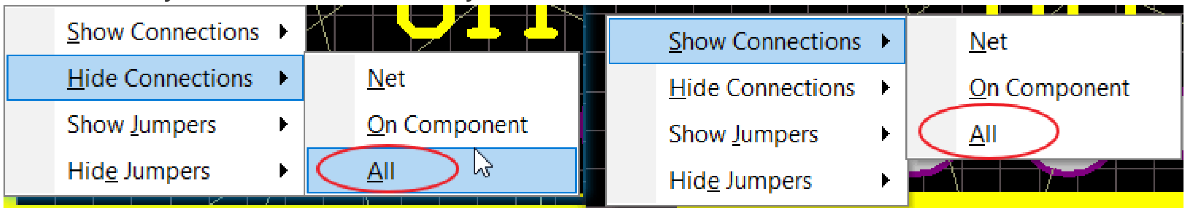


- **2. No Ratlines.**

If you export the PCB without ratlines, you need to show all connections first before routing :
Use: **Design** > **Netlist** > **Clean All Nets** (**D** > **N** > **A**),and then (**V** > **C** > **S**)



Or use hotkey: **N** > **H** > **A** followed by **N** > **S** > **A**:



- **3. Inner layer Plane Zone doesn't export perfectly.**

You need to rebuild the plane zone and re-assign plane zone's net.

- **4. Doesn't support DRC rule.**

Please check the DRC manually.

- **5. The text may be changed.**

Some text may change position depending on the chosen font. If the text does not display correctly just manually adjust the position.

Exporting Footprint and Symbol in Altium Designer Format

EasyEDA does not support exporting the Symbol or Footprint in Altium Designer library format, but you can place the libraries with the schematic or PCB and export that in Altium Designer format, and then extract the libraries from Altium Designer.

EasyEDA is not responsible for any loss due to library errors or format conversion!!! If you do not agree to this, please do not use Altium export!!!

Export SVG Source

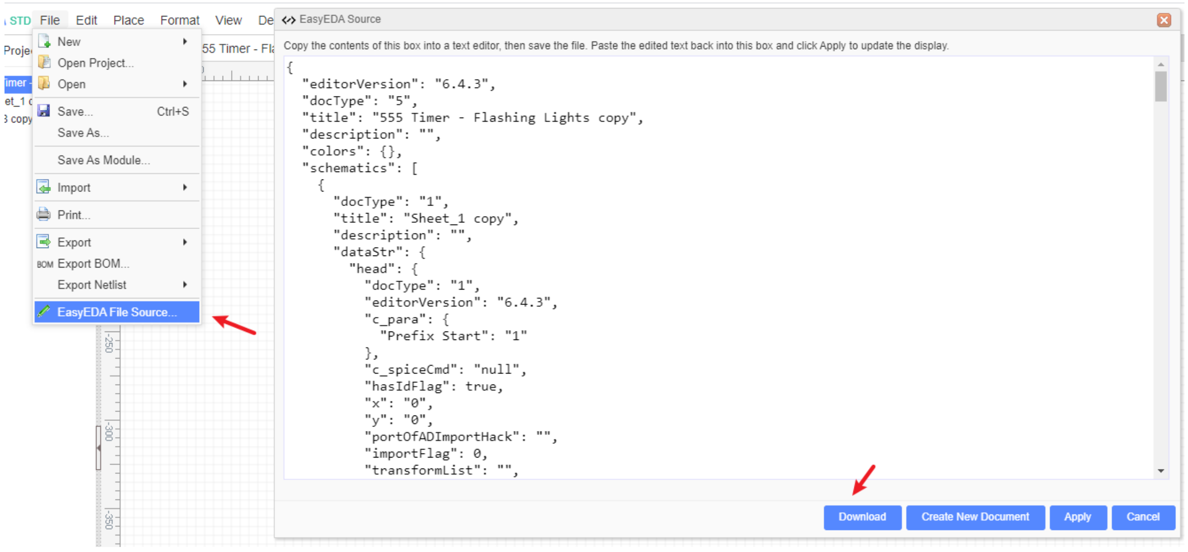
EasyEDA supports exporting or editing SVG source.

You can create an SVG source file using:

File > Export > SVG source...

First copy the contents of this box into a text editor and then save the file with the .svg extension. You can edit it in Inkscape or open it in your browser.

This solution doesn't need an Internet connection, you can use it off-line with EasyEDA.



Copy the contents of this box into a text editor, then save the file. Paste the edited text back into this box and click Apply to update the display.

```
{
  "editorVersion": "6.4.3",
  "docType": "5",
  "title": "555 Timer - Flashing Lights copy",
  "description": "",
  "colors": {},
  "schematics": [
    {
      "docType": "1",
      "title": "Sheet_1 copy",
      "description": "",
      "dataStr": {
        "head": {
          "docType": "1",
          "editorVersion": "6.4.3",
          "c_para": {
            "Prefix Start": "1"
          },
          "c_spiceCmd": "null",
          "hasIdFlag": true,
          "x": "0",
          "y": "0",
          "portOfADImportHack": "",
          "importFlag": 0,
          "transformList": "",
```

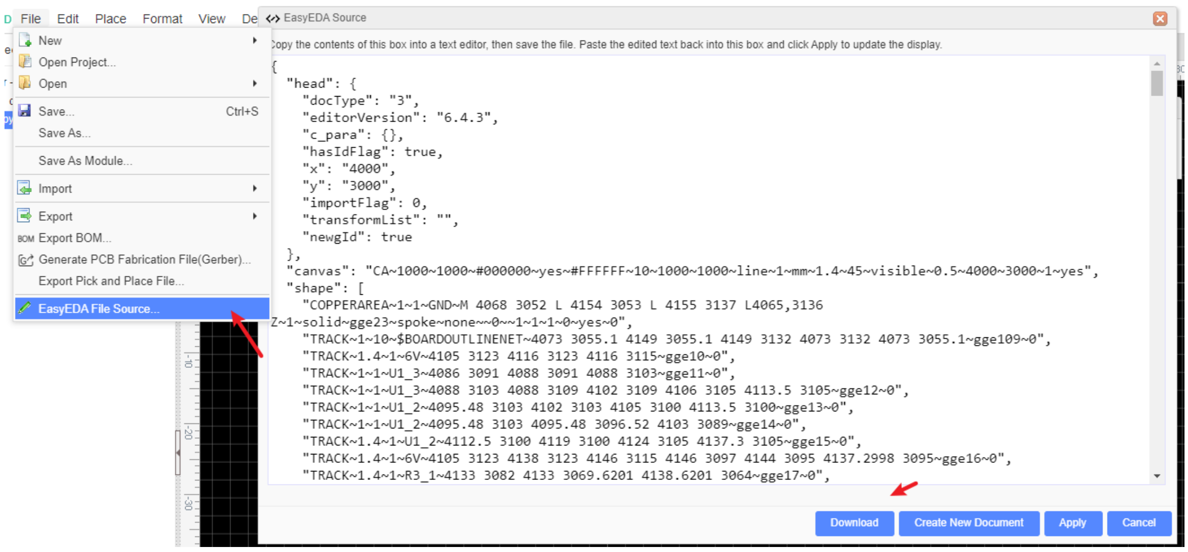
Export EasyEDA Source

EasyEDA supports saving your file to local storage, you can download your design as an EasyEDA source file.

1. Export EasyEDA document directly

You can create an EasyEDA source file with:

** > File > EasyEDA File Source...**

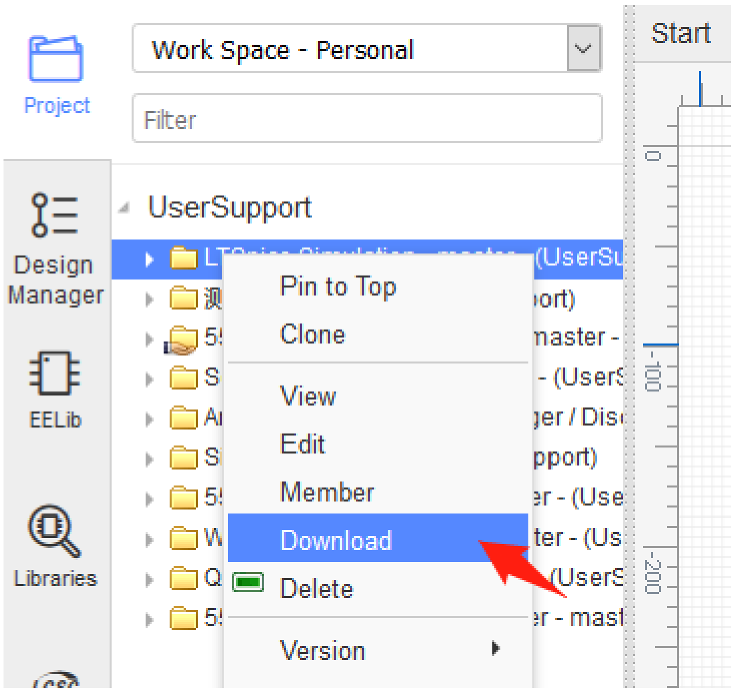


Copy the contents of this box into a text editor, then save the file. Paste the edited text back into this box and click Apply to update the display.

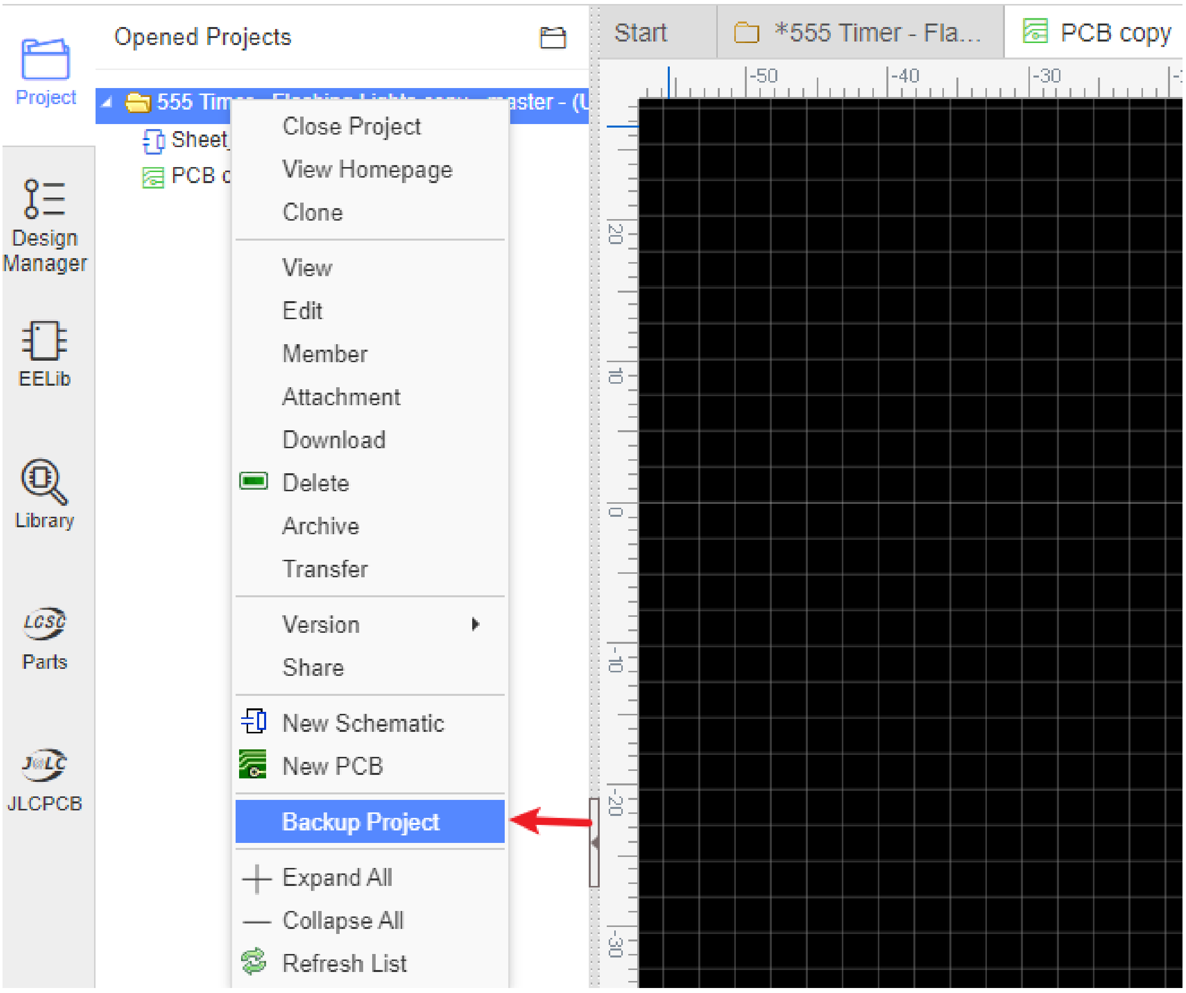
```
{
  "head": {
    "docType": "3",
    "editorVersion": "6.4.3",
    "c_para": {},
    "hasIdFlag": true,
    "x": "4000",
    "y": "3000",
    "importFlag": 0,
    "transformList": "",
    "newgId": true
  },
  "canvas": "CA~1000~1000~#000000~yes~#FFFFFF~10~1000~1000~line~1~mm~1.4~45~visible~0.5~4000~3000~1~yes",
  "shape": [
    "COPPERAREA~1~1~GND~M 4068 3052 L 4154 3053 L 4155 3137 L4065,3136
Z~1~solid~gge23~spoke~none~~0~~1~1~1~0~yes~0",
    "TRACK~1~10~$BOARDOUTLINENET~4073 3055.1 4149 3055.1 4149 3132 4073 3132 4073 3055.1~gge109~0",
    "TRACK~1.4~1~6V~4105 3123 4116 3123 4116 3115~gge10~0",
    "TRACK~1~1~U1_3~4086 3091 4088 3091 4088 3103~gge11~0",
    "TRACK~1~1~U1_3~4088 3103 4088 3109 4102 3109 4106 3105 4113.5 3105~gge12~0",
    "TRACK~1~1~U1_2~4095.48 3103 4102 3103 4105 3100 4113.5 3100~gge13~0",
    "TRACK~1~1~U1_2~4095.48 3103 4095.48 3096.52 4103 3089~gge14~0",
    "TRACK~1.4~1~U1_2~4112.5 3100 4119 3100 4124 3105 4137.3 3105~gge15~0",
    "TRACK~1.4~1~6V~4105 3123 4138 3123 4146 3115 4146 3097 4144 3095 4137.2998 3095~gge16~0",
    "TRACK~1.4~1~R3_1~4133 3082 4133 3069.6201 4138.6201 3064~gge17~0",
```

2. Download the project

Select: **Project folder > Right Click > Download**, this will download a zip file with EasyEDA Source files containing Schematics and PCBs.



You can also backup the project using: **Project folder > Right Click > Backup Project**



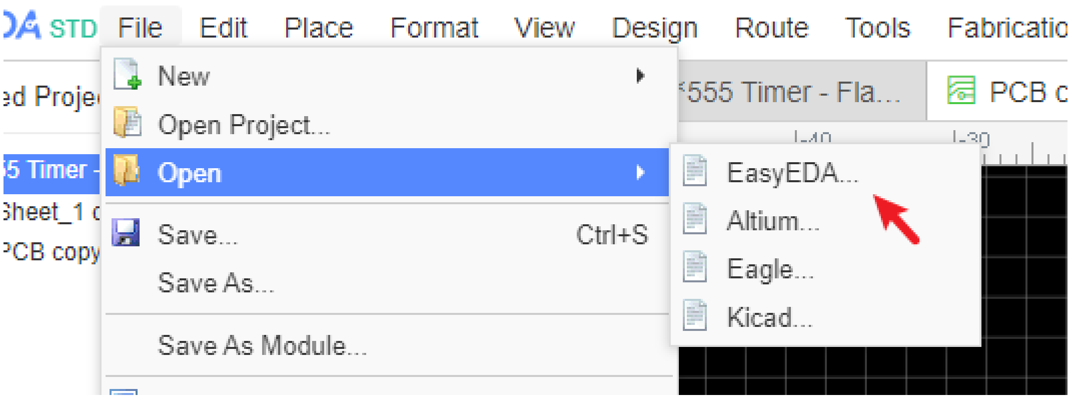
This will open a dialog and you can select the projects what you want to backup. Backup projects may be done once per day.

EasyEDA Source File is a **JSON** file which can be read by many other programs. JSON format is described here:

http://en.wikipedia.org/wiki/JSON

3. Open EasyEDA File

If you want to open the EasyEDA file you exported use: " - File - Open - EasyEDA...".



After editing you can save the document again.

Tutorial Video

Please refer at:

EasyEDA: EasyEDA Tutorials 2020

wegi: EASYEDA TUTORIAL

Engg. Softwares Technology Innovation: EASYEDA PCB DESIGN SIMULATION SOFTWARE TUTORIAL EDUCATIONAL PURPOSE

Arafa Microsys: Arabic EasyEDA Tutorials

Electronoobs : EasyEDA Full TUTORIAL + Create Component + TIPS

Easy HomeMade Projects: How to Design a PCB easily with EasyEDA & JLCPCB - Complete Tutorial

Delali: How to Design a PCB from Scratch with EasyEDA | COMPLETE Tutorial | JLCPCB LCSC ALIEXPRESS

PCB Design Experts: PCB Design for everyone with EasyEDA a free and online tool

Great Sprout: EasyEDA PCB design full tutorial for Beginners 2020

T Tech Mentor: Easy EDA Complete Course Playlist in Urdu\Hindi Easy EDA Class 00\How to make DIY pcb

Ricardo Teixeira: EasyEDA Esquemático e PCB Simples

ELECTRONOOBS España: Tutorial EasyEDA Completo + Crear Componente | Consejos

Design n Techie: EasyEda Tutorial - Absolute Beginners

Simulation Tutorial

Simulation eBook - Headings

Simulation eBook - Headings

Simulation eBook - Preliminary remarks

Preliminary remarks

Please note that before launching into simulations, we strongly recommend you read - and play with the examples in - the whole of this Simulation Tutorial.

This section covers some basic points about how to use EasyEDA for simulation.

Please note that the simulation tool in EasyEDA:

- **is primarily for analogue circuit simulation.**
- **is not designed for use in simulating systems containing microprocessors, microcontrollers, DSPs and FPGAs.**
- **does not support any form of code development tools.**

Whilst there are models for a few small to medium scale integration logic devices available in the Spice Simulation library, it is generally not practical to model the following types of devices in EasyEDA:

- Analogue to Digital Converters (ADC);
- Digital to Analogue Converters (DAC);

- Devices that use Serial Interfaces such as I2C or SPI for communications and configuration;
- Microprocessors,and microcontrollers including Arduino or similar types of modules;
- DSPs;
- FPGA;
- Complex State Machines;
- Look Up Tables or ROM;
- RAM;
- Large and Very Large Scale Integrated devices and processing functions;
- IBIS models (unless converted into Spice models by some 3rd party tool).

There are three basic reasons that such devices are impractical to model.

They generally require so many clock cycles to run through their internal states that the simulation times required quickly stretch into hours;
Modelling such devices at transistor or even gate level places massive demands on CPU and storage resources;
Spice is not designed to easily represent the complex states and processes of code driven devices.

In most cases, simulating systems containing microprocessors, microcontrollers and DSPs is best carried out in a dedicated development environment.

In many cases however the objective is to simulate the analogue circuitry around - and their interaction with - the IO ports of such devices. If the functions and interfaces of such devices can be simplified and modelled as behavioural blocks made for example from voltage sources configured as pulse sources, switches and some fairly basic logic devices such as gates, counters and shift registers, then in many cases the circuitry interfacing to such devices can still be simulated using the EasyEDA simulation tool.

How to find simulatable parts in EasyEDA

How to find Spice Symbols (which have models already associated with them)

All schematics in EasyEDA are created using symbols.

There are two types of schematics in EasyEDA:
A passive schematic representing a circuit that is intended to be converted into a PCB;
An active schematic representing a circuit that is intended to be simulated.

There are two types of symbols used in schematic creation;
Schematic Symbols. These are used to create a passive schematic;
Spice Symbols. These are used to create an active (simulatable) schematic.

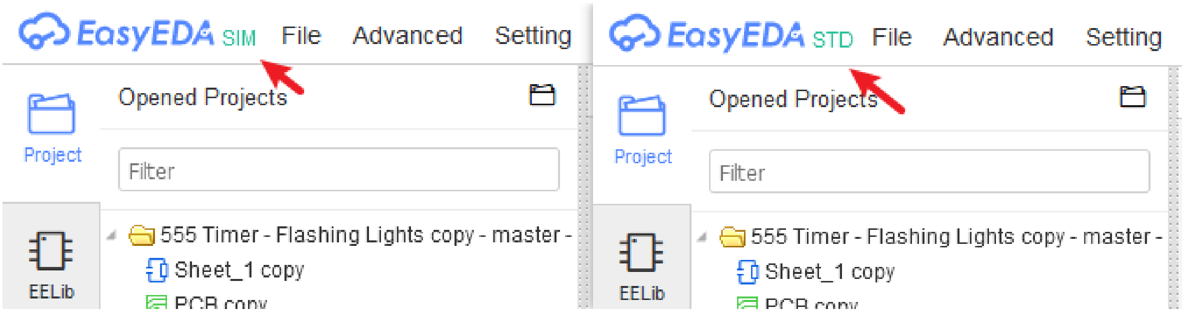
The major difference between them is that **Spice Symbols** have spice models associated with them whereas **Schematic Symbols** do not.

Schematic Symbols cannot be simulated because they do not have spice models associated with them.

Therefore schematics created using **Schematic Symbols** cannot be simulated.

Schematics can **only** be simulated if they are created using **Spice Symbols.**

From V6.x.the Schematic Editor has to be switched from **Std** to **Sim** mode to create and run simulations.

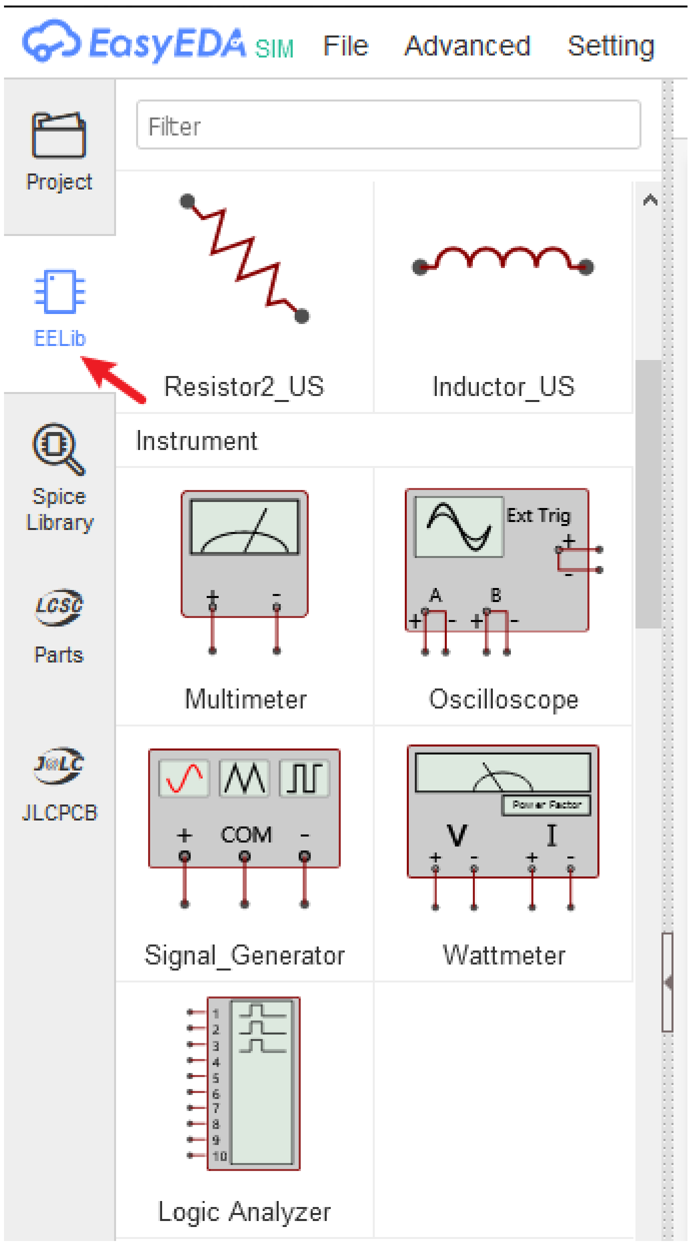


In **Std** mode:

- simulation is not supported.
- the Schematic Symbols available from the **EELib** and most of the symbols available from the SHIFT+F library Search Libraries tool **cannot be used in simulations because they do not have spice models associated with them**.

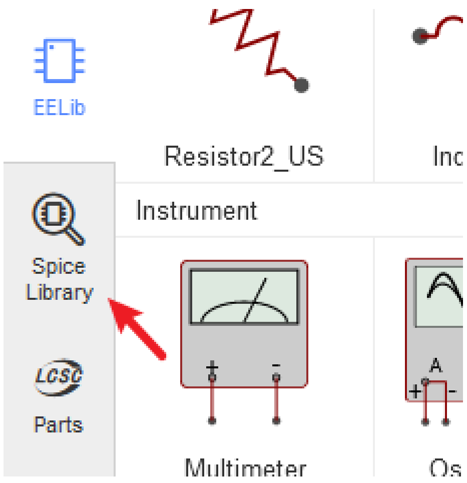
In **Sim** mode, at present:

- the **EELib** button only presents a very small number of the available Spice Symbols that have spice models associated with them;



- The range of Spice Symbols with spice simulation models attached to them can now be found using the:

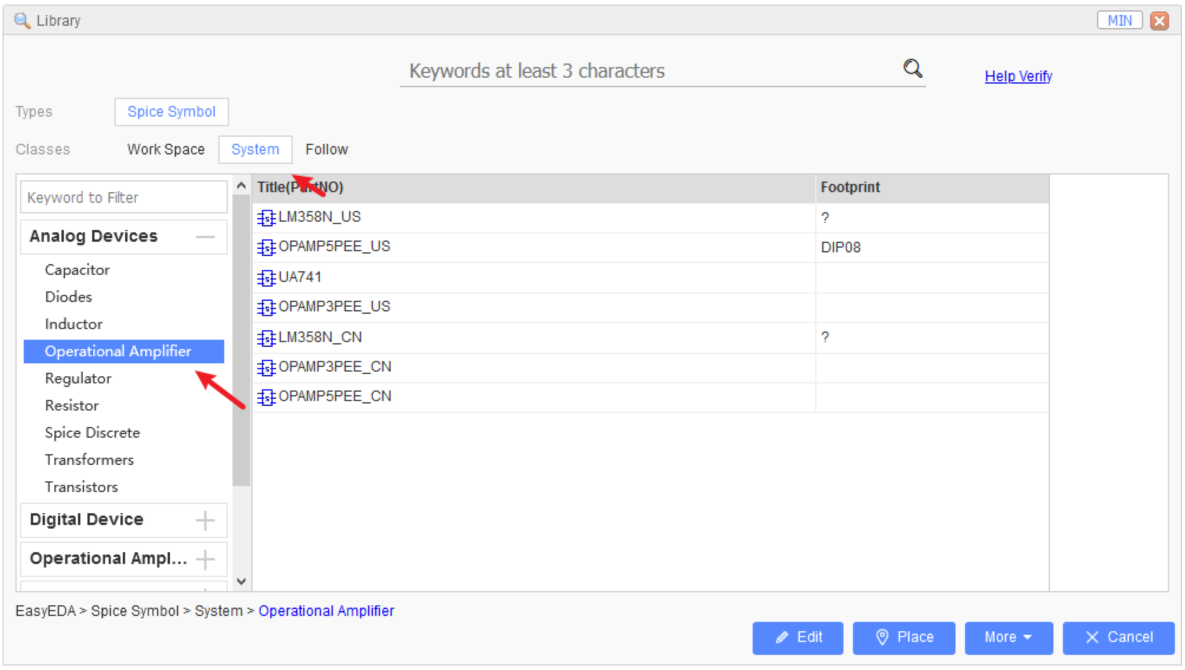
Spice Library button in the left-hand panel:



or;

SHIFT+F in any editor window.

then click on the **System** button then click on one of the choices in the left hand selection:



Types: Spice Symbol

Classes: Work Space | System | Follow

Keyword to Filter

Analog Devices
- Capacitor
- Diodes
- Inductor
- Operational Amplifier
- Regulator
- Resistor
- Spice Discrete
- Transformers
- Transistors

Digital Device

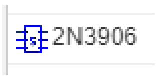
Operational Ampl...

Title(PartNO)	Footprint
LM358N_US	?
OPAMP5PEE_US	DIP08
UA741	
OPAMP3PEE_US	
LM358N_CN	?
OPAMP3PEE_CN	
OPAMP5PEE_CN	

EasyEDA > Spice Symbol > System > Operational Amplifier

In the Search Libraries tool, Spice Symbols are distinguished from Schematic Symbols by having a darker blue icon with a small **s** inside it:

A Spice Symbol for a simulatable 2N3906 transistor:



This symbol is simulatable.

A Schematic Symbol for a non-simulatable 2N3906 transistor:

2N3906_

This symbol is **not** simulatable.

How to find spice models and subcircuits (which may or may not already have a symbol associated with them)

You will find a list of all the spice models (.model) and spice subcircuits (.subckt) currently available for EasyEDA here:

To be updated

Some of the models in this list are already used by the Spice Symbols referred to above. Some of these models have no dedicated symbols but they can be associated with existing symbols just by editing the symbol names.

Alternate models can be assigned to the Spice Simulation symbols by changing the names of the symbols to that of the new model and also - depending on whether the new model is a .model or a .subckt defined model - by pressing the I key and changing the spice prefix.

You can enter these model names into any appropriate Spice Simulation symbol and then set the spice prefix to the prefix appropriate for the model (See: LTspice model types in: .model statements)

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

What to do for a part that has no model available in EasyEDA but for which a 3rd party model is available

It is also possible to import spice models from 3rd party sources to use with the Spice Simulation symbols from these list.

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

What to do if there is a 3rd party model available but for which there is no suitable symbol in EasyEDA

In the case of there being a 3rd party model available but for which there is no suitable symbol in EasyEDA then it is quite straightforward to create a spice symbol for it within EasyEDA either by editing an existing symbol or by creating a new one from scratch.

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

Note that not all 3rd party models are compatible with LTspice syntax. Spice3 and Pspice versions of models should run out of the box. Models for other versions of spice may require modifications to make them work in EasyEDA.

For help on this, please post to:

Spice Simulation

in the forums.

What to do if there is no model available for a part

Not all components have a publically available model. In some instances, a component may not have a model at all.

However, we have created models - unique to EasyEDA - for a number of parts, such as the LDR_EE, ELECTRET_MIC_EE, LM56EE and LM2907EE. So, although it is not always possible or practical to build a model, if there is a part for which you particularly need one then please contact Support to discuss your requirements.

How to run a simulation

Please see:

How to run a simulation

and for more information:

Advanced probing and simulation control

Simulation eBook - Introduction

Preliminary remarks

This section covers some basic points about how to use EasyEDA for simulation.

Please note that the simulation tool in EasyEDA is primarily for analogue circuit simulation.

Whilst there are models for a few small to medium scale integration logic devices available in the Spice Simulation library, it is generally not practical to model the following types of devices in EasyEDA:

- Analogue to Digital Converters (ADC);
- Digital to Analogue Converters (DAC);
- Devices that use Serial Interfaces such as I2C or SPI for communications and configuration;
- Microprocessors and microcontrollers including Arduino or similar types of modules;
- FPGA;
- State Machines;
- Look Up Tables or ROM;
- RAM;

- Large and Very Large Scale Integrated devices and processing functions;
- IBIS models (unless converted into Spice models by some 3rd party tool).

EasyEDA also does not support any form of code development tools.

Please note that before launching into simulations, we strongly recommend you read - and play with the examples in - the whole of this Simulation Tutorial.

The EasyEDA Simulation eBook

How to find simulatable parts in EasyEDA

How to find spice symbols (which have models already associated with them)

From V4.1.3. - except for the simple passive R, L, C and the voltage and current source devices - most of the schematic symbols for the active devices shown in the left hand panel under the `EELib` button no longer have spice models associated with them.

Schematic symbols with spice simulation models attached to them can now be found using the:

Libraries button in the left-hand panel

or;

SHIFT+F in any editor window.

then click on `Symbols`, hover the mouse over the `System` class then slide down and click on any of the grey buttons under the **Spice Simulation** heading:

How to find spice models and subcircuits (which may or may not already have a symbol associated with them)

You will find a list of all the spice models (.model) and spice subcircuits (.subckt) currently available for EasyEDA here:

https://docs.google.com/spreadsheets/d/1KM28xzXwgQeUUj3zRMlth9BN-vs6Q98KBk1FHXmf58U/pubhtml

Some of the models in this list are already used by the **Spice Simulation** symbols referred to above. Some of these models have no dedicated symbols but they can be associated with existing symbols just by editing the symbol names.

Alternate models can be assigned to the Spice Simulation symbols by changing the names of the symbols to that of the new model and also - depending on whether the new model is a .model or a .subckt defined model - by pressing the `I` key and changing the spice prefix.

You can enter these model names into any approriate **Spice Simulation** symbol and then set the spice prefix to the prefix approriate for the model (See: **Ngspice model types** in: .model statements)

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

What to do for a part that has no model available in EasyEDA but for which a 3rd party model is available

It is also possible to import spice models from 3rd party sources to use with the **Spice Simulation** symbols from these list.

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

What to do if there is a 3rd party model available but for which there is no suitable symbol in EasyEDA

In the case of there being a 3rd party model available but for which there is no suitable symbol in EasyEDA then it is quite straightforward to create a spice symbol for it within EasyEDA either by editing an existing symbol or by creating a new one from scratch.

The steps to do this are described in:

Device models

but see also:

Schematic symbols: prefixes and pin numbers

Note that not all 3rd party models are compatible with ngspice syntax. `Spice3` versions of models should run out of the box. `Pspice` models may require modifications to make them work in EasyEDA.

Please post to:

Spice Simulation

in the forums for help on this.

What to do if there is no model available for a part

Not all components have a publically available model. In some instances, a component may not have a model at all.

However, we have created models - unique to EasyEDA - for a number of parts, such as the `LDR_EE`, `ELECTRET_MIC_EE`, `LM56EE` and `LM2907EE`. So, although it is not always possible or practical to build a model, if there is a part for which you particularly need one then please contact Support to discuss your requirements.

How to run a simulation

Please see:

How to run a simulation

and for more information:

Advanced probing and simulation control

Introduction

EasyEDA is not just a way to draw circuit diagrams and design PCBs. It is also a circuit simulator.

A circuit simulator is basically a specialised mathematical program, optimised to construct and then solve the equations that define the behaviour of the circuit that has been described to it in the circuit diagram.

The circuit simulation program that EasyEDA uses is called ngspice. Ngspice is Free and Open Source Software (FOSS) that is in turn based on a simulator called SPICE that was originally written by Larry Nagel.

What this book is for

This book is an introduction to circuit simulation in EasyEDA using ngspice.
It starts with the basics of how to avoid some of the most common mistakes that cause simulations to fail and then goes on to illustrate how to set up a circuit so that it will simulate successfully and produce meaningful results. It also discusses some aspects of understanding how, what might at first appear to be unexpected or even nonsensical results, can arise.
The book then introduces and illustrates more advanced techniques such as:

- probing signals such as voltages, currents, powers and resistances;
- configuring signal sources;
- setting up different types of analyses;
- making measurements such as rise times, RMS values and bandwidths;
- defining component values using parameters;
- using expressions to calculate component values such as for a resistor to draw a given load current or capacitances for a specified filter cutoff frequency;
- using manufacturer's device models;
- setting up complex simulations including arbitrary voltage and current sources.

What this book is *not* for

- This is not a book about learning to use EasyEDA to draw schematics. For general information about using EasyEDA please refer to the >EasyEDA Tutorial.
- This book does not teach electronics. Whilst there may be examples of circuits and explanatory text that are helpful in understanding electronics, it is assumed that the user already has sufficient knowledge of electronics to understand the content of this book.
- Although ngspice is similar to other variants of spice and a lot of the information and techniques in this book may be applicable to some of those variants, this book is written specifically about circuit simulation in EasyEDA using ngspice.
- Except where necessary to help explain some aspect of the behaviour of simulation, this book does not go into any detail of how circuit simulation in general and ngspice in particular actually works. For more information about those areas, please see the links below:

More information about Larry Nagel and SPICE is available from here:

http://www.omega-enterprises.net/The%20Origins%20of%20SPICE.html

Larry Nagel's PhD Dissertation:

> *Laurence W. Nagel., "SPICE2: A Computer Program to Simulate Semiconductor Circuits,"*
> *Memorandum No. ERL-M520, University of California, Berkeley, May 1975.*

http://www.eecs.berkeley.edu/Pubs/TechRpts/1975/ERL-520.pdf

is actually very readable and instructive.

For more information about electronic circuit simulation and spice in particular, see:

http://en.wikipedia.org/wiki/Electronic_circuit_simulation

and:

http://en.wikipedia.org/wiki/SPICE

More information about ngpsice is available from here:

http://ngspice.sourceforge.net/presentation.html

Who this book is for

All simulation tools and how they interact with schematic capture tools are different so even for people with experience of using simulation tools, it is worth at least skimming through the sections of this book to get an idea of where to find information if things don't seem to go quite as expected.

However, for people with limited or no previous experience of using simulation tools, this book is *essential* reading.

Why?

Because, as a newcomer to the world of circuit simulation, it is very tempting just to launch straight into trying to build and run lots of interesting and exciting simulations.

Sadly, however, this will almost certainly be a very frustrating and discouraging experience because many of those simulations will either not run, will not function correctly as circuits or will appear to give nonsensical results! This can happen for all sorts of often very simple reasons but to newcomers with no experience of simulation, those reasons can seem utterly incomprehensible.

So, to avoid a lot of discouragement and wasted time, it is worth taking a few minutes to understand some of the most common things that can cause a simulation to fail or to run but give unexpected answers.

How the book is structured

This book is not some dry tome with lots of words, diagrams and snippets of code. It is written as a live, interactive document. Rather than having to read lots of text and then having to go into EasyEDA to create circuits to try things out, live simulation examples are linked into the text to illustrate the points being discussed.

Just go to a topic or a section heading; read a couple of paragraphs and then open and run the embedded simulation to see exactly what the text is about.

Every embedded simulation must be saved before it can be run. Users who have joined EasyEDA can save the files to their own project folder. The examples can be copied and edited so that users can try out different things in whatever way helps their understanding of what the example is trying to explain, building up their own personal library of teaching examples as they progress through the book.

Newcomers are encouraged to work through the book in a linear manner because each section builds on the knowledge and insights gained from all the previous ones. Skipping sections leaves gaps that can catch out the novice and hinder the understanding of ideas presented in later sections.

- The book begins by introducing some of the concepts and terms that are essential for a basic understanding of what a simulator is and how to use it effectively.
- Then, using simple interactive simulation examples it describes and illustrates how to avoid the most common pitfalls in building, running and interpreting the results from simulations.
- The book goes on to illustrate ways to create and show connectivity using wires and netlabels in schematics. It then discusses techniques for probing voltages and currents and how they relate to and can affect or be affected by the netnames that are assigned to nets either automatically by EasyEDA or manually by the user placing netlabels.
- More advanced techniques for probing voltages and currents using the `probe` command are described, moving on to using this command to measure power dissipations, resistances and conductances.

This section describes the use of 0V sources to measure currents also introduces and E, F, G and H dependent sources and the very powerful arbitrary or B Dependent Sources.
In this section, the concept of the `Spice Directive` is also introduced, describing how to turn inactive `comment` text into active `spice` text that the simulator then recognises as an instruction to do something.

- The `probe` concept is then extended by describing how several different 'probe' commands can be typed into a schematic and - simply by selecting which one is switched from being 'comment' text into 'spice' text - they can be used to swap between different selections of values and nodes to be probed.
- The use of the `let` command to further extend and simplify the probing of signals is introduced with a simple example. This command will feature heavily in later sections about making measurements based on the results of simulations.
- Early interactive simulation examples introduce the basic way of setting up and running a simulation using the **Tools > Simulation > Simulate this Sheet** option to open the `Run your simulation` dialogue and then select the type of analysis to run.
- Later examples show how to use **CTRL+R** as a shortcut to the `Run your simulation` dialogue.

This idea is then extended with the concept of using `spice directives` to run simulations straight from **CTRL+R** so that, with a few simple key strokes, it is easy to switch between several different simulations from a single simulation schematic.

- DC operating point, DC Transfer function, DC Sweep, AC (frequency domain) and Transient (time domain) Analyses are described in some detail.

Using the DC Sweep to sweep the value of a resistance in a circuit and to sweep the ambient temperature for circuit is covered in this section.

There are detailed sections on setting up each of the various time domain signal sources that are available, such as SINE, PULSE and more.

Configuring and using the frequency domain AC source that is built into to every independent signal source is covered.

- The need for and different ways of defining initial conditions such as voltages on nets and across capacitors and currents in inductors are demonstrated.

Using PULSE, EXP, PWL and B Sources to 'kick start' circuits is also illustrated in this section.

- The concept of using parameters (i.e. variables that are used in the simulation) to define things like multiple component values and signal source settings is described.
- The use of parameters in expressions (i.e. formulas or equations) to calculate values such as the capacitance for a given RC time constant with a given resistance is covered and then extended into using them in B Sources.
- The concept of predefined functions and their use in expressions is described in detail extending into the creation of user defined functions using the `.func` statement.
- The concept and scope of device models in simulation is discussed in detail.
- Ways to add 3rd party models and subcircuits into schematics and to use them with the predefined schematic symbols from the EasyEDA Libs is covered in detail.
- The use of 3rd party models is extended into attaching them to custom schematic symbols.
- The use of expressions and functions in the creation of custom behavioural models is covered with reference to some of the in-house EasyEDA (EE suffix) models.

Simulation eBook - Introductory concepts of simulation

Introductory concepts of simulation

For every circuit being simulated, EasyEDA converts the schematic into a textual description of the circuit that is then passed to the simulator.

This textual description of the circuit is called a **spice netlist**.

- Note that the spice netlist is not the same as the netlist that is generated from a schematic and which is then passed through to PCB layout.

The netlist is a list of all the components used, how they are connected together and descriptions of them, called **models**, so that the simulator knows the behaviour of each component is to be simulated. The netlist also includes instructions to the simulator, called **spice directives**, to tell it what type of **analysis** is to be run. It may also include lists of values, called **parameters**, that are to be used to directly define component values or as part of more extensive calculations, called **expressions**.

Just like real ones, circuits to simulated require power sources and often signal sources. Therefore the netlist will include any **voltage sources** and **current sources**. These can be **Independent Sources** or **Dependent Sources**. Independent Sources just generate DC voltages or currents or can generate time domain signals such as sinusoids and pulses or they can generate a signal of a given amplitude and phase for frequency domain simulations (in fact they can be configured to generate all three but that will be covered later). Dependent Sources can also be used to generate DC voltages or currents but their main use is to generate outputs that are **functions** of that other signals in the circuit, for example by using an expression to describe the output of a source as a current that is equal to the sum of the squares of two input voltages.

The netlists of more advanced simulations can include instructions, called **measure statements**, to the simulator to perform calculations on the results of the simulation itself, for example to measure the rise time of a pulse, the RMS value of a signal or the 3dB bandwidth of a filter circuit.

Simulation eBook - About naming conventions

About naming conventions

Before going any further it is important to understand the naming conventions used throughout this book and throughout all the simulations.

Unless stated otherwise, text in Spice simulations is case insensitive.

Permissible characters are: `a` to `z`, numbers `0` to `9` and the underscore character: `_`.

Fields on a line are separated by one or more blanks, a comma, an equal (=) sign, or a left or right parenthesis; extra spaces are ignored.

A line may be continued by entering a `+` (plus) in column 1 of the following line; LTspice continues reading beginning with column 2.

A name field must begin with a letter (A through Z) and cannot contain any delimiters.

A number field may be an integer field (12, -44), a floating point field (3.14159), either an integer or floating point number followed by an integer exponent (1e-14, 2.65e3), or either an integer or a floating point number followed by one of the following scale factors:

Suffix	**Name**	**Factor**
T	Tera	1e12
G	Giga	1e9
Meg	**Mega**	**1e6**
K	Kilo	1e3
mil	Mil	25.4×1e-6
m	**milli**	**1e-3**
u	micro	1e-6
n	nano	1e-9
p	pico	1e-12
f	femto	1e-15

Letters immediately following a number that are not scale factors are ignored and letters immediately following a scale factor are ignored. Hence, 10, 10V, 10Volts, and 10Hz all represent the same number, and M, MA, MSec, and MMhos all represent the same scale factor. Note that 1000, 1000.0, 1000Hz, 1e3, 1.0e3, 1kHz, and 1k all represent the same number.

- Note that `M` or`m` denote `'milli`, i.e. 1e-3
- The suffix `Meg` MUST to be used for 1e6

Node names may either be plain numbers or arbitrary character strings, not starting with a number.

The ground node must be named `0` (zero). For compatibility reasons `gnd` is accepted as the ground node and will internally be treated as a global node and be converted to `0`.
LTspice and therefore, EasyEDA, requires that the following topological constraints are satisfied:

- Each circuit has to have a ground node (`gnd` or `0`)!
- The circuit cannot contain a loop of voltage sources and/or inductors and cannot contain a cut-set (series connected set) of current sources and/or capacitors.
- Each node in the circuit must have a dc path to ground.
- Every node must have at least two connections except for transmission line nodes (to permit unterminated transmission lines) and MOSFET substrate nodes (which have two internal connections anyway).

These constraints will be covered in more detail later.

Simulation eBook - Introduction to using a simulator

Introduction to using a simulator

Using a simulator is not quite the same as building a real circuit. There are many things that can catch out the newcomer to simulation because neither real world nor most simulator components are ideal. The departures from ideal are often different between the two and if these differences between real and simulated components in a circuit are not clearly understood, they can lead to confusion when the results of a even a simple simulation are different from those expected.

Part of the problem is that a user's expectations are informed by experience of real world measurements. Therefore, it is important to understand exactly what measurements are being made and how the simulation circuit will affect the results. In cases where there are differences between real and simulated results this may also require a deeper understanding of what is going on in those circuits and what assumptions are being made - often unconsciously - about them.

Learning to use a simulator means thinking more about what the real world *really* looks like, how it differs from the theoretical world of textbook problems and simple diagrammatic circuit representations and therefore what the results of measurements are likely to be.

Avoiding common mistakes

The following section describes and illustrates some of the most common mistakes, misunderstandings and causes for confusion.

All simulation circuits MUST have a ground node

A feature of the way simulators work is that they MUST have a ground node (also referred to as the 0 net) somewhere in the circuit. All voltages probed in the circuit, unless explicitly probed as voltage differences are then measured with respect to that ground node. The ground node can be placed anywhere in the circuit that is convenient for the purposes of the simulation but it must exist somewhere in that circuit.

> *SIMULATION SCHEMATICS THAT DO NOT HAVE A GROUND NODE WILL NOT RUN.*

This is illustrated in the following two examples:

With no ground symbol, the simulation will not run:

> All simulation schematics MUST have a ground 01

Once any of the available ground symbols is added, the simulation will run:

> All simulation schematics MUST have a ground 02

The ground node MUST NOT have a Voltage Probe attached to it

The reason for this is explained later in the section on **Probing Voltages**.

All simulation circuits MUST have a power and/or signal source

In exactly the same way that a real circuit must have some sort of power supply - even if that power supply is actually the signal source itself (for example, a crystal set or a volume control potentiometer) or comes from a capacitor pre-charged to some voltage or an inductor pre-charged to some current prior switching the circuit on - a simulation schematic must have a power and/or a signal source or an initial condition such as a capacitor pre-charged to some voltage or an inductor pre-charged to some current prior to the start of the simulation.

Power supplies can be built from ideal zero series resistance voltage sources. Simple but more realistic models of battery and voltage regulator supplies can be built using voltage sources with some series resistance or current sources with a parallel resistance (i.e. Thevenin or Norton equivalent sources).

EasyEDA provides a wide range of signal sources. These will be covered in detail in their own section on Sources but the basic use of some of these sources to provide power to a circuit is illustrated below:

> All simulation schematics MUST have a power supply too

Every point in a simulation schematic MUST have a DC path to ground

Unlike real circuits, every point in a simulation schematic MUST have a DC path to ground (or 0 net). Attempting to run a simulation with a node that has no DC path to ground will fail with errors.

Before discussing the significance of the DC path to ground, however, it is essential to look at some of the basic components and sources that are used in almost all simulations and to understand how they affect these DC paths.

Values and DC paths of RLC components

This section is about the values of - and DC paths through - resistors, inductors and capacitors.

- By default, inductors in LTspice have zero series resistance (i.e. ESR = 0).

> - Therefore, inductors have a DC path through them.

- By default, inductors in LTspice have no parasitic parallel capacitance or resistance.
- By default, capacitors in LTspice have no parasitic parallel conductance (i.e. they have an infinite DC resistance).

> - Therefore capacitors in LTspice do not have a DC path.

- By default, capacitors in LTspice have zero series resistance (i.e. ESR = 0). By default, capacitors in LTspice have no parasitic series inductance (i.e. ESL = 0).
- Inductors and capacitors in LTspice can be set to positive, negative or zero values. Beware that setting a component to a negative value may cause the circuit connected to them to exhibit instability or oscillation. This in turn may cause the simulation to fail.
- Resistors obviously have a DC path through them.

Resistors in LTspice can be set to positive or negative values but cannot be set to exactly zero: they MUST have a non-zero value. Setting a resistance to zero will cause the simulation run to fail with an error. This is basically because any voltage difference across a zero resistance (such as may occur normally in a circuit or even just as a consequence of numerical "noise" in the simulation calculations) will generate an infinite current, which will obviously crash out of the top end of the simulators calculation dynamic range. This applies to all resistors in LTspice, including the Ron and Roff values in switches and resistances given in device models. Beware that setting a resistor to a negative value may cause the circuit connected to it to exhibit instability or oscillation. This in turn may cause the simulation to fail.

- A common cause of confusion in simulation is the apparently unexpected behaviour of circuits using open circuit switches.

Just like switches in the real world, switches in EasyEDA have non-zero ON resistances (Ron). They also have finite OFF resistances (Roff). If an EasyEDA switch is connected in series with a voltage source into an open circuit load (such as presented by a Voltage Probe) then the voltage at the output of the switch will be equal to the open circuit voltage of the voltage source *whether the switch is open or closed*. This is different from the real world experience because in a real circuit, the voltage would be measured with a voltmeter or an oscilloscope probe having a much lower resistance than the open circuit resistance of the switch.

To illustrate this, let’ s take a simple example of a switch connected in series with an ideal voltage source set to 1V and then measure the V(Output) of the switch when open and closed.

With the switch closed the expected output is V(Output) = 1V.

And that’ s what the simulation shows.

OK. With the switch open the expected output is V(Output) = 0V.

What the simulation actually shows is V(Output) = 1V.

What’ s that all about? Surely the switch must not be simulating properly and is stuck closed?

Well, no actually; the switch and voltage are doing exactly what they should. It’ s just that these results are unexpected because there's a misunderstanding about how the simulated circuit differs from a real circuit.

No real switch has an infinite OFF state resistance but similarly, there is no such thing as a real open circuit that presents an infinite load impedance.

The Voltage and Current Controlled Switches and the Static Switches in EasyEDA are not ideal: they all have a finite OFF resistance. For the voltage and current controlled switches, the OFF resistances are specified by the user editable R(off)(Ω) parameter in the right hand panel with a default value of 1GΩ. The Static Switches have a fixed 10GΩ OFF resistance. In either case these resistances are large but certainly not infinite.

However, the effective resistance to ground at the Output node using a Voltage Probe or plot expression, really is infinite. Therefore when operating into an infinite resistance load there would be no difference between the ON and OFF state voltages at the Output node.

Note that if a load resistance of 1GΩ were to be connected from the Output node to ground, then there would be a clear ON/OFF state difference.

This is illustrated in the following simulation:

> EasyEDA switches are not ideal

Some further examples of the effects of finite switch resistances are illustrated in this next simulation:

> Effects of finite switch resistances

Note that, in LTspice, the Ron and Roff values used in switches can be set such that Ron > Roff. This is a useful way to invert the logic sense of a switch.

At this point it is worth noting that if it is important that the simulation results are as close as possible to those expected to be observed when probing a real circuit using real test equipment, it is sometimes useful to place a realistic load on wires (nets) in the simulation schematic where voltage measurements are to be made in the real circuit. Similarly for current measurements, realistic ammeter insertion impedances should be connected in series with the wire. To avoid unnecessary loading of the simulated circuit however, only place such loads in the locations where external measurement devices are to be connected to the real circuit and only in the same numbers as there are measurement instruments being used at the same time. For example, if there are two oscilloscope probes connected to a circuit but one of them is moved around, only connect loads representing simulated oscilloscope probes to two places in the simulated circuit. If different places need to be probed then move the simulated probes and rerun the simulation.

DC paths through Voltage and Current Sources

This section is about the DC paths through Voltage and Current Sources.

Note that this section only discusses the **default** settings of the LTspice voltage and current sources. For much more detail about the options available for these highly versatile devices please refer to the relevant sections of the LTspice native Help pages or the online pages at:

http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp.html

- Voltage sources in LTspice (including V, B, E and H voltage sources) have, by default, zero source resistance and have no current limit. The output voltage will be constant for any load current. This is true when sourcing or sinking current. sinking. This is not the same behaviour as regulated and current limited bench power supplies set to give a constant voltage output. Except for specialised supplies such as Source Measure Units, bench supplies can usually only source currents for positive outputs or sink currents for negative output voltages, up to some current limit set by the user.
- Default Voltage Sources in LTspice therefore have a DC path through them.
- As a consequence of their zero source resistance, in the same way that damagingly high currents will flow if real batteries are connected in parallel without some resistance between them, voltage sources in simulation schematics cannot be connected in parallel without some resistance in series between them. This is true even if they are set to exactly the same voltage. Attempting to do so will cause the simulation run to fail with an error.
- Similarly, in the same way that damagingly high currents will flow if an inductor is connected directly across a real power supply without some resistance between them, inductors in simulation schematics cannot be connected directly in parallel with voltage sources without some resistance in series between them. Attempting to do so will cause the simulation run to fail with an error.

- Current sources in LTspice (including I, B, F and G sources) have, by default, an infinite source resistance and have no voltage limit. The output current will be constant for any load impedance and voltage across the current source. This is not the same behaviour as regulated and current limited bench power supplies when they are set to give a constant current output. Except for specialised supplies such as Source Measure Units, bench supplies can usually only source currents up to some maximum positive voltage at the output or sink currents down to some minimum negative voltage at the output, as set by the user.
- Default Current Sources in LTspice therefore do not have a DC path through them.
- As a consequence of their generating a constant current through an infinite source resistance, in the same way that damagingly high voltages can be generated if two real current sources are connected in series without some finite resistance across each source, current sources in simulation cannot be connected in series without some finite resistance connected in parallel with each source. Attempting to do so will cause the simulation run to fail with an error.
- In simulation, connecting a capacitor in parallel with a current source generating a current, I, without also connecting a resistor, R, in parallel, will cause the voltage on the capacitor to ramp towards infinity. Even with the resistor in parallel, the voltage will ramp to I*R which may still be a large voltage. If such a circuit exists in a simulation schematic then, unless it is shorted out by a switch with some suitably low value resistance or set to some initial voltage, the voltage across the capacitor will already have ramped towards infinity at time t=0, i.e. as the simulation starts. This can lead to unexpectedly high voltages right from the start of a simulation.

The effects of adding DC paths

This section is about creating DC paths and some of the effects they can have.

- A favourite of elementary electrical engineering classes and an example of a very tricky problem to solve using a simulator is a circuit that has two capacitors in series. This must have a DC path to ground from the common point between the two caps but the path to ground does not have to be direct. It can be via other elements in a circuit. For example if one end of one of the capacitors is connected to ground with the other end of the other capacitor connected directly to a grounded voltage source or to a Thevenin Source (a voltage source in series with a resistance) or to a Norton Source (a current source in parallel with a resistance) then placing a single high value resistor in parallel with either of the capacitors or simply from the junction of the two capacitors to ground would be sufficient.

The following simple example will run but not converge because there is no DC path to ground from the common node B, between the two capacitors:

> Capacitors in series 01

Because the top end of C1 already has a DC path to ground through the ammeter and voltage source and the bottom end of C2 is grounded, simply adding a resistor to ground from the junction of the two capacitors at node B will allow this simulation to run. It will however, show voltages across the capacitors that can be confusing if the effect of the DC path resistance is not clearly understood:

> Capacitors in series 02

- Resistors placed across capacitors to provide a DC path can be scaled so that the R*C time constant that they will create with the capacitors is very large compared to the time interval over which a Transient simulation is to be run. Another way to look at this for an*

*AC Analysis is that the 1/(2*pi*R*C) frequency that they form is far outside the frequency range of interest.

> The value of these resistances can be anywhere between milliΩ (1e-3 Ω) up to 1G Ω (1e9 Ω). In many cases the values can be smaller or larger than this range but in some circuits that can lead to simulations failing with errors. This is usually because the ratio of the largest to the smallest voltages or currents in the circuit is too high for the simulator to handle. As a general rule, keeping the ratio between the smallest and the largest component value for any given type of passive component to no more than 1e12 should help avoid this sort of simulation failure.

- It must be noted however that whilst a purely theoretical analysis of the voltages across capacitors in series due to a current flowing through them based on Q=I*t=C*V, may yield sensible values, setting up both real and simulated circuits to demonstrate this can be quite challenging. Voltage measurements in real circuits can be difficult whilst the need for a DC path in simulations can present different but no less tricky problems.

Any attempt to try to measure the voltages across a pair of capacitors in series using DC sources will either end in a failed simulation, because there are no DC paths to ground, will run but give the wrong voltages across the capacitors, or will just yield the steady state DC voltages across the resistors used to create the necessary DC paths to ground.

There are a couple of ways to measure capacitor voltages in this sort of circuit. As illustrated below, the simplest is to replace the DC voltage source with a PULSE source configured to generate a fast edged step from 0 to 1V. As long as the RC time constant of the DC path resistance and the capacitors is large compared to the simulation stop time then the voltage across each capacitor will be equal to that given by a theoretical analysis and will be defined by the ratio of the capacitors.

> Capacitors in series 03

Another possibility is to apply a linearly ramped DC voltage of 1V/s across the capacitors in order to generate a constant current through them. The voltages across the individual capacitors will then rise in inverse proportion to their capacitances. By dividing these voltage by the time, a voltage can be generated that is equal to those which would be observed in a purely theoretical analysis of a 1V DC source with two capacitors in series thrown across it. The following example illustrates a way to actually measure the capacitances. It also introduces some of the more advanced uses of the B source which will be covered in more detail later on.

> Capacitors in series 04

- Similarly, a transformer coupled circuit where the primary side is driven from mains live and neutral and one side of the secondary circuit is connected to earth (which is probably where the ground symbol would be placed in the circuit diagram), must have a DC path back to this ground from one or both ends of the transformer primary (or from a centre tap if one is available).

> Similarly to placing resistors across capacitors, DC path resistors used with inductors can be scaled so that the R/L time constant that they will create is very large compared to the time interval over which a Transient simulation is to be run. Another way to look at this for an AC Analysis is that the 1/(2*pi*R*C) frequency that they form is far outside the frequency range of interest.

> Another example of where very high value resistors would be added is in transformer coupled data-communications networks such a 100BaseTx or 1000BaseT Ethernet.

> Here, the connections between the two transceivers are floating and so have to have DC paths to ground added to keep the simulator happy. Using resistances that are high in comparison to the network cable characteristic impedance and hence termination resistance is important in order not to introduce an impedance mismatch and so disrupt the signal integrity. In practice it is simplest therefore to add two resistors: one from each of the two signal wires to ground. This preserves the symmetry of the signalling on the wires and doubles the effective resistance between them.

> In many transformer coupled circuits, the simplest solution is to ground one side of both the primary and the secondary sides of the circuit. This may look strange because the circuits are no longer galvanically isolated by the transformer but - as long as the reason for connecting the circuits this way is clearly indicated in the schematic (and removed or otherwise modified before passing into PCB layout!) - then it is a simple and very useful solution. Not only does it remove any problems with how large or small the DC path resistor has to be, it also refers the voltages on both the primary and the secondary sides to ground and so simplifies probing of many of the voltages on both sides of the transformer which might otherwise have to be probed differentially.

> Some examples of this use of the ground return path resistor can be seen in the following collection of examples of transformers and coupled inductors, including a simple example of an open loop flyback converter:

> > Transformers and coupled inductors

> > Introduction to transformers in EasyEDA

> > Transformers and coupled inductors 1

> > Transformers and coupled inductors 2

> > Transformers and coupled inductors 3

> > Transformers and coupled inductors 4

> > Open loop flyback converter

- To avoid confusion between passive schematics intended to be passed to PCB layout and simulation schematics that may have had DC path resistors added, it is a good idea to clearly identify any resistors added to a schematic soley to provide a DC path to ground. This can be done using resistor names such as RDCPATH1 or by labelling them for example as *For simulation only*.
- It must be remembered that all voltages probed in a schematic are with respect to ground. This is particularly important to remember when probing signals that are floating, such as the transformer coupled examples discussed above. This is when `Diff_V_Probe` symbols, B or E Sources can be used to probe two floating voltages and subtract them to simply generate the difference between them.

Common problems with DC paths

A DC path to ground is often provided by the rest of the circuit but here are some cases that are often overlooked:

- All current sources (independent I and dependent B and F sources) are ideal: they have an infinite DC resistance (and AC impedance) and so do not have a DC path through them. Connecting a current source across a capacitor with one side grounded will throw an error even if the current source is set to zero (the capacitor voltage would ramp to infinity if a non-zero current were set and that would throw a different error!). So a resistor must be connected across the current source, to provide the DC path to ground to the other

side of the current source / capacitor. A similar problem arises if two current sources are connected in series even if the two currents are identical (if they're not then the common node flies off to infinity again);
- Switches have their own internal DC path between the switch terminals because they have finite OFF resistances. However, **voltage controlled switches** do not have a DC path between their control inputs.
- **Current controlled switches, Current Controlled Current Sources (CCCS)** and **Current Controlled Voltage Sources (CCVS)** all have a zero resistance short circuit between their control inputs. At first sight it might appear that connecting the output of a current source to the input of a current controlled switch would not cause a problem because the switch control input places a short circuit across the current source output. There is, however, no DC path to ground from the these connections so a resistor must be placed from one of the current controlled switch input pins to ground. If the current source output/switch control input part of the circuit is not connected to anything else then the resistor can be replaced by a connection to ground.

This is illustrated by the placement of the RDC_PATH_TO_GROUND 1k resistor connected to output side of the CCCS, F1, and input side of the current controlled switch W1 in the example below:

CURRENT CONTROLLED SWITCHES ARE NOT YET IMPLEMENTED IN LTspice VERSION OF EasyEDA

> Controlling EasyEDA switches

- Capacitors (unless using the more advanced options in LTspice) are ideal: they have no parasitic (leakage) DC resistance in parallel with them. This is why both ends of a capacitor must have a DC path to ground either through the external circuit or explicitly by adding a resistor;
- Although the primary and the secondary or secondaries of transformers have DC paths, there is of course no DC path from primary to secondary. This must be added either by connecting one side of the primary and of the secondary to ground directly or through a resistor.
- Voltage sources (including independent V and dependent B and E voltage sources) have the opposite problem. They are ideal so they have zero resistance. The same is true for simple inductors. If you connect voltage sources directly in parallel then LTspice will throw an error even if the voltage sources are set to the same value (if they're not then an infinite current would flow and that would then throw a different error). The same problem arises if you connect a voltage source directly in parallel with a simple ideal inductor because the voltage source tries to drive an infinite current through the inductor.

This problem often occurs when driving a transformer from a voltage source. Adding a small series resistance fixes this little gotcha (in a real inductor or transformer there will always be a finite winding resistance anyway!).

Components are connected by netnames

It is important to understand that although nodes (the pins on component symbols) in a schematic can be shown joined by wires or by netlabels, in a spice netlist in Sim mode (and the non-simulation schematic netlist in Std mode) they are joined purely by the net names given to them either automatically by EasyEDA or manually by the user placing Net Labels. This includes nets joined by any of the NetFlag GND ground symbols. It also includes nets joined by the Net Port, NetFlag VCC and NetFlag +5V symbols as illustrated below:

> Nets can be joined by netnames 01

It is equally important to understand that when a wire between two components in a schematic is broken then - unless the wire on both sides of the break is explicitly given the same netname by manually placing netlabels with the same name - then *the wires on each side of the break will have different netnames* because even though one side may already have a manually assigned netname, the other side will automatically have a new and arbitrary netname assigned to it by EasyEDA.

The significance of manually assigning net names will become apparent a little later when looking at how to probe voltages and using the voltage on nets in expressions for B sources.

> Nets can be joined by netnames 02

As just illustrated, breaking a wire creates two segments of wire that are no longer connected to each other. This is because EasyEDA automatically assigns different netlabels to each end of a broken wire.

To rejoin the connection, add netlabels with the same name to each segment of the broken wire. NetPorts and NetFlags can be placed anywhere on a net but take care that the little grey connection dot is placed onto the wire.

To avoid accidentally connecting nets together that are not intended to be connected when manually assigning netlabels, take care to ensure the netnames assigned are unique to each net. For instance, in the following example, using the name 'mid' instead of 'ammeterneg' would connect the negative side of A1 to the 'mid' net and would short out R1.

> Nets can be joined by netnames 03

Simulation eBook - Probing signals

Probing signals

After spending ages building a schematic the time has finally arrived: the first simulation is run. The **WaveForm** window pop opens and there are a bewildering number of traces. In fact there is a trace for every node voltage (i.e. the voltage on every net) and for the current through almost every component.

What's gone wrong?

The most likely cause is one of the most overlooked beginners' mistakes: there are no signal probes in the schematic. The simulation has run fine. It just saved and displayed traces for the voltage on every net and for the current through almost every component because it had not been told which particular subset of all the available node voltages and component currents to save and display.

In order to view useful output from a simulation without it being swamped in the WaveForm viewer, the circuit must have at least one voltage probe or one ammeter in the circuit. In basic simulations these will be the Voltage Probe and Ammeter symbols from the EasyEDA Libs. In more advanced simulations, these can be implemented using a `.probe` or `.save` command. In either case, at least one type of probe must exist in the simulation schematic. There is also a `Diff_V_Probe` that can be found in the Search Libraries tool. This probe produces the difference between the voltages on the A and B input pins on its (A-B) output pin to which an ordinary voltage probe can be attached or a net label can be attached and the output voltage displayed using a `.probe` or `.save` command referring to the relevant netlabel..

Probing voltages

All voltage measurements in real circuits are actually measurements of voltage differences. In many cases such as when probing a voltage using an oscilloscope probe, it is easy to forget that the voltage being measured is, in reality, the difference between the voltage at the probe tip and wherever the probe ground lead is connected. In the same way it is easy to forget that probing a single ended voltage in a simulation schematic is with respect to wherever the ground node has been placed.

- A common mistake, however, is to attach a voltage probe to ground.

> In spice simulations, all voltages are referred to ground so not only is there no need to attach a Voltage Probe to the ground net, in fact doing so will throw an error in the simulation.

In a real circuit, probing a voltage between any two points places a resistive load between them. With a good quality voltmeter that resistance may be very high, in the order of hundreds of MegΩ. With a x10 oscilloscope probe it will be 10MegΩ. There will be some stray capacitance across that resistance. There will also be stray lead inductances. If the voltage being measured is an AC signal then impedances due to these stray and parasitic components will also load the circuit.

Note that in simulations, voltage probes present an infinite resistance and have no stray capacitance or inductance. In effect, voltage probes have an infinite bandwidth.

The following example illustrates some of the probing techniques described above:

> Probing voltages 01

The following example shows a number of ways to measure voltages with respect to ground or differentially using;

- The `Diff_V_Probe` with a Voltage probe to probe the output;
- An `E` source (a.k.a. Voltage Controlled Voltage Source or VCVS) with a Voltage probe to probe the output;
- A `B`, source (a.k.a. behavioural or dependent source) configured as a VCVS using a Voltage probe to probe the output.

The schematic also demonstrates the importance of:

- Giving voltage probes names that are identical to the nets to which they are attached;
- Naming all nets in a schematic;

> Probing voltages 02

Probing currents

In a real circuit, probing the current in a wire places a resistive load between them. This will cause some voltage drop across the ammeter. With a good quality ammeter that voltage drop may be very low, in the order of millivolts. There will be some stray capacitance across the insertion resistance and from the ammeter connections to ground. There will also be stray lead inductances. If the current being measured is an AC signal then impedances due to these stray and parasitic components will also load the circuit.

Note that in simulations, except for the Ammeter, which has a 1uΩ (1e-6Ω) series resistance, current probes present zero insertion resistance and have no stray capacitance or inductance. In effect, current probes have an infinite bandwidth.

The following example shows a number of ways to measure currents with respect to ground or differentially using;

- As a current using the Ammeter symbol;
- As a linearly scaled voltage using an H Current Controlled Voltage Source (CCVS) (or an F Current Controlled Current Source (CCCS) with a resistor);
- As a linearly scaled current using an F Current Controlled Current Source (CCCS) driving an Ammeter;
- As a voltage that can be an arbitrary function of the current flowing through a 0V Voltage Source using a BV source (or a BI source with a resistor);
- As a current that can be an arbitrary function of the current flowing through a 0V Voltage Source using a BI source driving an Ammeter.

Note however, that although a 0V source can be used to *monitor* a current, it cannot be used to *measure* a current so that it can be directly displayed in the **Simulation > Show your simulation report...** window or plotted in Waveform.

> Probing currents 01

Simulation eBook - Advanced probing and simulation control

Advanced probing and simulation control

So far, the examples have used two different ways of running the simulations:

- Using the **CTRL+J** Hotkey as a shortcut to the `Run your simulation` dialogue.
- Using `spice directives` to run simulations straight from the **F8** Hotkey (was CTRL+R) so that, with a few simple key strokes, it is easy to switch between several different simulations from a single simulation schematic.

The following sections will show how to do more advanced signal probing which, when combined with the use of spice directives to control simulations and the flexibility of the

> **Text type > spice/comment**

text attribute to switch between sets of probing and simulation commands, takes simulation to the next level.

The `probe` command

The section on probing signals showed how to probe voltages and currents in a schematic by adding symbols to the schematic in much the same way real pieces of test equipment such as DVMs and scopes and ammeters can be used to measure voltages and currents in a real circuit.

This section and the associated examples will introduce the '.probe' and `.save` spice directives to measure voltages and currents in a circuit.

This section reinforces the importance of naming all nets as discussed earlier.

What the `probe` and `.save` spice directive does

In LTspice, the '.probe' and `.save` spice directive do exactly the same jobs:

- Replaces voltage probes;
- Provides an alternative way to probe currents simply by placing a 0V Voltage source anywhere a current is desired to be measured;
- Allows several `.probe` or `.save` statements to be put into a single schematic and using:

> **Text Attributes > Text type > comment/spice**

provides an easy way to select between probing different sets of measurements by switching the state of probe statements between `comment` (simple inactive text which is ignored by the simulator) and `spice` (interpreted by the simulator as active spice directives).

For simplicity in the rest of this document, `.probe` statements will be referred to and demonstrated but all comments apply equally to `.save` statements.

The syntax of a `.probe` directive

- The `.probe` directive comprises the keyword ‘.probe’ followed by a space separated list of voltages or currents but not expressions;
- All probed signals must be in a single line following the keyword ‘.probe’ , with no line breaks;
- There can be any number of `.probe` statements active in a simulation schematic;
- Voltages can be expressed in the form: `netname` or `V(netname)`, where 'netname' is the name of the net on which the voltage is to be probed;
- Currents are expressed in the form: I(V_source_name); where `V_source_name` is the name of the voltage source through which the current to be probed is flowing;

These points are illustrated in the following sections.

Using the '.probe' command to make measurements

The examples, Probing voltages 01,
Probing voltages 02] and Probing currents 01 have shown how to probe voltages and currents in a schematic by adding symbols to the schematic in much the same way real pieces of test equipment such as DVMs and scopes and ammeters can be used to measure voltages and currents in a real circuit.

This next example introduces the '.probe' spice directive which can be used to measure voltages and currents.

> Using the probe command

Probing instantaneous power

Unfortunately in EasyEDA, the '.probe' directive cannot be used to measure power by probing the voltage difference across a device and the current through it and then simply multiplying the two signals together using an expression.

It is however, possible to measure the instantaneous power dissipation in a component by using the `pwrmeas` probe spice symbol. The power is calculated from the instantaneous voltage across the device multiplied by the instantaneous current through it and is represented as a voltage scaled as 1V/Watt. The output of the probe therefore needs to have a voltage probe attached to it in order to view the voltage representing the power.

> How to make instantaneous power and resistance measurements

Probing resistances

Unfortunately in EasyEDA, the '.probe' directive cannot be used to measure resistance by probing the voltage difference across a device and the current through it and then simply dividing the two signals together using an expression.

It is however, possible to measure the instantaneous resistance in a component by using the `resmeas` probe spice symbol. The resistance is calculated from the instantaneous voltage across the device divided by the instantaneous current through it and is represented as a voltage scaled as 1V/Ohm. The output of the probe therefore needs to have a voltage probe attached to it in order to view the voltage representing the resistance.

> How to make instantaneous power and resistance measurements

Using F8 (was CTRL+R) to run a simulation directly

This section introduces the idea of placing the simulation directive directly into the schematic. This makes running a simulation even easier because instead of using the:

> **CTRL+J > Select simulation type > Run**

type of key sequence to run a simulation, the key sequence simplifies to just:

> **F8**

(was CTRL+R)

Not only does this make running a simulation easier but it also extends the idea - introduced in `The 'probe' command` - of being able to quickly and easily change the selection of traces to be displayed in WaveForm, to being able to change the actual simulation to be run as well.

Adding the simulation directive directly into the schematic is easy.

All that has to be done is to type the directive as a line of text in the schematic and then do:

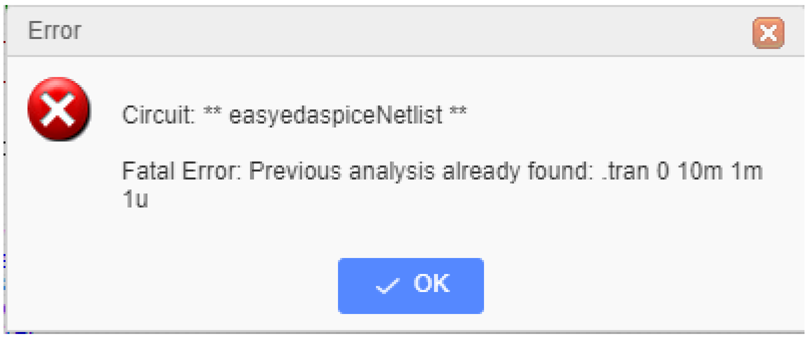
> **Text Attributes > Text type > spice**

to turn the passive text (blue font) into an active spice directive (black italic font).

Note that one and only one simulation directive can be active for any one simulation run. The example below illustrates how to do this:

> Using F8 (was CTRL+R) to run a simulation directly

Note also that doing F8 without having first placed an active simulation directive in the schematic sheet will automatically place a default .tran simulation directive in the schematic sheet in large text. If this is not removed or rendered inactive by converting it to plain inactive text, before another active simulation directive is manually placed in the schematic sheet then an error message may appear similar to this:


Error

Circuit: ** easyedaspiceNetlist **

Fatal Error: Previous analysis already found: .tran 0 10m 1m 1u

The same type of error message may appear after attempting to run a simulation with more than one active simulation directive in the schematic sheet.

Simulation eBook - Configuring Voltage and Current Sources

Configuring Voltage and Current Sources

EasyEDA Libs provides a range of Voltage and Current Sources whose outputs are defined by a list of values or parameters. The outputs do not depend on anything else.

These sources have already been discussed in terms of their DC source resistances and DC paths.

In most of the examples so far, they have been used to provide DC supply voltages either as ideal voltage sources or as Thevenin or Norton Sources.

Their use to provide time domain (time varying) signals has been introduced in the examples about transformers but has not so far been explained.

This section describes in detail how any V (and I) source can be set up to provide the following types of signal sources:

- SINE or SIN: a sinusoidal signal;
- PULSE: a general pulse waveform;
- EXP: a single pulse with exponential rising and falling edges;
- SFFM: (Single Frequency Frequency Modulated) a single sinusoidal carrier, frequency modulated by single sinusoidal frequency;
- PWL: (PieceWise Linear sources} an arbitrary waveform source with signals created as a list of times and levels with the signal linearly interpolated between each time point.

Although the examples in this section only illustrate how to configure V Sources, I Sources are configured in exactly the same way.
Please note that the descriptions and examples of the different sources describes the most common configurations. LTspice offers several unique V and I Source features which are beyond the current scope of this document. To find out about these advanced options please search for V and I in the native LTspiceXVII Help files either in a locally installed copy or from the LTwiki site here:
http://ltwiki.org/index.php?title=LTspice_Annotated_and_Expanded_Help*
Even more advanced options are described on the LTwiki site here:
http://ltwiki.org/index.php?title=Undocumented_LTspice#Standard_Sources

Configuring the SIN (or SINE) option

Configuring the SINE option to create an unmodulated, single frequency sinusoidal signal source.

> Spice Sinusoidal Source

More ways to use the SIN (or SINE) option

> Spice Sinusoidal Source: more examples

Configuring the PULSE option

Configuring the PULSE option to create a pulse signal source.

> Spice PULSE Source

More ways to use the PULSE option

> Spice PULSE Source: more examples

Configuring the EXP option

Configuring the EXP option to create a single pulse source with exponential rising and falling edges.

> Spice EXP Source

Configuring the SFFM option

Configuring the SFFM option to create a simple, single frequency, frequency modulated sinusoidal signal source.

> Spice SFFM Source

Configuring the PWL option

Configuring the PWL option to create an arbitrary piecewise linear waveform signal source.

> Spice PWL Source

Configuring the AC source option

As well as their more obvious use to generate time domain signals for Transient Analysis (time domain) simulations, V and I Sources can also be configured as AC Sources for AC Analysis (frequency domain) simulations. The amplitude and phase of the AC Source is specified and a list of frequencies at which the circuit is to be analysed is given in an AC Analysis spice directive. The result of an AC Analysis is a set of amplitudes and phases that are plotted to create an Amplitude and Phase versus Frequency plot such as a Bode plot showing the frequency response of a circuit.

It is important to understand that these plots are not generated as they would be for a real world, physical circuit where a Frequency Response Analyser presents a sinusoidal signal, either swept slowly or stepped at a series of discrete frequencies, to the input of a circuit and then the amplitude and phase of an output at some other point in the circuit is measured, relative to the input signal, in the time domain.

The way they are generated is the result of a purely mathematical analysis. In simple terms, the DC operating point of the circuit is examined and all the components in the circuit are replaced by their linearised small signal models. In other words, everything is assumed to be linear about the DC operating point so that the circuit can be represented as a small signal linear system in the frequency domain. The output from that set of linear equations is then solved at each of the input frequencies specified in the AC Analysis spice directive.

It can be quite hard to visualise what the Amplitude and the Phase settings in the AC Source options of the V and I Sources really mean when the signals in an AC Analysis cannot be viewed in the time domain in a Transient Analysis. To try to help visualise these settings and what they represent, the following couple of examples demonstrate the settings in ways that can be related back to their equivalents in the time domain.

The first example shows how more than one AC Source can be configured in a circuit to represent different signal sources at the same frequency but with different phases. The example also shows how the phase settings relate to the same signals in the time domain.

In this example both AC Sources are set to the same amplitude of 1. They could be set to different amplitudes: try it and compare the results with the same amplitude changes in the time domain part of the signal sources.

> Configuring AC Sources 01

Note, however, that the AC Analysis assumes that the circuit is perfectly linear so even if an AC Source amplitude 100 were to be specified, the output would still look as if it came from a perfectly linear circuit. Compare that with what happens if the time domain parts of the sources are set to 100!

The DC offset of the inphase AC Source in this example is important because it biases Q1 into a range where both the emitter and collector swings are operating in the linear region.
This can clearly be seen by probing V(Q1E) and V(Q1C) in a Transient Analysis. If the DC Offset is increased, eventually V(Q1C) falls and V(Q1E) rises will until they meet as Q1 saturates and V(Q1E) starts to pull V(Q1C) back up again. At the point where this happens the small signal gain of the collector output passes through zero and then becomes a non-inverting gain of somewhat less than unity.

If the DC offset is reduced to near ground or even below it, Q1 is cut off so both the collector and the emitter output small signal gains fall effectively to zero. Again this can clearly be seen in a Transient Analysis.

What is not so obvious is that although these effects still occur in the AC Analysis, because the DC conditions cannot be represented in the frequency domain plots, the results can sometimes be hard to interpret.

So, the information to take away from this is that if an AC Analysis seems to be showing a lower than expected gain then it is worth checking that the DC operating point of the circuit is not forcing some part of it into saturation or cutoff. One example of this is incorrectly biasing an opamp input so that the output has hit one or the other of the supply rails. Forgetting to connect up a power supply rail is another common mistake.

An AC Analysis can only be used to study the small signal frequency response of a circuit. Since the linearity of most circuits varies with the instantaneous value of the input signal the results of an AC Analysis cannot be used to infer the large signal response of a circuit in the frequency domain.

Care must also be taken in setting up the correct DC operating point when applying an AC Analysis to circuits including AGC and other forms of dynamic range compression and expansion where the DC operating point is set by some long term (compared to the period of the signal) averaging, or similar function of the amplitude of the output of the circuit itself.

A further point about the effects of the DC operating point is that an AC Analysis can not be used to study the frequency response of circuits such as Phase Locked Loop, Switch Mode Power Supplies and Class D amplifiers because these circuits usually contain elements that are always switched into one state or another where the linearised gain is reduced to zero. There are ways to study the frequency responses of such circuits but they require more advanced modelling techniques to replace the switching and other elements (such as the VCO in a PLL) with linearised equivalent circuits.

It must also be understood that although any number of AC Sources can be placed in a circuit, each with their own amplitude and phase, all the sources will operate at exactly the same frequency as this is determined by the AC Analysis settings and not by the sources themselves. In a circuit with several Independent Sources in it, AC sources can simply be added to, removed from or moved around it just by adding the AC amplitude and phase values to the required source. So in the example above, the response of the inphase and out of phase side of the all pass network can be observed simply by setting one AC Source or the other to have zero amplitude or just by deleting the AC parts of the Source configuration.

Another example of this might be that the frequency response of an amplifier from signal input to output can be plotted using an AC Source at the input source whilst the frequency response of the amplifier from power supply ripple to output can be plotted by swapping the AC Source settings to the voltage source being used for the power supply.

Simulation eBook - Setting up Analyses

Setting up Analyses

What are Analyses?

An analysis is simply an instruction to the simulator telling it what type of simulation to carry out on the spice netlist.

There are several different types of analysis that can be carried out when a simulation is run. EasyEDA directly supports a subset of the spice analyses that are available in LTspice. These analyses are accessed via **CTRL+J** and are described below.

SPICE Analyses available via CTRL+J

CTRL+J opens the **Run your simulation** dialogue box.

The **Run your simulation** dialogue box offers the following SPICE analyses.

1) DC op pnt
2) DC Transfer
3) DC Sweep
4) AC Analysis
5) Transient

For more information about what each analysis does, please scroll down to the relevant sections.

SPICE Analyses and Control Statement Syntax

The SPICE analyses listed above can also be entered directly into a text box in a schematic. Several analysis statements can be entered in a single schematic but one and only one can be made active for any one simulation run. To make an analysis statement active, do:

> **Text Attributes > Text type > spice**

To make an analysis statement inactive, do:

> **Text Attributes > Text type > comment**

When a SPICE analysis is placed directly in the schematic and made active, it can then be run simply by doing:

> **F8**

(was CTRL+R)

By entering a SPICE analysis directly in the schematic it is then possible to access some of the advanced options for some of the analyses.

A link to more information for each analysis is given in the relevant sections below.

1) .OP: Perform an Operating Point Analysis

General form:

> **.op**

Example:

> **.op**

Causes LTspice to perform an operating-point analysis to determine the quiescent DC Operating point of the circuit with inductors shorted and capacitors opened. The results of this analysis are used to calculate values for the linearised, small-signal models of nonlinear devices.

More information:

http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp/html/DotOp.htm

2) .TF: Perform a DC Transfer Function Analysis

The dc transfer function analysis portion of SPICE computes the following small signal characteristics:

- the ratio of output variable to input variable (gain or transfer gain)
- the resistance with respect to the input source
- the resistance with respect to the output terminals

The TF statement can be used to find the Thevenin small signal equivalent resistance. (The Thevenin voltage is given by the node voltage at the open circuit terminal, as a result of the OP statement).

General form:

> **.tf OUTvar inSRC**

Examples:

> **.tf V(5, 3) VIN**

> **.tf I(VLOAD) VIN**

The .TF command defines the small-signal output and input for the DC small-signal analysis. OUTvar is the small-signal output variable and inSRC is the small-signal input source. If this line is included, SPICE computes the DC small-signal value of the transfer function (output/input), input resistance and the output resistance.

More information:

http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp/html/DotTf.htm

3) .DC: Perform a DC-Sweep Analysis

During a DC-sweep analysis SPICE steps the value of a specified independent voltage or current source over the user-specified range and performs an operating point analysis at each value. This permits the evaluation of the DC transfer function, and also provides a mechanism for plotting the characteristic curves of devices and models.

General form:

> **.dc Source-Name Vstart Vstop Vincr >[Source2 Vstart2 Vstop2 Vincr2]**

Examples:

> **.dc vin 0.25 5.0 0.25**

> **.dc vin 0 10 .5 vgs 0 5 1**

> **.dc vce 0 10 .25 ib 0 10u 1u**

> **.dc R1 0 1k 100**

> **.dc TEMP 0 100 1**

The parameters define the dc transfer-curve source and sweep limits. Source-Name is the name of an independent voltage or current source, a resistor or the circuit temperature. Vstart, Vstop, and Vincr are the starting, final, and incrementing values respectively. The first example causes the value of the voltage source vin to be swept from 0.25 volts to 5.0 volts in increments of 0.25 volts. A second source (Source2) may optionally be specified with associated sweep parameters. In this case, the first source is swept over it's range for each value of the second source.

The following simulation illustrate a DC Sweep of a single voltage source:

> Plot and compare diode forward currents vs. voltage

4) .AC: Perform a Small-Signal AC (frequency domain) Analysis

The ac small-signal portion of SPICE computes the ac output variables as a function of frequency. The program first computes the dc operating point of the circuit and determines linearized, small-signal models for all of the nonlinear devices in the circuit. The resultant linear circuit is then analyzed over a user-specified range of frequencies. The desired output of an ac small-signal analysis is usually a transfer function (voltage gain, transimpedance, etc). If the circuit has only one ac input, it is convenient to set that input to unity and zero phase, so that output variables have the same value as the transfer function of the output variable with respect to the input.

General form:

> **.ac (DEC | OCT | LIN) N Fstart Fstop**

Examples:

> **.ac dec 10 1 10K**

> **.ac dec 10 1k 100Meg**

> **.ac lin 100 1 100Hz**

Use:

> 'dec' for decade variation, in which case N is the number of points per decade;

> 'oct' for octave variation, in which case N is the number of points per octave;

> 'lin' for linear variation, in which case N is the total number of points.

Please note however that due to limitations in the WaveForm tool, only log axes are currently implemented in EasyEDA so the 'oct' and 'lint' sweep options are of limited use.

Fstart is the starting frequency, and Fstop is the final frequency.

More information:

http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp/html/AC_analysis.htm

Please note however that due to limitations in the WaveForm tool, the syntax ".ac list " with a single analysis frequency is not implemented in EasyEDA.

5) .TRAN: Perform a Transient (time domain) Analysis

The transient analysis portion of LTspice computes the transient output variables as a function of time over a user-specified time interval. The initial conditions are automatically determined by a dc analysis. All sources which are not time dependent (for example, power supplies) are set to their dc value.

General form:

> **.TRAN [Tstart [dTmax]] [modifiers]**
> **.TRAN [modifiers]**

The first form is the traditional .tran SPICE command. Tstep is the plotting increment for the waveforms but is also used as an initial step-size guess. LTspice uses waveform compression, so this parameter is of little value and can be omitted or set to zero. Tstop is the duration of the simulation. Transient analyses always start at time equal to zero. If Tstart is omitted, it is assumed to be zero. However, if Tstart is specified, the waveform data between zero and Tstart is not saved. This is a means of managing the size of waveform files by allowing startup transients to be ignored.

In the interval prior to Tstart, the circuit is analyzed (to reach a steady state), but no outputs are stored. In the interval (Tstop-Tstart), the circuit is analyzed and outputs are stored.

The final parameter dTmax, is the maximum time step to take while integrating the circuit equations. If Tstart or dTmax is specified, Tstep must be specified but is usually set to zero.

Note that in EasyEDA, dTmax is limited as: dTmax=(Tstop-Tstart)/1000. This is a limitation of EasyEDA to limit server usage. This is not an inherent limit in native LTspice run on a local machine.

Several modifiers can be added to the end of the .tran statement. The most useful is probably:

> startup: Solve the initial operating point with independent voltage and current sources turned off. Then start the transient analysis and turn these sources on in the first 20 us of the simulation.

This simulates running a circuit from when it is first switched on.

> UIC: **Note that the 'uic' option must be used with caution.**

Normally, a DC operating point analysis is performed before starting the transient analysis. The results of this DC operating point analysis provide the initial conditions for the circuit at time t=0.
The 'uic' spice directive suppresses this initialization.

The initial conditions of some circuit elements can be specified on an instance-per-instance basis. For example: transistors can be specified to be in an OFF initial state; switches can be specified to be in an ON or an OFF initial state; the .IC spice directive allows the voltages on nets and the currents in inductors at t=0 to be specified.

If the 'uic' option is added to a tran spice directive then all specified initial conditions are used. It is important to realise however that if the 'uic' directive is used without explicitly stating the initial conditions then, because the DC operating point analysis is omitted, default values are assumed. This can cause problems in some simulations because the default values can lead to non-physical initial conditions around a circuit. For example, consider an ideal voltage source connected in parallel to an ideal capacitor. Unless it is specified otherwise, the default initial value of the voltage source is taken as zero. Therefore, the voltage across the capacitor is also zero at t=0. Then, in the first time step, the voltage source is set to the operating output voltage so an infinite current is drawn from it to charge the capacitor up to this operating voltage. The simulator cannot find a short enough time step to make this current finite, and a "time step too small convergence fail" message is issued.

For information on these and other modifiers that can be applied to the .tran directive, please refer to:

http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp/html/DotTranModifiers.htm

Examples:

> **.tran 1m**

> **.tran 1m startup**

> **.tran 0 1000ns 500ns 10ns**

> **.tran 10ns 1us 0us 20ns uic**

.IC: Set Initial Conditions

General form:

> **.IC [V()=] [I()=]**

Example:

> **.IC V(in)=2 V(out)=5 V(vc)=1.8 I(L1)=300m**

The IC line is for setting initial transient conditions. It has two different interpretations depending on whether the UIC parameter is specified on the .TRAN control line. One should not confuse this line with the .NODESET line. The .NODESET line is only to help DC convergence, and does not affect final bias solution (except for multi-stable circuits). The two interpretations of this line are as follows:

1) When the uic parameter is specified on the .tran line, then the node voltages specified on the .ic control line are used to compute the capacitor, diode, BJT, JFET, and MOSFET initial conditions. This is equivalent to specifying the `ic=...` parameter on each device line, but is much more convenient. The `ic=...` parameter can still be specified and takes precedence over the .ic values. Since no dc bias (initial transient) solution is computed before the transient analysis, one should take care to specify all dc source voltages on the .ic control line if they are to be used to compute device initial conditions.

2) When the uic parameter is not specified on the .tran control line, the dc bias (initial transient) solution is computed before the transient analysis. In this case, the node voltages specified on the .ic control line is forced to the desired initial values during the bias solution. During transient analysis, the constraint on these node voltages is removed. This is the preferred method since it allows LTspice to compute a consistent dc solution.

Note that if the 'uic' option is not used then any .IC directives included in the simulation are used anyway.

Simulation eBook - Initial conditions and starting up circuits

---

Initial conditions and starting up circuits

Some background and basic start-up techniques

There are several important points to understand about what happens at the start of a simulation.

All simulations start at time t=0.

Whatever DC conditions are applied to the simulation at t=0 are considered to have been present for all time t<0.

Even though it is possible to start plotting the results of a time domain simulation at some time after t=0, the simulation itself always starts at t=0.

Even though an AC Analysis has no time duration associated with it, the DC operating point is calculated at t=0.

If there are no .IC statements included in a simulation and the Startup and UIC modifiers are not used, then the initial DC conditions of a circuit are completely defined by the initial levels or DC offsets of any sources present in the circuit and these are treated as having been present at those levels for all time prior to t=0. So for example, a DC power rail that is set to

9V is treated by the simulator as having been applied to the circuit at 9V for all time prior to t=0.

Similarly, a SINE source with a -1V DC offset or a PULSE source with an initial level of -1V are treated by the simulator as having been at -1V DC for all time prior to t=0. Similarly, a COS source with a 0V DC offset would be treated by the simulator as having been at +1V DC for all time prior to t=0.

Therefore the voltages in a circuit, such as bjt base bias potential dividers and so on, will have been set to DC steady state values for all t<0. Consequently, the voltages across all capacitors and the currents through all inductors in the circuit will have reached their DC steady state values prior to the simulation starting at t=0.

Even if the circuit is an oscillator, prior to t=0 it will have been assumed to be in a stable non-oscillating steady state. At t=0 the circuit will then start from those initial DC conditions. It will then either continue in that steady non-oscillating state or will slowly drift away from the steady DC state and oscillations will build up.

The initial state of oscillators based on a tuned circuit such as phase shift, Wien Bridge and Crystal Oscillators will be defined by their DC bias conditions. If there are no noise sources in the circuit (the default state for all components unless otherwise specified such as resistors defined to have noise contributions) then there is nothing to nudge the circuit away from equilibrium and so it may never start oscillating.

Although in most cases such oscillators will eventually start up due to the 'hidden' noise source which is simply due to the mathematical noise generated by the finite resolution and rounding errors of the calculations carried out in running a simulation, this can take a very long time compared to the time taken to run the oscillator in a stable oscillatory state for a few cycles. Crystal oscillators in particular can take many hundreds of thousands of times the oscillator period to start up and reach a stable state.

To minimise the simulation time spent waiting for an oscillator to start, it is useful to introduce some initial start-up condition to 'kick-start' the circuit into oscillation.

The simplest way to kick-start most circuits in LTspice is to run a Transient (.tran) Analysis with the **startup** modifier appended to the end of the .tran directive. For example:

.tran 1m ; runs a time domain simulation for 1ms starting with the initial DC conditions

.tran 1m startup ; runs a time domain simulation for 1ms starting with the independent sources all set to zero and then ramps them up in the first 20us or so of the simulation.

Another simple way to kick-start an oscillator based on a tuned circuit is to replace a simple DC supply source with a PULSE source set to an initial level of the desired power supply voltage but configured to generate a short pulse of the supply voltage plus or minus some small voltage. So for example a circuit with a 9V supply that is to run for 1ms with a time step of 1us could have a PULSE source set to an initial level of 9V pulsed down to 8.5V for 1us with 100ns rise and fall times. Or an initial level of 8.5V with a delay of 1us before a 100ns rise-time step up to a pulse level of 9V.

The same trick can be carried out using a voltage source inserted almost anywhere in the circuit simply to inject a small step or pulse into a bias voltage but it must be remembered that if the voltage step or pulse does not return to 0 after the kick then it will represent an offset voltage in that part of the circuit.

An example of this is in forcing crystal oscillators to start.

Crystal oscillators take a very long time to start up because of the extremely high Q of the crystal.

The same thing is true of their simulations so, to avoid simulations taking too long to run and generating massive data files, they may need to be run in stages with increasing start and stop times but a small (Tstop - Tstart) value.

As discussed earlier, injecting a small step into the supply voltage or an internal node of the circuit can help to start the oscillations a little earlier but the idea of injecting a small step or pulse into a circuit to kick it into life is taken to an extreme in the EasyEDA crystal model.

The technique used in the XTALfast subckt is to include a PULSE source internal to the crystal model to introduce a very high amplitude impulse at t=0 inside the model. This starts the oscillator almost instantly and because an impulse with a very wideband spectrum is used rather than a step (or a sinusoidal burst at the crystals natural resonant frequency), the oscillations start at the actual resonant frequency of the crystal in the application circuit.

The start up time of the example below using the XTALfast subckt can be compared with the same crystal model but with an unassisted start simply by editing the name of the crystal model from XTALnokick to XTALkick.

This example also demonstrates a way to use the advanced features of the LTspice capacitor model to create a more computationally efficient crystal simulation but one which unfortunately cannot be kick started like the EasyEDA model.

> Crystal oscillator using the EasyEDA quick starting crystal model

The EXP and PWL sources can also be used as kick-starter supply sources.

Most relaxation oscillators such as the classic two transistor astable multivibrator or the 555 timer have two stable states. Oscillation is normally maintained by the circuit switching between these two states however, under DC bias prior to t=0, these circuits often settle into one or the other of these stable states and so are stuck there at t=0. This means that they never start oscillating.

This is an example of a simple RC relaxation oscillator that does not start up by itself:

> Relaxation oscillator startup 01

This type of circuit may need a rather more vigorous kick to get it started. This can be done using a PULSE source but instead of introducing a small step, the supply is ramped up to the desired supply voltage from 0. So, for example, by setting an initial level of 0 and a pulse level of 9 with zero delay time, with a rise-time of 200us the circuit starts up cleanly up from all internal nodes being at zero.

This example also demonstrates starting up the same circuit with only a simple DC simply by adding the startup modifier to the .tran statement to ramp the DC source automatically:

> Relaxation oscillator startup 03

Another possibility - which will be explained in more detail later - is to use an expression that is a function of time in a B source.

For example, this expression in a B source:

> **V=9*(1-exp(-time/100u))**

generates a voltage that starts at zero and rises exponentially to a final value of 9V with a time constant of 100us:

> Relaxation oscillator startup 04

With symmetrical circuits such as the two transistor astable multivibrator, even this may not be enough to disturb the equilibrium enough to get them oscillating. It may be necessary to introduce some deliberate asymmetry or imbalance into the circuit, for example by making a one base pullup resistor or a timing capacitor a fraction different from the other. Even changes of less than the expected real component tolerance can be enough to tip the circuit into self sustained oscillation with a zero to rated voltage ramped supply start-up.

Sometimes complex circuits - or simple circuits with complex models - may fail to simulate because the simulator cannot find the DC operating point prior to t=0. Such circuits will very often simulate fine if the supply (or supplies) are ramped up from 0. In the same way as slightly imbalancing symmetrical components can help start-up, slightly imbalancing the voltages, delay or rise times of the supply voltage ramps may help start-up more 'difficult' simulations.

There is however, a downside of using the start-up from zero supply ramp. As already stated, at the start of a transient simulation, the voltages across all capacitors and the currents through all inductors in the circuit will have reached their DC steady state values prior to the simulation starting at t=0. If the simulation starts with the supply voltages set to zero at t=0 then obviously all the internal voltages and currents must also be zero (except for some small offsets due to the initial levels of any signal sources and sometimes, badly designed sources internal to models that do not collapse to zero with zero supply voltages).

If all the internal voltages and currents are zero at t=0 then it may take much longer than the desired simulation stop time for all the internal nodes to reach their DC steady states.

A simple solution to this is to run the simulation for long enough for everything to have settled but to only plot the results for just long enough before the stop time to show the signals of interest. So, for example a circuit that is driven by a 1kHz source but which takes 95ms to settle could be observed from say, t=98ms to t=100ms by setting the transient analysis to have a stop time of 100ms and a Start time of 98ms, like this:

> **.tran 0 100m 98m**

This solution works very well but, in this example, over 95% of the simulation time is used just to get the circuit to a steady state before any useful results are generated. This is very wasteful of simulation time and for complex simulations such as switch mode power supply (SMPS) simulations - where exactly this situation is likely to arise - can take many minutes of real time.

This is where another technique for setting initial conditions in a circuit can sometimes be useful.

Setting initial voltages on nets and currents through components

There are occasions where it is required to start a simulation in some predetermined state. For example, a capacitor may be required to start a transient simulation at time t=0, precharged to some given voltage. Similarly the current in an inductor may need to be specified at time t=0. In a larger simulation it may be helpful to precharge the output smoothing capacitor of a power supply to approximately the right voltage to save the time taken for it to be charged up from zero. If the capacitor is at the output of an SMPS then it may be useful to charge the inductor(s) in the SMPS to their average operating current(s) too.

Using the `.ic` spice directive to set an initial voltage condition on a net

For more information about the .ic spice directive see:

> About-spice-analyses-in-EasyEDA

Use of the '.ic' spice directive is illustrated in these two examples:

> Setting initial circuit conditions 01

> Relaxation oscillator startup 02

This approach is not recommended but an alternative to using the .ic statement is shown in the these next examples show two ways to use append the 'uic' option to a Transient Analysis but for the reasons already given in the description of the 'uic' option in the section on 'IC: Set Initial Conditions' in 'Setting up Analyses', care should be taken in using this option.

> Relaxation oscillator startup 05

> Relaxation oscillator startup 06

Using a current source to set an initial current through an inductor

In LTspice, the `.ic` spice directive can be used to set initial current conditions through one or more inductors. This is illustrated in the following example:

> Setting initial circuit conditions 02

Setting the initial voltage on a capacitor

This approach is not recommended but an alternative to using the '.ic' statement to set the initial voltage across a capacitor is to append ic= to the capacitor value and uic to the .tran statement:

> Setting initial circuit conditions 03

Setting the initial current through an inductor

This approach is not recommended but an alternative to using the '.ic' statement to set the initial current through an inductor is to append ic= to the inductor value and uic to the .tran statement:

> Setting initial circuit conditions 04

Some circuits may start up on their own but simply changing a component model may cause it to fail to start-up. Don't be afraid to try these techniques before spending ages trying to find some other obscure cause.

Using a 1V source to help start-up

This is an advanced technique which has limited application but which in the right circumstances can be very useful.

By placing a 1V voltage source in a schematic, naming it's output as, say, 'unity' and then multiplying using expressions in B Sources by V(unity), it is possible to force all the B sources in the simulation to start from zero during the early DC operating point parts of a transient simulation.

This can sometimes help produce a clean startup from zero internal initial states.

It must be borne in mind however, that as with other techniques that bring a simulation up from zero, any attempt to run an OP, TF or an AC simulation will usually return zero results because all the internal states and in many cases the gains of B sources will have been forced to a zero initial value.

Replacing ideal and Thevenin voltage sources with band-limited Norton Sources to help start-up

This is technique is very widely applicable since it helps improve the overall convergence of simulations and not just their startup behaviour.

Ideal voltage sources in simulations are capable of producing infinite currents so a load which looks capacitive at any point during the simulation has the potential to cause an instantaneous infinite current flow. This can cause a simulation to go into overflow or to fail to find a valid next step from which to proceed. In either case, the simulation will fail to converge. To prevent this it is good practice to always include some resistance in series with every voltage source, whether that is an independent V, dependent E or H source or a dependent B source configured as a voltage source.

This turns all ideal voltage sources into Thevenin Sources, i.e. voltage sources with a finite source resistance.

Internally, however, the simulator is actually more efficient at calculating the currents and voltages around a Norton Source than a Thevenin source. So, the next step of the process is to convert all the Thevenin sources in a simulation into Norton Sources.

A Norton Source is the exact equivalent circuit of a Thevenin Source. To convert a Thevenin source to the Norton Equivalent source, the voltage source of the Thevenin source is replaced with a current source equal to the short circuit current of the Thevenin source and the series resistance of the Thevenin source is reconnected in parallel with this current source. Both sources than have identical source resistances, open circuit voltages and short circuit currents. So far these steps of converting ideal voltage sources to Thevenin sources and then to Norton equivalent sources generally increases the simulation speed of most simulations but the final step of limiting the bandwidth of the Norton sources can significantly improve the start-up and convergence of many simulations.

Limiting the bandwidth (or band-limiting) a Norton Source is easy: all that is required is to connect a capacitor in parallel with the source resistance that has already been connected in parallel with the current source. This creates a lowpass filter across the output of the Norton source which reduces the output at high frequencies. This means that even if the source is passing or generating highly non-linear signals that can produce exactly the kind of very fast and even discontinuous signals that spice struggles to deal with in its internal calculations, as the frequency content of these signals and transitions exceeds the lowpass filter cutoff frequency, they are reduced to relatively slower edged (i.e. reduced high-frequency content) signals. Such signals then have finite rise and fall times and so spice no longer struggles to deal with them. They become much more spice friendly continuous signals.

Even better is that if there are several such band-limited sources sprinkled around the circuit (such as the input signal sources) then very often the derivatives of these signals also become continuous. This is also a big help to spice because much of the internal calculation in spice is in estimating not just where a signal is but also making estimates about where it is going and how fast it is getting there.

Some words of warning are needed here though. Replacing Thevein voltage sources with band-limited Norton sources is a great technique but it requires a high degree of understanding of what is really going on in a simulation circuit.

The statement above about:

"...all that is required is to connect a capacitor in parallel with the source resistance that has already been connected in parallel with the current source. This creates a lowpass filter across the output of the Norton source which reduces the output at high frequencies."

is all very well but it skirts around the question of what should the value of this capacitance actually be? There is no single answer to that question because the cutoff frequency required to avoid convergence or start-up problems will vary from one circuit to another and even from one location to another in a circuit. The single biggest factor affecting the value is simply that the cutoff frequency must be much higher than the working frequencies of not the circuit itself but of the devices in it. So for example, an audio amplifier circuit designed for a bandwidth of 20kHz may use an opamp that is operated with a closed loop bandwidth that is 200kHz but which internally may have a gain bandwidth of 2MHz or even 20MHz.

In practice, setting the cutoff frequency with an RC time constant of 1ps (1e-12s) i.e. a cutoff frequency of approximately 160GHz, should be safe for most 'ordinary' simulations of amplifiers, linear and switch mode power supply circuits but for high frequency and RF circuits the band-limiting must be increased accordingly.

The value of the resistor in this parallel RC circuit can be also be very tricky to decide. For many lightly loaded voltage sources, a value of 1 Ω is OK and simplifies the calculation of the parallel capacitance to 1pF. In some sources, however, it may be necessary to use a higher (or lower value) of parallel resistance. This in turn will require the use of a proportionately smaller (or larger) capacitance for the same cutoff frequency. This will also require that the current value of the source is reduced to maintain the same open circuit voltage (calculated from V=I*R).

This band-limiting technique should generally not be applied around any existing current sources in a simulation circuit. Placing an RC circuit in parallel with a pure current source may do more damage to the operation of the circuit by reducing the in-band source impedance of the current source than the band-limiting avoids!

Although not an example of a start-up or initialisation problem, the Ideal and Thevenin to Norton Source conversion is demonstrated in steps (i), (ii) and (iii) of this simulation:

> Parameters, expressions, functions and B Sources

Using the 'OFF' option to help start-up

Some components, such as switches, bjts, jfets, MOSFETs and MESFETs have an 'OFF' option to specify the device to be in an initial OFF state.

Switches also have an 'ON' option to specify the device to be in an initial ON state.

This option can be very useful to ensure that for example, one side of a two transistor bistable or monostable circuit or an astable mutlivibrator is off, so avoiding the situation described above where both transistors are on in the initial state. Whilst this does not in itself guarantee that the circuit will start up from t=0 but it may simplify any other measures that have to be taken to ensure it does.

These states are simply invoked by appending the keyword OFF (or, for switches only, ON) after the device name. This can be done either by directly editing the name in place in the schematic or via the right hand panel Properties dialogue. For example, to set a switch with the name MYSWITCH to be initially ON the name should be edited to:

> **MYSWITCH ON**

Similarly, to set a bjt with the name 2N2222 to be initially OFF the name should be edited to:

> **2N2222 OFF**

Simulation eBook - Expressions

Expressions

Expressions can be used to define component values and to help configure Voltage and Current Sources.

Operators

In expressions, parentheses are evaluated before the other operators. The operators are evaluated following a list of precedence as shown in the table below. For equal precedence binary operators, evaluation goes left to right. Functions operate on real values only!

Operator	**Alias**	**Precedence**	**Description**
-		1	unary negate (see Note 1 below)
!		1	unary not
*		*2*	*power (but see also the pow(x,a), pwr(x,a) and pwrs(x,a) functions)*
/		3	multiply
/		3	divide
+		4	add
-		4	subtract
==		5	equality
<=		5	less or equal
>=		5	greater or equal
<		5	less than
>		5	greater than
&		6	boolean and
&#124		7	boolean or
^		7	boolean exclusive or
if(x,y,z)		8	ternary operator

The number zero is used to represent boolean False. Any other number represents boolean True. The result of logical operators is 1 or 0.

Some examples of logical operators used to defined value of voltage sources:

> **V1or 1 0 {1 | 0}**

> **V2and 2 0 {1 & 0}**

> **V3not 3 0 { ! 1}**

> **V4not 6 0 { ! 0}**

Note that when used directly in component and source value fields in a schematic, expressions MUST be on a single line. When used like this, expressions cannot be wrapped over more than one line.

Using Expressions to define component values

The -3dB frequency, fc, of a first order RC lowpass filter is given by:

> fc = 1/(2*pi*R*C)

Exactly the same expression applies to a first order RC highpass filter.

If fc is specified as 10kHz and R is chosen as 1k then:

> C = 1/(2*pi*1k*10k)

Suppose the high pass filter output is required to be attenuated by a factor, A.

The total value of R for the highpass filter is still 1k but it must be split into a lower resistor with a value given by:

> Rlower = (Rupper+Rlower)*1/A

whilst the upper value is given by:

> Rupper = (Rupper+Rlower)*(1-1/A)

If we choose A=3 then for the chosen value of R=1k

> Rlower = 1k*1/3

and:

> Rupper = 1k*2/3

Simply by entering the right hand side of these expressions into the component values fields, enclosed in curly brackets like this:

> `{expression}`

the value of those components will be defined directly by those expressions, as illustrated in Rupper and Rlower in this example.

Using Expressions to configure voltage and current sources

In this example, PULSE source V1 is configured to generate a signal with a 20us rise and fall time, a frequency of 5kHz and exactly equal high and low times: in other words, a slow edged squarewave of 200us period and a 50% duty cycle.

Because a PULSE source is defined in terms of Trise and Ton, it can be helpful to think of the time interval from the start of the rising (leading) edge to the start of the falling (trailing) edge as the 'pulse width', Twidth:

> Twidth = Trise+Ton

It is then a simple matter to define the PULSE source in terms of Trise and Twidth without having to manually calculate a value for Ton because:

> Ton = Twidth-Trise

From this we can also see that if the 'duty cycle' is defined as:

> D = Twidth/Tperiod

then for a given D:

> Ton = D*Tperiod - Trise

Lastly, it is sometimes convenient to define the period of a PULSE source in terms of a frequency:

> Frequency = 1/Tperiod

To use an expression in a source, simply enter it in place of the value you wish to calculate and enclose it in curly brackets like this:

> `{expression}`

The use of expressions is illustrated in the following example:

> Using expressions 01

Simulation eBook - Parameters

Parameters

Usually, the values of components are specified directly in the component's value field. There are, however, occasions where it is desirable to be able to set or change the value of several components at once without having to edit each individual component value.

The simple resistor attenuator circuit used to illustrate several of the early examples has one resistor of 1k and two of 2k. Instead of entering the value 1k into one resistor resistor 2k into each of the others, it is possible to set up two variables to represent these values.

To create two variables, R1val=1k and R2val=2k, a `.param` statement is placed into the schematic and turned into a spice directive (by doing: **Text Attributes > Text type = spice**):

> `.param R1val=1k R2val=2k`

The parameters are then used to define the values of the components in their value fields. By placing more than one .param statement in a schematic and doing:

> **Text Attributes > Text type = comment**

And

> **Text Attributes > Text type = spice**

It is possible to switch sets of values without having to edit individual components every time. Using .param statements therefore makes it possible to:

- change the value of several components a single edit;
- define parameters in terms of other parameters;
- define parameters in terms of functions of other parameters.

Note that it is also possible to have several .param statements active in a simulation at the same time but to avoid the conflict caused by duplicate definitions, the parameter identifier names in each set must be unique.

The syntax of .param statements is:

```
`.param <param_name1>=<value1> <param_name2>=<value2> ... <param_nameN>=
<valueN>`
```

'.param' statements can wrap over more than one line by using the '+' continuation character:

```
.param
+ <param_name>=<value1>
+ <param_name2>=<value2>
+ ...
+ ...
+ ...
+ <param_nameN>=<valueN>
```

Parameters can be numbers, other defined parameters or expressions made up from any combination of numbers and defined parameters.

Parameter identifier names must begin with an alphabetic character. The other characters must be either alphabetic, a number, or;

```
! # $ % > [ ] _
```

as special characters.

Note that when used in .param statements, expressions can wrap over more than one line by using the '+' continuation character:

```
.param
+ <param_name>=<value1>
+ <param_name2>=<value2>
+ <param_name3>=<{expression1}>
+ <param_name4>=<{part of expression2
+ continuation of expression2}>
+ ...
+ <param_nameN>=<valueN>
```

Care must be taken in choosing the line break points to clearly distinguish the use of the '+' character as a continuation character from any mathematical use of the '+' character as an addition operator in an expression.

For example:

```
.param x=3 y=4
+ hypotenuse={sqrt(x^2+
+ y^2)}
```

and:

```
.param x=3 y=4
+ hypotenuse={sqrt(x^2 +
+y^2)}
```

are valid wrappings of expression in a .param statement which will give the expected results, whereas:

```
.param x=3 y=4
+ hypotenuse={sqrt(x^2
+ y^2)}
```

or:

```
.param x=3 y=4
+ hypotenuse={sqrt(x^2+
y^2)}
```

or:

```
.param x=3 y=4
+ hypotenuse={sqrt(x^2
+y^2)}
```

may give unexpected results or may fail with errors.

- Note that the variables TIME and TEMP are NOT valid identifier names because in LTspice, they are reserved names.

Note that to use a parameter in a component value field, it must be enclosed in curly brackets:

> `{...}`

A parameter should also be enclosed in curly brackets if it is being used to define the value of another parameter, as in the example here of:

> `R3val={R2val}`

Although not shown here, the use of curly brackets to enclose expressions containing parameters which are then used to define other parameters is mandatory so, even though the use of curly brackets in the example shown above is not mandatory, it is good practice to always enclose in curly brackets any parameter or expression used in a parameter definition.

- Note, however, that parameters in expressions used in B Sources should not be enclosed in curly brackets.

The basic use of parameters is illustrated below:

> Using parameters 01

Using parameters in expressions

Expressions and parameters can be combined to simplify and automate the calculation of component and source configuration values, as illustrated in the following example:

> Using parameters in expressions 01

Functions

The sections on expressions and parameters shows how component and source values can be defined using arithmetic equations. The examples used to illustrate this used only simple linear expressions. This section introduces the concept of functions.

Functions hugely expand the power of parameters and expressions by allowing the creation of expressions including non-linear functions of other parameters.

Predefined functions

EasyEDA has a number of pre-defined functions. With the exception of the softlim(ip, lo, hi, sharp) function which must be copied and pasted by hand, all are immediately available to be used in expressions because they are built-in to LTspice.

For a complete list of functions and information on their usage, please see Help in a locally installed copy of LTspiceXVII or:

> B. Arbitrary Behavioral Voltage or Current Sources

> B sources (complete reference)

and:

> B sources (common examples)

Some of the most frequently used functions are listed, together with illustrative examples, in the table below.

Note that all the functions in this list can be used in any context in EasyEDA: in the value fields and in expressions for component values (E), Independent Sources (I) and for B Sources (B).

####**Table of functions**

Function	**Description**	**LTspice native or EasyEDA special**	**Where useable**
abs(x)	Absolute value of x	LTspice	E, I, B
acos(x) arccos(x)	Arc cosine of x.	LTspice	E, I, B
acosh(x)	Real part of the arc hyperbolic cosine of x, e.g., acosh(.5) returns 0, not 1.0472i	LTspice	E, I, B
asin(x) arcsin(x)	Arc sine of x.	LTspice	E, I, B
asinh(x)	Arc hyperbolic sine	LTspice	E, I, B
atan(x) arctan(x)	Arc tangent of x	LTspice	E, I, B
atan2(y,x)	4 quadrant arc tangent of x/y (tan^-1(x/y))	LTspice	E, I, B
atanh(x)	Arc hyperbolic tangent.	LTspice	E, I, B
buf(x)	Returns 1 if x > 0.5, else 0	LTspice	E, I, B
ceil(x)	Integer equal or greater than x	LTspice	E, I, B
cos(x)	Cosine of x	LTspice	E, I, B
cosh(x)	Hyperbolic cosine of x	LTspice	E, I, B
exp(x)	e to the power of x	LTspice	E, I, B
floor(x)	Integer equal to or less than x	LTspice	E, I, B
if(x,y,z)	IF x > 0.5, THEN y ELSE z	LTspice	E, I, B

placeholder	Placeholder for other function	LTspice	E, I, B
int(x)	Convert x to integer	LTspice	E, I, B
inv(x)	Returns 0 if x > 0.5, else 1	LTspice	E, I, B
limit(x, L, U)	Value of x, bounded by L and U	LTspice	E, I, B
ln(x) log(x)	Natural logarithm of x.	LTspice	E, I, B
placeholder	Placeholder for other function	LTspice	E, I, B
log10(x)	Base 10 logarithm. Generates a real valued output for all x.	LTspice	E, I, B
max(x,y)	The greater of x or y	LTspice	E, I, B
min(x,y)	The smaller of x or y	LTspice	E, I, B
pow(x,a)	Real part of x raised to the power of a. Zero for negative x and fractional a	LTspice	E, I, B
pwr(x,a)	The absolute value of x raised to the power of a	LTspice	E, I, B
pwrs(x,a)	pwr(x) multiplied by the sign of x	LTspice	E, I, B
sgn(x)	Sign of x. Returns -1 for x < 0, 0 for x == 0 (where == means 'exactly equal to') and 1 for x > 0	LTspice	E, I, B
sin(x)	Sine of x	LTspice	E, I, B
sinh(x)	Hyperbolic sine of x	LTspice	E, I, B
softlim(ip, lo, hi, sharp)	Value of ip, bounded by lo and hi with the sharpness of the transition between linear and limited regions defined by 'sharp'.	EasyEDA	E, I, B
sqrt(x) |Real part of the square root of x. Zero for negative x |LTspice |E, I, B ||tan(x) |Tangent of x |LTspice |E, I, B
tanh(x)	Hyperbolic tangent of x	LTspice	E, I, B
u(x)	Unit step, i.e., 1 if x > 0., else 0	LTspice	E, I, B
placeholder	Placeholder for other function	LTspice	E, I, B
uramp(x)	x if x > 0, else 0	LTspice	E, I, B

User defined functions

There may be occasions where a function is required that maybe has to be used in several places in a schematic or it is useful in several different schematics. To save having to copy and paste a complicated expression as a block of text each time it is needed, the .func statement makes it is possible to create a user defined function.

The syntax of the .func statement is very simple:

> `.func myfunctionname(a,b,c, ...n) {expression of functions of a, b, c ... n}`

For example:

> `.func hypotenuse(x,y) {sqrt(x^2+y^2)}`

defines a function that calculates the length of the hypotenuse of a right angle triangle with sides length x and y.

Once a function has been defined in a schematic in a project in this way it can be used anywhere in that schematic in that project. It does, however, have to be defined in every sheet in every project in which it is to be used but, if it's a really useful function, let us know and maybe we'll add it to the growing list of pre-defined functions!

To use the function all that is then required is to paste `hypotenuse(x,y)` into wherever it is needed and to substitute the 'x' and 'y' with the required variables. So, for example to use the function in a parametric expression:

> `.param a=3 b=4 hypot=hypotenuse(a,b)`

or in a current output B Source driven by two voltages, V(oneside) and V(otherside):

> `I=hypotenuse(V(oneside), V(otherside))`

There are many examples of functions defined by the .func statement in the simulations linked to in the table above and in all EasyEDA spice netlists for the automatically appended predefined functions.

Note that when used in .func statements, expressions can wrap over more than one line by using the '+' continuation character.

The softlim(ip, lo, hi, sharp) function is an example of this in the table above:

```
.func softlim(ip, lo, hi, sharp)
+ {uramp(((u(ip/2+0.125/(max(abs(sharp),1)*0.5/(hi-lo))-hi/2))*
(max(abs(sharp),1)*0.5/(hi-lo))*
+ -1*(uramp(0.25/(max(abs(sharp),1)*0.5/(hi-lo))+hi-ip)**2) +
+ (1-u(ip/2+0.125/(max(abs(sharp),1)*0.5/(hi-lo))-hi/2))*(ip-hi)+hi)-(hi+lo)/2)
-
+ uramp(-1*((1-u(ip/2-0.125/(max(abs(sharp),1)*0.5/(hi-lo))-lo/2))*
(max(abs(sharp),1)*0.5/(hi-lo))*
+ uramp(ip+0.25/(max(abs(sharp),1)*0.5/(hi-lo))-lo)**2 +
+ u(ip/2-0.125/(max(abs(sharp),1)*0.5/(hi-lo))-lo/2)*(ip-lo)+lo)+(hi+lo)/2) +
(hi+lo)/2}
```

Simulation eBook - B sources

B sources

B sources are one of the most powerful components in EasyEDA. They are available as a BV voltage source and as BI current source (although in fact at the spice netlist level they are the same device just defined to have a voltage or a current output).

The function of every B source is defined by an equation.

The left hand side of the equation defines whether the output of the source is a voltage or a current.

The right hand side is an expression made up from numbers, the basic arithmetic operators and functions not only of parameters but, crucially, of dynamic voltages and currents from within the circuit being simulated. In other words, B sources can perform a range of functions in simulations that is limited only by the imagination of the simulation designer.

The syntax of the equations to define a BV source in EasyEDA is very simple:

> **V=expression**

For example:

> **V=3*V(a,b)**

defines a BV source that generates an output voltage equal to 3 times the difference between the voltage on the 'a' net (V(a)) and the voltage on the 'b' net (V(b)).

> **V=scale*uramp(V(a,b))/ABS(I(Vimon))**

defines a BV source that generates an output voltage equal to the parameter, scale, multiplied by the positive difference between the voltage on the 'a' net (V(a)) and the voltage on the 'b' net (V(b)), divided by the absolute value of the current through the 0V source Vimon (I(Vimon)).

> **V=Vswing*tanh(V(a,b)*Avol)**

defines a differential gain block with an small signal gain defined by a parameter, Avol, and an output voltage swing which is limited with a tanh function to +/- the value of another parameter, Vswing.

The syntax of the equations to define a BI source in EasyEDA is basically the same as for BV sources except that **V** for voltage is replaced by **I** for current:

> **I=expression**

> **I=V(a)*I(Vimon)**

defines a BI source that generates an output current equal to the voltage on the 'a' net (V(a)) multiplied by the current through the 0V source Vimon (I(Vimon)).

> **I=LIMIT(V(a), 3, minval**2)**

defines a BI source that generates an output current equal to the voltage on the 'a' net (V(a)) but clamped to the value of 3 and the square of the value of the minval parameter for all values of V(a) outside the range defined by 3 and minval**2.

> **I=V(a,b)/Rval**

when the '-' and '+' terminals of the B source are named 'a' and 'b' respectively then this expression defines a resistor of value Rval.

Note that curly brackets are not used in expressions for B Sources.

There are several examples of the uses of B Sources in the following simulations.

> B Sources 01

> limit(x, L, U)

> Parameters, expressions, functions and B Sources

Note that when entered directly in a B Source value field in a schematic, expressions MUST be on a single line. When used like this, they cannot be wrapped over more than one line.

Expressions entered into a netlist, however, such as inside a .subckt model definition, can wrap over more than one line by using the '+' continuation character. Several examples of this can be found by inspecting the netlists of circuits using some of the EasyEDA .subckt models. For example in the netlist of the opamp5pEE Parameterised 5 pin opamp model there are these B sources:

```
Bipbias1 inp isum I=(ibias+ios)*V(supply_ok) +
+ ( uramp(V(inn)-(V(vp)+inmax)) - uramp(-V(inn)+(V(vn)-inmin)) )/Rser
```

and:

```
Bstg1 0 stage1 I=Islew*tanh(V(indiff)*Kg)
+ - ( uramp(V(stage1)-(V(vp)-outhi)) - uramp(-V(stage1)+(V(vn)+outlo)) )/Rser
+ - sel*( uramp(V(stage1)-(V(out)+oooclmphi)) - uramp(-V(stage1)+(V(out)-
oooclmplo)) )/Rser
```

and in the opamp_ANF01 .subckt found elsewhere in this document, there is another example:

```
B1 out 0
+ V=(TANH((V(inp)-V(inn))*{Avol}*2/(V(vcc)-V(vee)))*(V(vcc)-V(vee))
+ +(V(vcc)+V(vee)))/2
```

- It is worth mentioning one special case of the syntax of expressions used in a B source.

If an expression in a B source contains an IF statement which itself contains an equals sign, `=`, like this:

```
B out 0 V=V(in)+IF(offset=1, 1, 0)
```

then this will throw an error of:

> `Unexpected equals sign`

To avoid this error the single equals sign `=` in the IF statement must be replaced with a double equals sign `==` likw this:

```
B out 0 V=V(in)+IF(offset==1, 1, 0)
```

For more information about B Sources please search **Help** in a locally installed copy of LTspice for **B. Arbitrary Behavioral Voltage or Current Sources** or go to:

> B. Arbitrary Behavioral Voltage or Current Sources

> B sources (complete reference)

and:

> B sources (common examples)

```



---

##  Simulation eBook - Device models
---

<a name="Device models"></a>
# Device models
In order to simulate the behaviour of the individual components, they have to be
described mathematically. The underlying equations that describe the behaviour
of a component are written into the simulator program (sometimes they can be
added by the user).
```

```
The equations that describe basic components such as resistors (I = V/R),
capacitors (I = C*dV/dt) and inductors (V = L*di/dt) may be reasonably
straightforward. The equations that describe diodes, bipolar (bjt) and a variety
of field effect (jfet) and MOS transistors become increasingly complex,
sometimes with several equations describing the behaviour of different aspects
of device performance in different regions of operation.

Because these sets of equations are very much based on the semiconductor physics
of devices and the manufacturing processes used to fabricate them, for some
families of devices, such as MOSFETs, different sets of equations may be used to
describe devices in the same family. The different sets of equations may be used
because the manufacturer wishes to describe the operation of their devices to a
greater or lesser degree of accuracy.

Although the equations themselves are hidden deep in the source code for a
simulator, in general the coefficients of the sets of equations are collected
together in the form of a list. Individual devices of any particular device
family can then be described by a list of coefficients.
This list of coefficients is called a **model**.

The individual coefficients in a model are called the **model parameters**.

A device model written in this way is called a **.model** statement.

Some devices such as Thyristors, opamps, linear regulators and switch mode
supply chips are made up from a number of other devices connected together to
form subcircuits.

A spice netlist of a device defined by a subcircuit is also referred to as a
**model**.

A device model written in this way is called a **.subckt**.

Subcircuit models may themselves contain .model statements.

Subcircuits can also contain parameters and can also have parameters passed to
them to change their characteristics for example to tailor them to a particular
device variant.


<a name="Why are there different models for the same device?"></a>
### Why are there different models for the same device?
Because each family of devices (resistors, diodes, bjts, jfets, MOSFETs etc.) is
described by one or more sets of equations, each family has one or more models
available for it.

One reason there are different models available for devices in the same family
is because manufacturers give device models away for free. Therefore they do not
want to spend any more time on developing device models than they need to.
Basically, the more complex a model is, the more time the manufacturer has to
spend on making measurements in order to derive the model parameters. Therefore
if one manufacturer feels that a device can be adequately described by a simple
model then they will use that rather than a more accurate but more complex and
so more expensive one that may be available from a different manufacturer.

Another reason there may be differences between models of the same device is
that there may be slight differences in the semiconductor fabrication processes
of different manufacturers.
```

```
In the same way that there may be more than one .model available for a device,
there may be different .subckt defined models available.

There may be differences between .subckt models because there are implementation
differences in the device models and/or the physical devices from different
manufacturers. For example there are slight differences in internal timings and
even a subtle difference in the internal circuitry of the oscillator section of
the UC384x family of SMPS controllers between the various different
manufacturers.

Sometimes, there are differences in the models just to get around the copyright
protection. Some differences are to optimise the model for a particular
simulator and some differences are simply down to the preferences of the model
writer.


<a name=".model statements"></a>
### .model statements
In the spice netlist of a circuit, the user can see the models listed in .model
statements. When a schematic is saved, these .model statements are pulled in to
the netlist by EasyEDA recognising the symbols and their associated device names
given in the schematic. Each model may either be pulled in from a library or -
for devices that are not in the EasyEDA libraries - by downloading a model from
a manufacturer's website and then manually pasting it directly into the
schematic (the process of doing this will be described later).


<a name="LTspice model types"></a>
#### LTspice model types
To help identify model types and in particular if they are for N or P type
devices, the following table of model types may be helpful.




|**Code**   |**Model Type** |
| --- | --- |
| R | Semiconductor resistor model |
| C | Semiconductor capacitor model |
| L | Inductor model |
| SW | Voltage controlled switch |
| CSW | Current controlled switch |
| URC | Uniform distributed RC model |
| LTRA | Lossy transmission line model |
| D | Diode model |
| NPN | NPN BJT model |
| PNP | PNP BJT model |
| NJF | N-channel JFET model |
| PJF | P-channel JFET model |
| NMOS | N-channel MOSFET model |
| PMOS | P-channel MOSFET model |
| NMF | N-channel MESFET model |
| PMF | P-channel MESFET model |


Although it is beyond the scope of this document to go into detail there are
some other points about models that are worth mentioning.
```

```
* Models for the basic resistors, capacitors and inductors used in a schematic
are usually not shown in the netlist;
* Some device models have a full list of parameters, some may only have a
partially completed list. Missing parameters in models are simply replaced by
default values.
* Different simulators support different sets of models so in some cases the
simulator may warn the user that some parameters are unrecognised and so are
ignored. This often has little effect on the simulation results but if the user
is particularly concerned about their effects then the only options are to find
a model which only uses parameters recognised by LTspice or change to using a
simulator that supports all the relevant parameters.


<a name=".subckt definitions"></a>
### .subckt definitions
Not all devices are described by .model statements.

Models of more complex devices such as Thyristors (SCRs, Triacs and also Diacs),
Insulated Gate Bipolar Transistors (IGBTs), operational amplifiers (opamps) and
even many MOSFETs are often made up by connecting lower level devices to make a
circuit that behaves like the desired device. This is called a **subcircuit**.
The spice netlist of this subcircuit is then used to create a type of device
model defined by what is called a **.subckt**. The low level components in
subcircuits are described by the same sort of models (those lists of parameters
or coefficients) as for the basic diodes etc., already referred to so a .subckt
will often contain a list of .model statements describing the devices that are
used to build the .subckt itself. Complex .subckts may even call other .subckts.


<a name="Behavioural models"></a>
### Behavioural models
Using Behavioural Voltage and Current Sources and expressions it is possible to
create what are called **behavioural models** of components. These are models
that behave like a device but which have little or none of the actual underlying
realistic circuit defined and are mostly - or perhaps completely - described by
explicitly defined expressions (equations). The models for most devices
internally comprising more than one active component, i.e. ICs, are largely
behavioural. This is a way of hiding the detailed information about the
manufacturer's process technology that low level spice modelling reveals.

The use of expressions and behavioural sources in EasyEDA is explained later in
the book.


<a name="What if there is no model available for a device?"></a>
### What if there is no model available for a device?
Not all devices have spice models that can be run in EasyEDA. There are a number
of possible reasons for this.

1) Some models are encrypted and can only be run in certain proprietary
simulation tools;

2) Some proprietary simulators support models that are not available in LTspice;

3) Some devices have models that only run in specific non-spice based simulation
tools and which, for whatever reason, cannot be translated into spice models;

4) Some devices do not have publically available models;
```

```
5) Many devices predating the creation of the original spice program do not have
models;

6) Models for some devices simply do not exist because the manufacturers have
never created them;

7) Some models may be unavailable in EasyEDA because they are restricted by
copyright or end user licenses so they can only be run in certain proprietary
simulation tools or cannot be shared publicly.

In cases (1) to (3), there is no way they can be run in LTspice. They must be
run in the simulation tools for which they were written.

In cases (4) to (7), it is sometimes possible to find an equivalent, alternative
or similar device for which a spice model is available. The user must exercise
caution and use their judgement in deciding if such an approach offers
satisfactory simulation results.

It must be noted that spice was not originally written with support for
thermionic devices (valves or tubes) so models for such devices exist only in
.subckt form. They are usually created by enthusiasts rather than manufacturers
and so they (a) can be hard to find and (b) should be used with caution. EasyEDA
does have a library of valve models gathered from sources that we believe have
written reasonably accurate models.

Note that models obtained from manufacturers are often subject to copyright
restrictions. Please respect any copyright notices contained either in end user
license agreements that may have to be accepted prior to the granting of access
to a downloadable copy of a model or in the models themselves.

Similarly, models contained in the libraries of commercial simulation tools are
subject to copyright restrictions.

It is often possible to find device models offered in forums, discussion groups
and various online collections of models. Again, the user must exercise caution
and use their judgement in deciding if such models really are suitable. Often it
is not possible to establish where they originate from so their validity is very
hard to verify. It is also possible that such models have been copied in breach
of the originators’ copyright.


<a name="The relationship between spice models and device datasheets"></a>
### The relationship between spice models and device datasheets
Although some of the device models in EasyEDA have been specially written so
that the user can easily tailor them to simulate a range of devices by editing
parameters that can be found directly in - or inferred from - device datasheets
(see: About the relationship between spice models and real world behaviour
below), most of them are off-the-shelf models from the device manufacturers.

It is important to understand that, for many of these off-the-shelf models, the
underlying equations and therefore the .model parameters and .subckt definitions
bear little relationship to the sort of information that is given in typical
component datasheets. Therefore it is usually not possible to take a device
datasheet and simply write down a device models from the information given in
it.
```

```
Whilst it is possible to extract spice parameters for a variety of devices from
device datasheets and from actual device measurements, it is beyond the scope of
this document to describe how this can be done.

More information about what the model parameters mean in diodes, bipolar
transistors and MOSFETs, is available from:

http://www3.imperial.ac.uk/pls/portallive/docs/1/56133736.PDF

with individual slide sets:

http://www3.imperial.ac.uk/pls/portallive/docs/1/7292571.PDF

http://www3.imperial.ac.uk/pls/portallive/docs/1/7292572.PDF

http://www3.imperial.ac.uk/pls/portallive/docs/1/7292573.PDF

For more detailed information about bjt's in particular, this book:

>[Modelling the Bipolar Transistor by Ian Getreu]
(http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01480193)

is available from:

http://www.lulu.com/spotlight/iangetreu

And:

http://www.amazon.com/Modeling-Bipolar-Transistor-Ian-Getreu/dp/B000EYPQLU

Another excellent (and free) book about transistor modelling, is available by
going to:

http://www.aeng.com/spice_modeling.htm

and registering to get a copy of:

>[Definitive Handbook of Transistor Modeling]
(http://www.aeng.com/pdf/RGVmaW5pdGl2ZSBIYW5kYm9vayBvZiBUcmFuc2lzdG9yIE1vZGVsaW5
n.pdf)

More information about LTspice is available from here:

http://ltwiki.org/index.php?title=Main_Page

More information about Larry Nagel and SPICE is available from here:

http://www.omega-enterprises.net/The%20Origins%20of%20SPICE.html

Larry's PhD dissertation Dissertation:

Laurence W. Nagel., "SPICE2: A Computer Program to Simulate Semiconductor
Circuits,"

Memorandum No. ERL-M520, University of California, Berkeley, May 1975.

http://www.eecs.berkeley.edu/Pubs/TechRpts/1975/ERL-520.pdf
```

```
is actually very readable and instructive.

For more information about electronic circuit simulation and spice in
particular, see:

http://en.wikipedia.org/wiki/Electronic_circuit_simulation

And:

http://en.wikipedia.org/wiki/SPICE


<a name="The relationship between spice models and real world behaviour"></a>
### The relationship between spice models and real world behaviour
Not all spice models are created equal.

Here are just some of the things to be aware of.

Models of the same device from different manufacturers may offer differing
degrees of accuracy. Sometimes models are kept simple in the interests of
speeding up the simulations at the expense of accuracy. Sometimes they are
complex because accuracy is considered to be more important than simulation
speed. Models may contain some text at the beginnings of them to describe some
of their limitations or their special features. It is often useful to read this
information as it can help improve the convergence of simulations using them.

Not all diode models simulate reverse breakdown voltage.

Zener diode models can be of varying accuracy and are best put into test jigs to
run a curve trace on them to compare them with the datasheet. Zener diodes are
sometimes used as white noise sources. Zener models do not accurately generate
the levels and spectrum of noise seen in real devices.

None of the bjt models simulate the reverse bias base-emitter breakdown voltage.
Very few model collector-emitter or collector-base junction breakdown voltages.

Some models, particularly of high speed and high frequency devices may include
package parasitics such as lead inductances and pin capacitances. Such models
are almost always .subckt definitions of devices defined by .model statements
but which have the parasitics connected to form a subckt. If the high frequency
behaviour is not important, simulation speed can be improved by using only the
.model statement without the parasitics. This .model can be cut and pasted out
of the .subckt definition but often the .model statement will be for a
transistor that is defined as a .model in its own right somewhere else on the
manufacturer’s site or as an equivalent from another vendor.

Thyristor and Triac models can be of varying accuracy or simulate only a limited
selection of all the device parameters.

EasyEDA has an in-house behavioural Thyristor macromodel and a behavioural Triac
macromodel.

As far as possible, the EasyEDA in-house Thyristor and Triac models model almost
all the datasheet parameters of the target device with the exception of di/dt
behaviour with inductive loads. These devices can be tailored to model almost
any device simply using the values taken from the datasheet for the target
device.
```

Metal Oxide Varistors (MOVs) are a nightmare to model and are best avoided! Even the commercially available models sometimes do not run reliably in all conditions.

Some opamp models are highly detailed and can be very accurate but care must be taken to check that they are written using a syntax that is compatible with LTspice. Devices tailored for some of the commercial simulators will not run in LTspice without some syntax changes. Some may require special **.options** to be invoked for the simulator.

Beware that even some quite complex opamp models do not simulate supply current drains even as simple DC quiescent currents let alone the dynamic behaviour with load currents added in. This can be an advantage since it reduces the signal currents that have to be simulated. It also means that there is absolutely no point in including any supply rail decoupling for those device that are known to not model supply current drains since they do not draw any current: they only use the supply voltage to define things like common mode range or output swing.

Here is an example of a third party opamp model that does not model supply or output currents:

>[LM108 test jig]
(https://easyeda.com/editor#mode=sim,id=0072ad0fd0ef4bca8e14ce8a378449fc)

Some opamp models may make no attempt to accurately simulate the output stage behaviour versus load current. Similarly, many device models do not simulate the behaviour of inputs and outputs when they are taken above or below the supply rails.

Few device models simulate the excessive supply current drain of a supply reversed misconnection or a correctly connected device that is subject to a supply voltage above the stated absolute maximum supply differential.

There are many device models in EasyEDA that have been specially written to reproduce the real world behaviours - such as described above - of the devices that they model.

For example, the EasyEDA in-house opamp behavioural macromodel can be set up to give an output voltage swing anywhere from a rail-to-rail to the more restricted swings of non-rail-to-rail output opamps. The output swing can be asymmetric.

Input resistance, bias and offset current and input offset voltage are modelled.

The input differential and common mode voltage ranges are modelled.

The current drain behaviour of the device if input or output pins are taken above or below the supply rails or if the supply polarities are reversed are modelled. Output polarity reversal due to inputs exceeding the common mode range is modelled for devices that exhibit such behaviour.

Frequency dependent common mode and power supply rejection are modelled.
Noise and temperature dependent effects are not modelled at present.

EasyEDA has an in-house behavioural macromodel which can be tailored to model a wide range of 3 terminal fixed and adjustable positive and negative linear voltage regulators which feature similar real-world behaviour to the opamp models.

```
For some of the in-house EasyEDA models, more information about them can be
found in the .subckt definition itself simply by viewing the spice netlist of
any saved circuit they have been put into.


<a name="How to change the model attached to a symbol"></a>
### How to change the model attached to a symbol
**Please note that before attempting to edit device models, it is essential that
the user is familiar with and understands the relationship between spice pin
names and numbering, described in the section on `Schematic symbols: prefixes
and pin numbers`.**

There are a couple of ways to change the model for a device.

1) Place a device from the EasyEDA Libs and then edit the device model name
either in place in the schematic or in the right hand properties panel.

>For instance, when an NPN bjt is placed in a schematic, it comes in with a
default name of editing the model name of 2DC2412R. This name pulls the
associated default 2DC2412R model into the spice netlist. Editing the device
name from 2DC2412R to 2N2222 will pull the 2N2222 model from EasyEDA's spice
model library into the netlist.
The problem here is that until a model search function is up and running this
approach is obviously too hit and miss for an arbitrary choice because there is
no way to see which models are available to choose from.

2) The second option is a bit more fiddly but it allows almost any unencrypted
device model to be run in a simulation. The process is similar for both .model
and .subckt defined models.


<a name="For .MODEL defined models"></a>
#### For .MODEL defined models
1) Find a spice .model for your target device;

2) Copy and paste it into a text placeholder (the T hotkey) in your schematic
(but please respect the EULA and copyright of commercial files);

3) In the right hand properties panel, change the text type from comment to
spice;

**Properties > Text type > spice**

4) Place a symbol for the device from the EasyEDA Libs palette onto the
schematic;

5) Edit the model name to the exact name of the model in the pasted file.

6) Done!

There is an example of this here:

>[Playing with model parameters]
(https://easyeda.com/editor#mode=sim,id=8b454f0e409e4ebf8c3f0e96b458d1e0)

This is another example showing using a generic depletion mode MOSFET.
```

```
It also shows a way to hack a MOSFET defined by a LEVEL 3 .model statement but
which has a problem with some of the parameters not being recognised as being
part of the model by LTspice, so that it can still be used directly with the
MOSFET symbol.

In this example the L and W parameters of the original model are recognised as
part of the .model statement. Note also that some of the other parameters are
also simply not recognised by LTspice.

Here's how:

1) Find a spice .model for your target device;

2) Copy and paste the .model statement into the schematic canvas;

3) Turn it into a spice directive:

    >**Text Attributes > Text type > spice**

4) Place an N channel depletion mode MOSFET symbol onto the schematic;

5) Edit the 'model' attribute for M1 to include the unrecognised or modified L
and W parameters so they look like this:


>**IXTT20N50D L=2E-6 W=5.5**

>This can be done either in place or via:

>**Part Attributes > Model > IXTT20N50D L=2E-6 W=5.5**

Note that adding an asterisk at the start of the two lines in the .model
statement that define the L and W (and any other unrecognised parameters as
deemed necessary) parameters will comment them out. This stops these parameters
being reported as model issues in the simulation report but it is not required
to do so.

This process is illustrated in the following example:

>[N channel depletion mode MOSFET using a .model statement]
(https://easyeda.com/editor#mode=sim,id=07a2743a497647419fc68f253e84c295)



<a name="For .SUBCKT defined models"></a>
#### For .SUBCKT defined models
The process described above works fine for simple .model defined models but for
.subckt defined models it is a little more complicated because you need to tell
EasyEDA that the model is a .subckt and not a simple .model.

Even some humble diode models are in fact .subckt defined to include things like
package parasitics. For example, compare the 1N4148 and the 1N4148W-V models in
the netlist.

There are three stages in attaching a .subckt to a symbol that already has a
spice prefix of 'X' and so is expecting to call a .subckt statement.

* Place the .subckt text into the schematic and activate it;
* Place the symbol in the schematic;
* Change the name of the symbol to exactly the same as the name of the .subckt;

The detailed steps to associate a new .subckt model to the symbol are:

1) Find a spice .SUBCKT for your target device;

2) Copy and paste it into a text box (the T hotkey) in your schematic (but
please respect the EULA and copyright of commercial files);

3) In the right hand properties panel, change the text type from comment to
spice;


**Text Attributes > Text type > spice**

4) Place a symbol for the device from the EasyEDA Libs palette onto the
schematic;

5) Edit the model name to the exact name of the model in the pasted file;

6) Press the 'I' Hotkey or:

Click the 'Edit Symbol...' button in the Properties panel:

**Part Attributes > Edit Symbol...**

or do:

**Right-click on the symbol > Edit Symbol**

7) In the 'Modify your symbol information' dialogue box, check that the 'Spice
Prefix' is 'X';

8) Check that the NUMBER of pins in 'Edit Pin Map information' is exactly the
same as in the .SUBCKT pasted into the schematic: if it is not then the wrong
symbol has been placed for the chosen .SUBCKT (or vice versa) so a different
symbol (or .SUBCKT) must be chosen.

Note that 'number of pins' here means how many pins, not the pin numbers or
names used to describe the nets they connect to in the .subckt netlist;

9) Check that the ORDER of the pins in 'Edit Pin Map information' is exactly the
same as in the .SUBCKT pasted into the schematic. This can be very confusing
because the pin NAMES may be different between the symbol and the .SUBCKT so it
is first necessary to reconcile the two sets of names before attempting to
confirm their order.

10) Click OK in the 'Modify your symbol information' dialogue box;

11) Done!


This process is illustrated in the following example:

>[Attaching a .subckt to a symbol 01]
(https://easyeda.com/editor#mode=sim,id=38bb9615b9ae4194825f924025d312ca)
```

```
Some of the EasyEDA symbols such as bjts and all the MOSFETs have a Spice Prefix
of 'M' and so are expecting to call a .model statement. To associate a .subckt
to a symbol with a Spice Prefix of 'M' there are four stages.

So, in the following example, the NMOS_E symbol placed into the schematic from
the EasyEDA Libs palette must be edited to change the 'Spice Prefix' of the
symbol from 'M' (for a .model defined part) to 'X' (for a .subckt defined part).

* Place the .subckt text into the schematic and activate it;
* Place the symbol in the schematic;
* Change the name of the symbol to exactly the same as the name of the .subckt;
* Change the 'Spice Prefix' of the symbol from 'M' (for a .model defined part)
to 'X' (for a .subckt defined part).

The detailed steps to associate a new model to a symbol and to tell EasyEDA that
a device model is a .subckt and not a simple .model are:

1) Find a spice .SUBCKT for your target device;

2) Copy and paste it into a text box (the T hotkey) in your schematic (but
please respect the EULA and copyright of commercial files);

3) In the right hand properties panel, change the text type from comment to
spice;

**Text Attributes > Text type > spice**

4) Place a symbol for the device from the EasyEDA Libs palette onto the
schematic;

5) Edit the model name to the exact name of the model in the pasted file;

6) Press the 'I' Hotkey or:

Click the 'Edit Symbol...' button in the Properties panel:

**Part Attributes > Edit Symbol...**

or do:

**Right-click on the symbol > Edit Symbol**

7) In the 'Modify your symbol information' dialogue box, change the 'Spice
Prefix' from 'M' (for a .model defined part) to 'X' (for a .subckt defined
part);

8) Check that the NUMBER of pins in 'Edit Pin Map information' is exactly the
same as in the .SUBCKT pasted into the schematic: if it is not then the wrong
symbol has been placed for the chosen .SUBCKT (or vice versa) so a different
symbol (or .SUBCKT) must be chosen.

Note that 'number of pins' here means how many pins, not the pin numbers or
names used to describe the nets they connect to in the .subckt netlist;
```

```
9) Check that the ORDER of the pins in 'Edit Pin Map information' is exactly the
same as in the .SUBCKT pasted into the schematic. This can be very confusing
because the pin NAMES may be different between the symbol and the .SUBCKT so it
is first necessary to reconcile the two sets of names before attempting to
confirm their order.

10) Click OK in the 'Modify your symbol information' dialogue box;

11) Done!


This process is illustrated in the following example:


>[Attaching a .subckt to a symbol 02]
(https://easyeda.com/editor#mode=sim,id=38bb9615b9ae4194825f924025d312ca)

Another example of the process described above to change the Spice Prefix of a
symbol is illustrated with the same EasyEDA N channel depletion mode MOSFET
symbol from the EasyEDA Libs that was used earlier with the IXTT20N50D .model
statement. In this example the MOSFET symbol is attached to a .subckt that has
been created from the orignal IXTT20N50D .model statement in order to wrap up
the L=2E-6 W=5.5 parameters and so make using the original model easier.

>[An N-channel depletion mode MOSFET using an EasyEDA .subckt]
(https://easyeda.com/editor#mode=sim,id=97747a105ae44816b7a29e73bebc29da)


<a name="Attaching models to custom symbols"></a>
#### Attaching models to custom symbols
This is basically the same as attaching a model to any of the predefined symbols
from the EasyEDA Libs except that the symbol is one that has been created from
scratch or by editing an existing symbol. The rules for assigning and checking
that the spice prefix matches the type of model to be attached ('M' for .model
or 'X' for .subckt) and checking that the spice pin numbering matches that of
the type of device defined by the .model statement or by the pin sequence of a
.subckt defined model.

---

##  Simulation eBook - Schematic symbols prefixes and pin numbers
---


<a name="Schematic symbols: prefixes and pin numbers"></a>
# Schematic symbols: prefixes and pin numbers
**Please note that before attempting to edit device models, it is *essential*
that the user is familiar with and understands the relationship between spice
pin names and numbering, described in this section.**

Device and subcircuit symbols created for use in schematics that are intended to
be run as spice simulations, in addition to having a PCB Prefix that is used for
the reference designator in the schematic, also have a Spice Prefix. They also
have two sets of pin numbers: PCB pins and Spice pins.


<a name="PCB and Spice Prefix"></a>
### PCB and Spice Prefix
```

```
The rules on the assignment of the PCB Prefix or reference designator of a
schematic symbol are somewhat dependent on the EDA tool and on the user's
preferences. Depending on how a device is graphically represented by it's
schematic symbol it may have a different PCB Prefix or reference designator. For
example, a single discrete MOSFET device may have a PCB Prefix of Q, M or
perhaps TR, whereas if it is part of a monolithic multiple transistor array it
may have a PCB Prefix of U or IC.

The rules on the assignment of the Spice Prefix of a schematic symbol are
strict. This is because the Spice Prefix is used to tell the simulator which
circuit element the symbol represents and therefore which simulation model it is
to use.

Simulation models for most of the spice circuit elements are in the form of a
single-line **.model** statement however some of them may be in the form of a
multi-line **.subckt** subcircuit definition.

For example, some MOSFETs may be described by a .model statement in which case
their Spice Prefix is M but many MOSFETs are described by a .subckt and so their
Spice Prefix is X.

Therefore, irrespective of the PCB Prefix chosen for a schematic symbol, the
Spice Prefix for a schematic symbol representing a given circuit element must
match the type of model required to simulate that instance of that circuit
element in your schematic.

For example, if there are two different n-channel MOSFETs in a schematic;

Q1, a BSS123 which is modelled by a .model statement:
```

*BSS123
*SRC=BSS123 ;DI_BSS123 ;MOSFETs N; Enh ;100V 0.170A 1.00ohms
*Diodes Inc. MOSFET
.MODEL DI_BSS123 NMOS(LEVEL=1 VTO=1.00 KP=6.37m GAMMA=1.24

- PHI=.75 LAMBDA=625u RD=0.140 RS=0.140
- IS=85.0f PB=0.800 MJ=0.460 CBD=19.8p
- CBS=23.7p CGSO=36.0n CGDO=30.0n CGBO=124n

- -- Assumes default L=100U W=100U --

```
and Q2, a BSS127S which is modelled by a .subckt:
```

- BSS127S
  *---------- BSS127S Spice Model ----------
  .SUBCKT BSS127S 10 20 30
- TERMINALS: D G S
  M1 1 2 3 3 NMOS L = 1E-006 W = 1E-006
  RD 10 1 84.22
  RS 30 3 0.001
  RG 20 2 29
  CGS 2 3 1.958E-011
  EGD 12 0 2 1 1

VFB 14 0 0
FFB 2 1 VFB 1
CGD 13 14 2E-011
R1 13 0 1
D1 12 13 DLIM
DDG 15 14 DCGD
R2 12 15 1
D2 15 0 DLIM
DSD 3 10 DSUB
.MODEL NMOS NMOS LEVEL = 3 VMAX = 8E+005 ETA = 1E-012 VTO = 3.419

- TOX = 6E-008 NSUB = 1E+016 KP = 0.127 U0 = 400 KAPPA = 1.044E-015
  .MODEL DCGD D CJO = 1.135E-011 VJ = 0.9232 M = 0.9816
  .MODEL DSUB D IS = 2.294E-010 N = 1.601 RS = 0.1079 BV = 65
- CJO = 1.956E-011 VJ = 1.514 M = 0.8171
  .MODEL DLIM D IS = 0.0001
  .ENDS
  *Diodes BSS127S Spice Model v1.0 Last Revised 2012/6/6

```
then even though both have the same PCB Prefix of Q: Q1 must have a Spice Prefix of M and Q2 must have a Spice Prefix of X.

A list of Spice Prefixes and their associated circuit elements is given in the table below.


| **Element description** | **Spice Prefix** | **Spice pin orde** | **Comment** |
| --- | --- | --- | --- |
| A | XSPICE code model | Depends on model | analogue, digital, mixed signal |
| B | Behavioural (arbitrary) source | Source+, Source-,<br>Control+, <br>Control- |  |
| C | Capacitor | Fixed by model but has no polarity |  |
| D | Diode | A K |  |
| E | Voltage-controlled voltage source (VCVS) | Source+, Source-,<br>Control+, <br>Control- | linear, non-linear |
| F | Current-controlled current source (CCCS) | Source+, Source-,<br>Vsrcname | linear<br><br>Vsrcname is for a voltage source external to the switch. |
| G | Voltage-controlled current source (VCCS) | Source+, Source-,<br>Control+, <br>Control- | linear, non-linear |
| H | Current-controlled voltage source (CCVS) | Source+, Source-,<br>Vsrcname | linear<br><br>Vsrcname is for a voltage source external to the switch. |
| I | Current source | Source+, Source- |  |
| J | Junction field effect transistor (JFET) | D G S |  |
| K | Coupled (Mutual) Inductors | Fixed but must respect winding phases |  |
| L | Inductor | Fixed by model but has no polarity |  |
| M | Metal oxide field effect transistor (MOSFET) | D G S |  |
| N | Numerical device for GSS |  |  |
| O | Lossy transmission line | Fixed by model |  |
| P | Coupled multiconductor line (CPL) | Fixed by model |  |
| Q | Bipolar junction transistor (BJT) | C B E |  |
| R | Resistor | Fixed by model but has no polarity |  |
| S | Switch (voltage-controlled) | Switch+, Switch-,<br>Control+,<br>Control- |
|
| T | Lossless transmission line | Fixed by model |  |
| U | Uniformly distributed RC line | Fixed by model |  |
| V | Voltage source | Source+, Source- |  |
| W | Switch (current-controlled) | Switch+, Switch-,<br>Vsrcname | Vsrcname is
for a voltage source external to the switch. |
| X | Subcircuit | Depends on subckt |  |
| Y | Single lossy transmission line (TXL) | Fixed by model |  |
| Z | Metal semiconductor field effect transistor (MESFET) | D G S |  |



For more information on circuit elements in LTspice, please refer to:

http://ltwiki.org/index.php?title=Main_Page


<a name="PCB and Spice pin numbers"></a>
### PCB and Spice pin numbers
The two sets of pin numbers are:

**PCB pin number**: these are the numbers for the real, physical device pins in
its package. They are required so that the pins of a device symbol in a
schematic can be mapped onto the physical pins of a PCB footprint. In other
words, so that the connections shown in the schematic, end up connected properly
by copper on the PCB.

**Spice pin number** or pin order: these are the numbers that map the pins on
the symbol to their respective functions in the spice model or subcircuit.

* The spice pin ordering of .model defined models is fixed as shown in the table
above.

* The spice pin ordering of .subckt defined models is determined by the .subckt
line in the subcircuit.

Actually the spice pin ordering has a slightly deeper meaning.

Spice has no concept of component symbols: they are a construct of the schematic
editor.

When a spice netlist is generated, the symbol in the schematic editor is either
- in the case of model defined devices such as resistors, capacitors, inductors,
diodes, transistors and sources - mapped directly to the relevant models
(defined by the device prefix such as R, C, L, D, Q and so on), or in the case
of a subcircuit, converted into a subcircuit call statement.

The spice pin ordering for the majority of built-in models such as resistors,
capacitors, inductors, diodes, transistors and sources are defined and generally
taken care of by the schematic editor, more care has to be taken with the spice
pin ordering of subcircuits.

This can be illustrated by a simple opamp with 5 pins: inverting and non-
inverting inputs; output and positive and negative supply pins but the principle
applies to all spice subcircuits.

The subcircuit call for this opamp might look like this in the spice netlist:

>**X1 input feedback vpos vneg output opamp_ANF01**
```

```
Where:

>**X1** is the name of the subcircuit in the top level (i.e. the calling)
circuit;

>**input feedback vpos vneg output** are the netnames in the circuit calling
(i.e. containing) the subcircuit and:

>**opamp_ANF01** is the name of the subcircuit being called.

* Pay special attention to the order of the netnames in the subcircuit call.

The spice pin ordering for the majority of opamp subcircuits is like that shown
in the example below:
```

*--- *

- opamp_ANF01 * * Simplified behavioural opamp *
- Node assignments
- noninverting input
- | inverting input
- | | positive supply
- | | | negative supply
- | | | | output
- | | | | |
- spice pin order: 1 2 3 4 5
- | | | | |
  .subckt opamp_ANF01 inp inn vcc vee out ; these are the netnames
- used internally to the
- subcircuit.
  B1 out 0 + V=(TANH((V(inp)-V(inn))*{Avol}*2/(V(vcc)-V(vee)))*(V(vcc)-V(vee)) + + (V(vcc)+V(vee)))/2
- .ends opamp_ANF01
- *---

```
Note that the spice pin order of the subcircuit call is in exactly the same
order as that of the pins in the .subckt line of the subcircuit.

Although the physical pin numbering of any device is critical for successfully
mapping the pins on a schematic symbol onto a physical package footprint when
laying out the PCB, because spice only knows about single devices and does not
care about how they are physically packaged, each instance of any device in a
spice schematic has to be mapped onto its own copy of the spice model or
subcircuit, irrespective of where it is in any physical package.

Therefore, for the physical, package pin numbering of the four opamps in a quad
opamp in say, a SOIC14 or a DIP14 package, as shown below, to work with the
example subcircuit above, the spice pin ordering would be:



| **Opamp A** | **Pin Number** | **Spice Pin Order** |
| --- | --- | --- |
```

OUT	1	5
IN-	2	2
IN+	3	1
V+	4	3
V-	11	4
Opamp B	**Pin Number**	**Spice Pin Order**
OUT	7	5
IN-	6	2
IN+	5	1
V+	4	3
V-	11	4
Opamp C	**Pin Number**	**Spice Pin Order**
OUT	8	5
IN-	9	2
IN+	10	1
V+	4	3
V-	11	4
Opamp D	**Pin Number**	**Spice Pin Order**
OUT	14	5
IN-	13	2
IN+	12	1
V+	4	3
V-	11	4

The physical package pin numbering reflects that of each opamp in the package.

The spice pin ordering is the same for each instance of the individual opamps.

Of course there is only one physical instance of each supply pin on the schematic symbol for the quad opamp in this example but each spice subcircuit must have the supply pins explicitly defined.

Exactly how this is handled is at the schematic symbol level depends on how the schematic capture package handles symbols for multiple devices with shared supply pins but the generation of a spice netlist from the schematic will always generate the complete set of pins required in the subcircuit calls.

In cases where the subcircuit is built by the user as opposed to where it is supplied by a vendor for a particular device, exactly the same rules apply except that it is up to the user to specify the subcircuit pin order and to construct the symbol appropriately.

Although as described earlier, built-in spice models usually have defined spice pin orders, not all subcircuits have the same spice pin numbering. Therefore if your spice circuit throws errors - especially if there are warnings about pin numbers or pin names - it is worth remembering to check that the pin order of the symbol that is netlisted to form the calling statement matches that of the subcircuit that is being called!

## Simulation eBook - Custom modelling

```
---

<a name="Custom modelling"></a>
# Custom modelling
This is an advanced topic and this section will be filled out when time permits.

Learning to write good quality models is not easy.

Before trying to make models, it is first necessary to have a deep understanding
of electronics in order to understand the devices that are desired to be
modelled.

Unlike creating a commercial model which has to be good at modelling all the
device performances, in custom modelling it is often useful to be able to select
what is modelled and what is not as this can help speed up simulations and
improve convergence. Therefore it is necessary to understand what parameters and
behaviours are important and what can be simplified or even safely ignored. This
is because some aspects of a device behaviour may not be important to model in
some applications whereas they may be critical in others.

Then, to make a model it is necessary to have a deep understanding of how to use
and edit the parameters of spice basic device models such as diodes transistors
and - in LTspice - some of the clever resistor, capacitor, diode and switch
models.

It is also necessary to have a very deep understanding of how to use behavioural
sources, expressions and functions to model internal and even whole device
behaviours.

The phrase "deep understanding" has been used three times already but that is
because it is easy to make bad models but it is much harder to make good ones!

However, browsing through some of the EasyEDA in-house models can be informative
because, although they do not come with a 'How it works' written into them,
there is some documentation in their .subckt definitions that may give some
insights into the wild and wonderful world of custom modelling.

Some examples to look through the netlists of are:

>[LDR test]
(https://easyeda.com/editor#mode=sim,id=a21460865ebc4ffd8434d10946c48ec9)

>[Electret microphone model]
(https://easyeda.com/editor#mode=sim,id=b4a2a22f33324ed2afe4dcd012a0f564)

>[Electret microphone model .subckt]
(https://easyeda.com/editor#mode=sim,id=46a20a0af877436fa0d4e8ff293cd5b1)

>[Electret microphone model test jig]
(https://easyeda.com/editor#mode=sim,id=302cb429fc5b4d2f946eaf59b9f72266)

>[LM56EE demo jig]
(https://easyeda.com/editor#mode=sim,id=41c37ad8190340babd5b36978dfadb2a)

>[How to include a 5 Pin Comparator in a schematic]
(https://easyeda.com/editor#mode=sim,id=26936cd661744564accff2961cb28c17)
```

```
---

##  Simulation eBook - Making measurements of simulation results
---



<a name="Making measurements of simulation results"></a>
# Making measurements of simulation results

In the same way that digital storage oscilloscopes (DSO) allow measurements to
be made of signals displayed on the screen using cursors and directly reading
values from their positions and from mathematical analysis of the waveform data
stored in the DSO memory, EasyEDA allows measurements to be made of simulation
results directly using cursors and by analysis of the data used to create the
WaveForm trace display using the `.meas` command.


<a name="Using the WaveForm display cursors"></a>
### Using the WaveForm display cursors

The WaveForm X and Y cursor functions are a simple and quick way to make
measurements of points in waveforms and to make measurements of differences
between points.

WaveForm allows the display of traces in any selection of up to three vertically
stacked plot panes. The Y axes automatically scale to fit the units and the
range of the traces being displayed. Traces can be hidden but at least one trace
must be visible. X and Y trace data can be seen on-screen just by moving the
mouse cursor around the plot area of a pane with the readout adapting to the Y
axes in each pane.

THE FEATURE DESCRIBED IN THE FOLLOWING PARAGRAPH IS CURRENTLY NOT AVAILABLE.
SEE BUG REPORT:
   https://easyeda.com/forum/topic/Left-click-and-drag-Delta-X-and-Delta-Y-info-
no-longer-works-in-Waveform-6a533f06cfca4915b3af0ba40ef90c5e

Delta X and delta Y trace data can be seen on-screen using a Left-Click and Drag
select box, with the readout adapting to the Y axes in each pane. Returning the
cursor to within a small radius of the starting point of the select box -
without releasing the Left-Click - returns the readout to X and Y trace data.

The cursor placement and results produced are volatile, meaning that they cannot
be copied and pasted. They are not saved as part of a saved WaveForm file.
However, using a screenshot utility, it is possible to save an image of the
WaveForm display showing the cursor positions and their associated readout.

Note that the screenshot utility must have a user definable time delay to allow
cursor placement to be carried out between initiating the screenshot and the
screenshot actually being taken.
For more information on displaying simulation results in Waveform, please refer
to the section on **WaveForm** in the [the EasyEDA Tutorial]
(https://easyeda.com/Doc/Tutorial/spiceSimulation.htm#WaveForm).


<a name="Using the EasyEDA Oscilloscope"></a>
### Using the EasyEDA Oscilloscope
```

```
EasyEDA provides an interactive oscilloscope symbol with a short tutorial on how
to use it given here:

>[How to use the EasyEDA Oscilloscope](https://easyeda.com/forum/topic/How-to-
use-the-EasyEDA-Oscilloscope-759bdd4ce3604beab90df403dbc14b69)



<a name="Using the .meas command"></a>
### Using the .meas command
The .meas command is used to analyse the output data of a .ac, .dc. Op, .tf,
.tran, or .noise simulation. The command is executed immediately after the
simulation has finished.
It is entered in the schematic in the same way as other spice directives with
the:
**Text Attribute > Text type > spice**

To use .meas commands, run the simulation in the usual way by pressing:
    **F8**
(was CTRL+R)
 Then after the simulation has completed, go to the top toolbar of the Schematic
tab and do:
    **Simulation > Show your simulation report...**
 to view the .meas statement results.



<a name="The meaning of terms in .MEAS commands"></a>
### The meaning of terms in .MEAS commands
The .meas type **[AC|DC|OP|TRAN|TF|NOISE]** depends on the data which are to be
evaluated, either originating from an AC analysis, DC, Operating point,
Transient, Transfer Function or Noise analysis simulation. The .meas type is
optional but when used allows a .MEAS statement to be used only for the given
analysis type. This allows different selections of .meas statements to be used
depending on the simulation type being run.

**name** will be a variable containing the result of the measurement. A name is
required so that the result can be used as a parameter in other .MEAS
statements.

**trig_variable, targ_variable**, and **out_variable** are vectors stemming from
the simulation, e.g. a voltage vector v(out).

**VAL=val** expects a real number val. val may also be a real parameter or an
expression enclosed by {} that expands to a real number.

**TD=td** and **AT=time** expect a time value if the .meas type is **tran**. For
**ac**, **AT** will be a frequency value and **TD** is ignored.

For **DC** analysis **AT** is a voltage (or current), and **TD** is ignored.

**CROSS=#** requires an integer number #. **CROSS=LAST** is possible as well.
The same is expected by **RISE** and **FALL**.

Frequency and time values may start at 0 and extend to positive real numbers.
Voltage (or current) inputs for the independent (scale) axis in a dc analysis
may start or end at arbitrary real valued numbers.
Results of .MEAS statements can be used as parameters in other .MEASURE
statements.
```

```
Values defined as parameters in .param statements elsewhere in the simulation
can be used as values and in expressions in .meas statements.
Functions can be used in values and expressions in .meas statements except for
those invoking time as a variable (such as delay(x,time), the derivative
function ddt(x) and the integral functions sdt(x) or idt(x)).


<a name="Types, syntaxes and examples of .meas commands"></a>
### Types, syntaxes and examples of .meas commands

<a name="Types"></a>
#### Types
There are two basic different types of .MEASURE statements. Those that refer to
a point along the abscissa (the independent variable plotted along the
horizontal, or x, axis, i.e., the time axis of a .tran analysis or the frequency
axis of a .ac analysis) and .MEASURE statements that refer to a range over the
abscissa.

For the purposes of this Tutorial, the first type of .MEASURE statements will be
referred to as **Point** and the second as **Range**.


<a name="Point"></a>
##### Point
.MEAS statements that point to a single point on the horizontal axis, are used
to print a data value or expression thereof at a specific point or when a
condition is met. The following syntax is used:

Syntax: .MEAS[SURE] [AC|DC|OP|TRAN|TF|NOISE] <name>
+ [<FIND|DERIV|PARAM> <expr>]
+ [WHEN <expr> | AT=<expr>]]
+ [TD=<val1>] [<RISE|FALL|CROSS>=[<count1>|LAST]]

The analysis type is optional but if used specifies the type of analysis to
which the .MEAS statement applies. This allows .MEAS statements to be applied to
only certain analysis types.

The result of the measurement is contained by the variable called name. The
variable name can then allows the result to be used as a parameter in other
.MEAS statements.

 Below are examples of Point type .MEAS statements that return a value referring
to a single point on the horizontal axis:

`.MEAS TRAN res1 FIND V(out) AT=5m`

Print the value of V(out) at t=5ms labelled as res1.

`.MEAS TRAN res2 FIND V(out)*I(Vout) WHEN V(x)=3*V(y)`

Print the value of the expression V(out)*I(Vout) the first time the condition
V(x)=3*V(y) is met. This will be labelled res2.

`.MEAS TRAN res3 FIND V(out) WHEN V(x)=3*V(y) cross=3`

Print the value of V(out) the third time the condition V(x)=3*V(y) is met. This
will be labelled res3.
```

```
`.MEAS TRAN res4 FIND V(out) WHEN V(x)=3*V(y) rise=last`

Print the value of V(out) the last time the condition V(x)=3*V(y) is met when
approached as V(x) increasing wrt 3*V(y). This will be labelled res4.

`.MEAS TRAN res5 FIND V(out) WHEN V(x)=3*V(y) cross=3 TD=1m`

Print the value of V(out) the third time the condition V(x)=3*V(y) is met, but
don't start counting until the time has reached 1ms. This will be labelled res5.

`.MEAS TRAN slope DERIV V(out) AT=5m`

Print the value of the slope of V(out) at t=5ms labelled as slope.

`.MEAS TRAN res6 PARAM 3*res1/res2`

Print the value of 3*res1/res2. This form is useful for printing expressions of
other .meas statement results. It's not intended that expressions based on
direct simulation data, such as V(3), are present in the expression to be
evaluated, but if they are, the data is taken from the last simulated point. The
result will be labelled res6.

Note that in the above examples, whilst the result refers to a single point
along the horizontal axis, it is based on ordinate data (the dependent variables
plotted on the vertical or y axis). If no ordinate information is requested,
then the .MEAS statement prints the point on the abscissa at which the
measurement condition occurs:

`.MEAS TRAN res6 WHEN V(x)=3*V(y)`

Print the first time the condition V(x)=3*V(y) is met. This will be labelled
res6.

Similarly in an AC analysis:

`.MEAS AC acres6 WHEN mag(V(x))=1/sqrt(2)`

Print the lowest frequency that the condition mag(V(x))=1/sqrt(2) is met. This
will be labelled acres6.




<a name="Range"></a>
##### Range
The other type of .MEAS statement refers to a range over the abscissa (the
independent variable plotted along the horizontal, or x, axis, i.e., the time
axis of a .tran analysis or the frequency axis of a .ac analysis). The following
syntax is used:

Syntax: .MEAS [AC|DC|OP|TRAN|TF|NOISE] <name>
+ [<AVG|MAX|MIN|PP|RMS|INTEG> <expr>]
+ [TRIG <lhs1> [[VAL]=]<rhs1>] [TD=<val1>]
+ [<RISE|FALL|CROSS>=<count1>]
+ [TARG <lhs2> [[VAL]=]<rhs2>] [TD=<val2>]
+ [<RISE|FALL|CROSS>=<count2>]
```

The range over the abscissa is specified with the points defined by "TRIG" and "TARG". The TRIG point defaults to the start of the simulation if omitted. Similarly, the TARG point defaults to the end of simulation data. If all three of the TRIG, TARG, and the previous WHEN points are omitted, then the .MEAS statement operates over the entire range of data. The types of measurement operations that can be done over an interval are:

keyword	**Operation performed over interval**
AVG	Compute the average of <expr>
MAX	Find the maximum value of <expr>
MIN	Find the minximum value of <expr>
PP	Find the peak-to-peak of <expr>
RMS	Compute the root mean square of <expr>
INTEG	Integrate <expr>

If no measurement operation is specified, the result of the .MEAS statement is the distance along the abscissa between the TRIG and TARG points.

Below are example interval .MEAS statements:

`.MEAS TRAN res7 AVG V(NS01)`
`+ TRIG V(NS05) VAL=1.5 TD=1.1u FALL=1`
`+ TARG V(NS03) VAL=1.5 TD=1.1u FALL=1`

Print the average value of V(NS01) from the 1st fall of V(NS05) to 1.5V after 1.1us and the 1st fall of V(NS03) to 1.5V after 1.1us. This will be labelled res7.

For .AC analyses, the conditional expressions of complex data are translated to real conditions by considering only the real part of the complex value of the expression.

This is an example of how the result of a .MEAS statement can be used in another .MEAS statement. In this case, the 3dB bandwidth is computed:

`.MEAS AC tmp max mag(V(out)); find the peak response and call it "tmp"`

`.MEAS AC BW trig mag(V(out))=tmp/sqrt(2) rise=1`
`+ targ mag(V(out))=tmp/sqrt(2) fall=last`

Print the difference in frequency between the two points 3dB down from peak response. NOTE: The data from a .AC analysis is complex and so are the .measurement statements results. However, the equality refers only to the real part of the complex number, that is, "mag(V(out))=tmp/sqrt(2)" is equivalent to Re(mag(V(out)))=Re(tmp/sqrt(2)).

The AVG, RMS, and INTEG operations are different for .NOISE analysis than the analysis types since the noise is more meaningfully integrated in quadrature over frequency. Hence AVG and RMS both give the RMS noise voltage and INTEG gives the integrated total noise. Hence, if you add the SPICE directives

`.MEAS NOISE out_totn INTEG V(onoise)`

```
`.MEAS NOISE in_totn INTEG V(inoise)`

the total integrated input and output referenced rms noise will be printed in
the Simulation report.


<a name="Examples of the use of .MEAS statements"></a>
### Examples of the use of .MEAS statements

The .MEAS statements in LTspice are incredibly powerful but it can take a while
to understand how to use them and get the best results.

It is therefore **strongly** recommended that a simplified test simulation
should be set up and played with to try different set ups before using up the
simulation allowance running blockbuster sims from which the .MEAS statements
consistently fail.

To help in this, the following examples illustrate some of the measurements that
can be made using .MEAS statements:

>[Measuring WaveForm parameters 01]
(https://easyeda.com/editor#mode=sim,id=b91ef5c0dbf04a478f3573227affa625)

>[Measuring WaveForm parameters 02]
(https://easyeda.com/editor#mode=sim,id=bcd3a7d6b7b9451db9c2cb2c1e3ce2e3)

>[Measuring settling time]
(https://easyeda.com/editor#mode=sim,id=34e0665e96a1407590d3a5544891a60e)

>[Find gain and bandwidth]
(https://easyeda.com/editor#mode=sim,id=f7e5299c32794f4482fa41178f2627be)

>[How to measure period and frequency using a .meas statement]
(https://easyeda.com/editor#mode=sim,id=|496b9ba3e24a4ec0928e8a0b6d2276a5)


For more information on the .meas statement, search for .measure in Help in a
locally installed copy of LTspiceXVII or visit ltwiki.org here:

   http://ltwiki.org/LTspiceHelpXVII/LTspiceHelp/html/DotMeasure.htm


<a name="More examples of measure statements"></a>
#### More examples of measure statements
```

.meas tran inv_delay 2 trig v(in) val='vp/2' td=1n fall=1 targ v(out) val='vp/2' rise=1

.meas tran test_data1 trig AT=1n targ v(out) val='vp/2' rise=3

.meas tran out_slew trig v(out) val=' 0.2*vp' rise=2 targ v(out) val=' 0.8*vp' rise=2

.meas tran skew when v(out)=0.6

.meas tran skew2 when v(out)=skew_meas

.meas tran skew3 when v(out)=skew_meas fall=2

.meas tran skew4 when v(out)=skew_meas fall=LAST

.meas tran skew5 FIND v(out) AT=2n

.meas tran v0_min min i(v0) from='dfall' to='dfall+period'

.meas tran v0_avg avg i(v0) from='dfall' to='dfall+period'

.meas tran v0_integ integ i(v0) from='dfall' to='dfall+period'

.meas tran v0_rms rms i(v0) from='dfall' to='dfall+period'

.meas dc is_at FIND i(vs) AT=1

.meas dc is_max max i(vs) from=0 to=3.5

.meas dc vds_at when i(vs)=0.01

.meas ac vout_at FIND v(out) AT=1MEG

.meas ac vout_atd FIND vdb (out) AT=1MEG

.meas ac vout_max max v(out) from=1k to=10MEG

.meas ac freq_at when v(out)=0.1

.meas ac vout_diff trig v(out) val=01 rise=1 targ v(out) val=01 fall=1

.meas ac fixed_diff trig AT=10k targ v(out) val=0.1 rise=1

.meas ac vout_avg avg v(out) from=10k to=1MEG

.meas ac vout_integ integ v(out) from=20k to=500k

.meas ac freq_at2 when v(out)=01 fall=LAST

.meas ac vout_rms rms v(out) from=10 to=1G

```
<a name="On the accuracy of the results of .meas statements"></a>
### On the accuracy of the results of .meas statements

.MEAS statements are carried out on post-simulation data, in other words after
the simulation has completed and the data from it stored.

It is important to note that data generated by LTspice is in floating point
format.

LTspice has optional settings on the level of compression that applies to the
data that it generates as it is stored.

The combination of floating point number formats and data compression means that
there is some scope for small errors in numerical accuracy due to the finite
precision used in storing numbers.

Another consequence of this is that the accuracy of the .MEAS statement output
is limited by the accuracy of the waveform data after it has been stored by
LTspice using whatever the default data compression setting is used.

To limit the amount of data being handled by the EasyEDA servers, the full
control of compression available through the Control Panel in a local
installation of LTspice, is not made available to users.
```

It is however possible to reduce the level of compression applied to the stored data - and which is often enough - by including the following .option statement in a simulation schematic as a spice directive:

```
   .option plotwinsize=0
```

Improving accuracy by reducing the level of compression of course may result in so much data being generated during the simulation that it will exceed the EasyEDA internal data storage limits and generate a

```
   "Your Simulation Results are too large"
```

warning.

LTspice is very good at choosing its own value for the Maximum Timestep so it is perfectly permissible to enter a .tran statement simply as:

```
   .tran <Stop Time>
```

with no Maximum Timestep assigned.

The accuracy of the results from a Transient Analysis (.tran) however, is strongly dependent on the ratio:

```
   (Stop Time)/(Maximum Timestep)
```

Therefore to increase the accuracy it may be desirable to add a Maximum Timestep value.

The larger this ratio the more accurate the results will be at the expense of longer simulation times and possibly exceeding the EasyEDA internal data storage limits and generating a

```
   “Your Simulation Results are too large"
```

warning.

Similarly, the accuracy of the results from an AC Analysis (.ac) is strongly dependent on the number of points per decade/octave/sweep.

The larger the number of points, the more accurate the results will be but again there is the risk of generating too much data as described above.

(Note that due to a bug in EasyEDA a linear sweep is diplayed on a log frequency axis with a minimum span of a decade. This currently renders the display of linear frequency sweeps in EasyEDA useless.)

In the light of these accuracy considerations, it is therefore important to understand that when testing a condition such as:

```
   "when <cond1> = <cond2>"
```

It is more reliable to require that the conditions cross each other, rather than to require an exact equality, which may not be met due to an error in numerical accuracy.

```
This can be particularly problematic when trying to find exactly where the peak
value of a signal or plat may occur on the x axis ( i.e. the time or the
frequency axis).

It is easy enough to find the maximum (MAX) value but the x axis value at which
this maximum may occur may not be available as a parameter. It is therefore
tempting to use an equality testing WHEN the signal equals the MAX value to
return the x axis value. Sometimes this may work, other times it may not.

There are several ways to solve this problem but about the most reliable
solution is to test when the signal crosses a point slightly less than the MAX
when approaching from the low side of the peak and again when departing from the
high side.

To do this the signal could be compared to the maximum value minus and offset -
or the maximum value multiplied by a number just less than 1 - on a RISE and
then again on the next FALL. The RISE test returns a value on the x axis just
below when the peak occurs and the FALL test another value just above it. It is
then up to the user to decide whether to use one or the other or some function
such as the average or the geometric mean of the two to give a better estimate
of where the peak actually lies.

Since the slope of a function passes through zero at a peak or a trough, a more
complex solution - although one that only works for Transient analyses (i.e.
with time domain signals only) and which may not work well with noisy signals -
could be to use a B Source with the ddt(x) function to generate the derivative
of the signal being measured and then test for the derivative crossing zero to
find the exact time of the peak or trough.


---

# EasyEDA Document Format


##  EasyEDA Document Format - Standard
---

[TOC]

# Overview

2020.08.11

**document format**is the standard version of EasyEDA of the**file source
format**, divided into two types:

+**sch type**document
+**pcb type**document

The document is a string of the strict 'JSON' type. The following is explained
in JSON format**by the different fields**.

The format of the two document types is very similar. The main**difference is
the difference between the**primitive and some special attribute values. For the
same part, please refer to the**General Document Format**section.
```

~~~
# General Document Format

>Among different types of documents, some formats are common, or some are document-specific, as summarized below.

### separator

```js
//Compressed format separator
var SP = '~',
    SEG = '^^',
    NEWLINE = '#@$',
    PARASP = '`',//Separation marks for individual parameters para SP
    SHEET = '#sheet#';//Load multiple files from the server, Eagle imports + netlistHandle
```

### The file header `head`

> Field name: 'head'. Type Json, which contains the basic information for the entire document.

Examples and data formats:

```js
{
    "docType": "1",              // [Document Type] - string
    "editorVersion": "6.3.0",    // [Document Version Number] - string
    "c_para": {},                // [Custom attribute] - josn
    "x": "4000",                 // [Document the origin X coordinates] - string
    "y": "3000",                 // [Document the origin Y coordinates] - string
    "hasIdFlag": true,           // [The component whether if binding with the library's uuid] - boolean
    "newgId": true,              // [The component whether if has unique id] - boolean
    "importFlag": 0,             // [Marking for importing eagle] - number
    "transformList": "",         // [The offset data] - string(abandoned)
    "c_spiceCmd": "",            // [Simulation command] - string(Only for schematic document)(abandoned)
    "isSheet": true,             // [is drawing symbol or not] - boolean
    "importedFrom": ''           // [importedFrom] - already abandoned
}
```

### The canvas configuration `canvas`

>The field name: `canvas`,  Type string,  Use the separator '~' to separate. Used to identify canvas related properties.

for example:

```js
"CA~1000~1000~#000000~yes~#FFFFFF~10~1000~1000~line~0.5~mm~1~45~~0.5~4000~3000~0~none"
```

Format specification:
~~~

~~~
**schematic**:

```txt
1.[CA]:  Fixed identity, canvas
2.[viewWidth]:  The width attribute value of an SVG element, the larger the
zoom, the larger the value. (Used in previous versions)
3.[viewHeight]:  The height attribute value of an SVG element, the larger the
zoom, the larger the value. (Used in previous versions)
4.[backGround]:  background color
5.[gridVisible]:  grid visible or not
6.[gridColor]:  grid color
7.[gridSize]:  grid size
8.[canvasWidth]:  The logical width of the canvas, that is, the width in the SVG
viewBox, does not change with ZOOM (used in previous versions)
9.[canvasHeight]:  The logical height of the canvas, that is, the width in the
SVG viewBox, does not change with ZOOM (used in previous versions)
10.[gridStyle]: grid style
11.[snapSize]:  snap size for snap
12.[unit]:  unit
13.[altSnapSize]:  snap size for ALT snap
14.[originX]:  the origin X coordinates for canvas
15.[originY]:  the origin Y coordinates for canvas
```

**pcb**:

```txt
1.[CA]:  Fixed identity, canvas
2.[viewWidth]:  The width attribute value of an SVG element, the larger the
zoom, the larger the value. (Used in previous versions)
3.[viewHeight]:  The height attribute value of an SVG element, the larger the
zoom, the larger the value. (Used in previous versions)
4.[backGround]:  background color
5.[gridVisible]:  grid visible or not
6.[gridColor]:  grid color
7.[gridSize]:  grid size
8.[canvasWidth]:  The logical width of the canvas, that is, the width in the SVG
viewBox, does not change with ZOOM (used in previous versions)
9.[canvasHeight]:  The logical height of the canvas, that is, the width in the
SVG viewBox, does not change with ZOOM (used in previous versions)
10.[gridStyle]: grid style
11.[snapSize]:  snap size for snap
12.[unit]:  unit
13.[routingWidth]:  routing width
14.[routingAngle]:  routing Angle
15.[copperAreaDisplay]:  copper area fill date visible or not
16.[altSnapSize]:  snap size for ALT snap
17.[originX]:  the origin X coordinates for canvas
18.[originY]:  the origin Y coordinates for canvas
19.[routeConflict]:  routing Conflict(Ignore | TrackAround | Block)
20.[removeLoop]:  Whether to remove the loop
```

### Primitive data `shape`

> The field name:`shape`. String array type. Each piece of data corresponds to
the compressed data of each graph element.
~~~

~~~
The schematic contains data for rectangles, text, and ellipses

for example:

```js
[
    "R~440~300~~~50~110~#000000~1~0~none~gge5~0~",
    "T~L~360~300~0~#0000FF~~9pt~~~~comment~Text~1~start~gge8~0~pinpart",
    "E~410~355~10~25~#000000~1~0~none~gge12~0"
]
```

Refer to the single data structure of the primitive format: **Primitive
format**.

### Bounding box data `BBox`

> field name  `BBOX`. Type json. Box model data for the entire document.

for example:

```js
{
    "x": 4033.1,        // The abscissa of the upper left corner of the document
canvas boundary
    "y": 3282.1,        // The ordinate of the upper left corner of the document
canvas boundary
    "width": 113.3,     // Document canvas border width
    "height": 147.8     // Document canvas border height
}
```

### Layer configuration `layers`(PCB)

> The field name:`layers`. String array type. Layer configuration information is
only in the PCB type document,  The display and activation of layer tools are
recorded.

for example:

```js
[
    "1~TopLayer~#FF0000~true~true~true~",
    "2~BottomLayer~#0000FF~true~false~true~",
    "3~TopSilkLayer~#FFCC00~true~false~true~",
    "4~BottomSilkLayer~#66CC33~true~false~true~",
    "21~Inner1~#999966~true~false~true~0~Plane",
]
```

数组中每条数据表示对应层的相关层信息，使用间隔符`~`分开，数据格式为：

```txt
1.[layerid]：层的id标识
2.[name]：层名称
3.[color]：层颜色
4.[visible]：层是否可见
5.[active]：是否为当前激活层
~~~

~~~
6.[config]：是否配置当前层
7.[transparency]：层透明度
8.[type]：层类型（内电层 | 信号层），内层专有配置
```

### 元素的可选与显示配置`objects`(PCB)

> 字段：`objects`。字符串数组类型。记录层工具的元素可选与显示设置信息（pcb类型文档专用）。

示例：

```js
[
    "All~true~false",
    "Component~true~true",·
    "Prefix~true~true",
    "Name~true~false",
    "Track~true~true",
    "Pad~true~true",
    "Via~true~true",
    "Hole~true~true",
    "Copper_Area~true~true",
    "Circle~true~true",
    "Arc~true~true",
    "Solid_Region~true~true",
    "Text~true~true",
    "Image~true~true",
    "Rect~true~true",
    "Dimension~true~true",
    "Protractor~true~true"
]
```

数据格式：

```txt
1.[key]：元素关键字标识符（不能包含空格，会用_替换带有空格的key）
2.[selected]：是否可选中
3.[visible]：是否显示
```

### 偏好设置`preference`(PCB)

> 字段：`preference`。json类型。记录文档的一些偏好设置（pcb类型文档专用）。

示例：

```js
{
    "hideFootprints": "",
    "hideNets": ""
}
```

### drc检测规则`DRCRULE`(PCB)

> 字段：`DRCRULE`。json类型。记录drc检测规则配置信息（pcb类型文档专用）。
~~~

~~~
示例：

```js
{
    "Default": {                // 默认配置
      "trackWidth": 1,          // 线宽
      "clearance": 0.6,         // 检测间距
      "viaHoleDiameter": 2.4,   // 孔外径
      "viaHoleD": 1.2           // 孔内径
    },
    "isRealtime": false,                // 是否实时检测
    "isDrcOnRoutingOrPlaceVia": false,  // 是否在布线与放置过孔时应用设计规则
    "checkObjectToCopperarea": true,    //  是否检测元素到铺铜的距离
    "showDRCRangeLine": true            // 是否在布线时显示DRC安全边界
  }
```

### 自动布线规则`routerRule`(PCB)

> 字段：`routerRule`。json类型。记录布线规则的配置信息（pcb类型文档专用）。

```js
{
    "unit": "mil",              // 单位
    "trackWidth": 6,            // 线宽
    "trackClearance": 6,        // 间距
    "viaHoleD": 12,             // 孔内径
    "viaDiameter": 24,          // 孔外径
    "routerLayers": [           // 布线层配置
      1,
      2
    ],
    "smdClearance": 6,          // 贴片和器件的间距
    "specialNets": [            // 特殊网络配置
      {
        "net": "AAA",           // 网络
        "width": "15mil",       // 线宽
        "clearance": "11mil"    // 间距
      }
    ],
    "nets": [                   // 所有网络
      "AAA"
    ],
    "padsCount": 1,             // 焊盘数量
    "skipNets": [],             // 忽略的网络配置
    "realtime": true            // 实时检测
}
```

### 网络颜色`netColors`(PCB)

> 字段：`netColors`。json类型。记录网络颜色配置信息（pcb类型文档专用）。

示例：

```js
{
    "DSA": {
~~~

~~~
      "color": "#339933"
    },
    "FSADFDSA": {
      "color": "#FF0000"
    }
}
```

### 仿真数据(波形图)`waveForm`

> 字段：`waveForm`。json字符串类型。仿真波形图专用字段，用于存储波形图数据。

# 图元格式

> 基础图元组合起来就是一个完整的器件（封装），比如最基本的矩形、圆这种**简单图元**，复杂一点的
图元如引脚、网络标签、焊盘等。**复杂图元**其实也是简单基本图元组成的。
>
> 原理图的基础图元和pcb的基础图元基本都不一致。
>
> 两种文档类型的基础图元都有一种**特殊图元**，特殊图元独立存在，不再组成其他元素，比如原理图的
`schlib`和pcb的`FOOTPRINT`，即原理图器件和pcb封装，这两种图元基本都**由其他基础图元组成
**，但不允许有本身图元组成。
>
> 数据类型：string。基本数据以分隔符`~`隔开。

### 原理图部分

#### 1.简单图元

##### 折线`polyline`

示例：

```js
"PL~230 290 430 180 550 340~#000000~1~0~none~gge5~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，默认为"PL"。
2.[pointerStr]:折线polyline的points数据字符串
3.[strokeColor]:线条颜色
4.[strokeWidth]:线条宽度
5.[strokeStyle]:线条样式
6.[fillColor]:填充颜色
7.[gId]:元素id
8.[locked]:是否锁定
```

##### 导线`wire`

示例：

```js
"W~495 -580 495 -475 580 -475~#008800~1~0~none~gge79~0"
```
~~~

~~~~markdown
导线的数据格式除了首位标识符为`W`，其余结构同`polyline`一致。

##### 总线`bus`

示例：

```js
"B~410 -565 400 -455~#008800~2~0~none~gge80~0"
```

总线的数据格式除了首位标识符为`B`，其余结构同`polyline`一致。

##### 多边形`polygon`

示例：

```js
"PG~290 260 340 350 440 230 390 200 390 200~#000000~1~0~none~gge5~0"
```

多边形的数据格式除了首位标识符为`PG`，其余结构同`polyline`一致。

##### 矩形`rect`

示例：

```js
"R~360~250~~~70~80~#000000~1~0~none~gge6~0~"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，R
2.[x]：横坐标
3.[y]：纵坐标
4.[rx]：水平轴向的圆角半径尺寸
5.[ry]：垂直轴向的圆角半径尺寸
6.[width]：矩形宽度
7.[height]：矩形高度
8.[strokeColor]：线条颜色
9.[strokeWidth]：线条宽度
10.[strokeStyle]：线条样式
11.[fillColor]：填充颜色
12.[gId]：元素id
13.[locked]：是否锁定
14:[c_etype]:c_etype属性值（自定义的用于细分图元类型的属性）
```

##### 图片`image`

示例：

```js
"I~360~290~104~26~0~data:image/png;base64,iVBORw......TkSuQmCC~gge11~0~"
```

数据格式：
~~~~

~~~markdown
```txt
1.[cmdKey]：图元标识符，I
2.[x]：x坐标
3.[y]：y坐标
4.[width]：图片宽度
5.[height]：图片高度
6.[rotate]：旋转角度
7.[href]：图片的数据信息
8.[gId]：元素id
9.[locked]：是否锁定
10.[transform]：偏移属性
```

##### 圆`circle`

注：新版本已弃用（椭圆可代替）。

示例：

```js
 "C~685~315~35~#000000~1~0~none~gge21~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，C
2.[cx]：圆心x坐标
3.[cy]：圆心y坐标
4.[r]：半径
5.[strokeColor]：线条颜色
6.[strokeWidth]：线条宽度
7.[strokeStyle]：线条样式
8.[fillColor]：填充颜色
9.[gId]：元素id
10.[locked]：是否锁定
```

##### 椭圆`ellipse`

示例：

```js
"E~420~295~40~25~#000000~1~0~none~gge22~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，E
2.[cx]：圆心x坐标
3.[cy]：圆心y坐标
4.[rx]：x轴上椭圆的半径
5.[ry]：y轴上椭圆的半径
6.[strokeColor]：线条颜色
7.[strokeWidth]：线条宽度
8.[strokeStyle]：线条样式
~~~

9.[fillColor]：填充颜色
10.[gId]：元素id
11.[locked]：是否锁定

##### 直线`line`

示例：

```js
"L~0~470~-765~480~-775~gge105~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，L
2.[x1]：起点横坐标
3.[y1]：起点纵坐标
4.[x2]：终点横坐标
5.[y2]：终点纵坐标
6.[strokeColor]：线条颜色
7.[strokeWidth]：线条宽度
8.[strokeStyle]：线条样式
9.[fillColor]：填充颜色
10.[gId]：元素id
11.[locked]：是否锁定
```

##### 路径`path`

示例：

```js
"PT~M370 280 C410 380 490 300 440 240~#000000~1~0~none~gge23~0~"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，PT
2.[pathString]：path路径
3.[strokeColor]：线条颜色
4.[strokeWidth]：线条宽度
5.[strokeStyle]：线条样式
6.[fillColor]：填充颜色
7.[gId]：元素id
8.[locked]：是否锁定
```

##### 圆弧`arc`

示例：

```js
"A~M 459.9983 319.997 A 70 170 0 1 1 461.2807
267.9875~~#000000~1~0~none~gge26~0"
```

~~~markdown
数据格式：

```txt
1.[cmdKey]：图元标识符，A
2.[pathString]：path路径
3.[helperDots]：已移除，占位。
4.[strokeColor]：线条颜色
5.[strokeWidth]：线条宽度
6.[strokeStyle]：线条样式
7.[fillColor]：填充颜色
8.[gId]：元素id
9.[locked]：是否锁定
```

##### 文本`annotation`

示例：

```js
"T~L~390~300~0~#0000FF~~9pt~~~~comment~Text~1~start~gge27~0~pinpart"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，A
2.[mark]：文本标记，可选值：L(普通文本) | N(器件名称) | P(器件编号) | PK(封装名)
3.[x]：横坐标
4.[y]：纵坐标
5.[rotation]：旋转角度
6.[fillColor]：填充颜色
7.[fontFamily]：字体
8.[fontSize]：文字大小
9.[fontWeight]：文字粗细
10.[fontStyle]：文字样式（自动 | 正常 | 斜体）
11.[dominantBaseline]：文字的dominant-baseline属性
12.[type]：文本类型（注释 | 仿真）
13.[value]：文本值
14.[visible]：是否可见
15.[textAnchor]：文字的text-anchor属性
16.[gId]：元素id
17.[locked]：是否锁定
18.[c_etype]：c_etype属性值（c_etype是自定义的用于细分图元类型的属性）
```

##### 表格图片`pimage`

注：目前只用于表格内的图片，为了与普通图片做区分使用的一种新类型图元表示。

示例：

```js
"Pimage~L~1~gge74~0~gge75~694.99995~-38~104~26~data:image/png;base64,iVBORw0KG..
....pCvHNepOOktyLeC61RcZ3JlkHPt9dXiX4IAntNzzH4CAAAAAElFTkSuQmCC"
```

数据格式：
~~~

~~~
```txt
1.[cmdKey]：图元标识符，Pimage
2.[mark]：特殊标记（未使用）
3.[visible]：是否可见（未使用）
4.[locked]：是否锁定
5.[imgid]：元素id
6.[imgx]：x坐标
7.[imgy]：y坐标
8.[imgwidth]：图片宽度
9.[imgheight]：图片高度
10.[imghref]：图片base64数据信息
```

#### 2.复杂图元

##### 引脚`pin`

注：引脚的数据格式使用了两种分隔符，外层整体数据使用`^^`，内层单一数据再使用`~`分隔。

示例：

```js
"P~show~0~1~420~300~~gge31~0^^420~300^^M 420 300 h
-20~#880000^^1~398~303~0~1~end~~~#0000FF^^1~405~299~0~1~start~~~#0000FF^^0~403~3
00^^0~M 400 297 L 397 300 L 400 303"
```

数据格式：

```txt
#1.[configure] - 配置信息
1.[cmdKey]：图元标识符，P
2.[dispaly]：是否显示
3.[electric]：电气属性
4.[spicePin]：仿真编号
5.[x]：横坐标   //弃用，参考path
6.[y]：纵坐标   //弃用，参考path
7.[rotation]：旋转角度
8.[gId]：元素id
9.[locked]：是否锁定

#2.[pinDot] - 引脚吸附点(起点)信息
1.[x]：起点横坐标   //弃用，参考path
2.[y]：起点纵坐标   //弃用，参考path

#3.[path] - 引脚路径信息
1.[pathString]：路径数据
2.[pinColor]：引脚颜色

#4.[name] - 引脚名称信息
1.[visible]：是否可见
2.[x]：横坐标
3.[y]：纵坐标
4.[rotation]：旋转角度
5.[text]：文本值
6.[textAnchor]：文字的text-anchor属性
7.[fontFamily]：字体
~~~

~~~
8.[fontSize]：文本大小
9.[fillColor]：文本填充颜色

#5.[num] - 引脚编号信息
1.[visible]：是否可见
2.[x]：横坐标
3.[y]：纵坐标
4.[rotation]：旋转角度
5.[text]：文本值
6.[textAnchor]：文字的text-anchor属性
7.[fontFamily]：字体
8.[fontSize]：文本大小
9.[fillColor]：文本填充颜色

#6.[num] - 引脚终点信息
1.[visible]：  终点是否可见
2.[x]：终点横坐标
3.[y]：终点纵坐标

#7.[clock] - 引脚时钟信息
1.[visible]：  时钟是否显示
2.[pathString]：时钟路径信息
```

##### 连接点`junction`

示例：

```js
"J~425~-170~2.5~#CC0000~gge130~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，J
2.[x]：横坐标
3.[y]：纵坐标
4.[r]：半径
5.[fillColor]：填充颜色
6.[gId]：元素id
7.[locked]：是否锁定
```

##### 标识符`netflag`

注：同引脚的数据格式类似，使用了两种分隔符，外层整体数据使用`^^`，内层单一数据再使用`~`分隔。

示例：

```js
"F~part_netLabel_+5V~425~-170~270~gge124~~0^^425~-170^^+5V~#000000~435.5~-166~0~
start~1~Times New Roman~9pt~gge126^^PL~435 -170 425
-170~#000000~1~0~none~gge128~0^^PL~435 -175 435
-165~#000000~1~0~transparent~gge129~0"
```

数据格式：
~~~

~~~
```txt
#1.[configure] - 配置信息
1.[cmdKey]：图元标识符，F
2.[partId]：标识符类型以及文本信息
3.[x]：横坐标
4.[y]：纵坐标
5.[rotation]：旋转角度
6.[gId]：元素id
7.[transform]：偏移信息（旧版本）
8.[locked]：是否锁定

#2.[pinDot] - 吸附点信息
1.[x]：吸附点横坐标
2.[y]：吸附点纵坐标

#3.[mark] - 标识符文本信息
1.[netFlagString]：文本值
2.[fillColor]：填充颜色
3.[x]：横坐标
4.[y]：纵坐标
5.[rotation]：旋转角度
6.[textAnchor]：文字的text-anchor属性
7.[visible]：是否可见
8.[fontFamily]：字体
9.[fontSize]：文本大小
10.[gId]：文本元素id
```

##### 网络标签`netlabel`

示例：

```js
"N~515~-430~0~#0000ff~netLabel1~gge78~start~517~-432.5~Times New Roman~7pt~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，N
2.[x]：横坐标
3.[y]：纵坐标
4.[rotation]：旋转角度
5.[fillColor]：填充颜色
6.[textAnchor]：文字的text-anchor属性
7.[visible]：是否可见
8.[fontFamily]：字体
9.[fontSize]：文本大小
10.[gId]：文本元素id
```

##### 总线分支`busentry`

示例：

```js
"BE~0~495~-405~505~-415~gge78~0"
~~~

~~~
```

数据格式：

```txt
1.[cmdKey]：图元标识符，BE
2.[x1]：起点横坐标
3.[y1]：起点纵坐标
4.[x2]：终点横坐标
5.[y2]：终点纵坐标
6.[gId]：元素id
7.[locked]：是否锁定
```

##### 箭头`arrowhead`

示例：

```js
"AR~part_arrowhead~410~300~gge5~0~M 410 300 L 395 307.5 L 398.75 300 L 395 292.5
Z ~#000000~0~3~15"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，AR
2.[partType]：放置类型，固定为"part_arrowhead"
3.[x]：横坐标
4.[y]：纵坐标
5.[gId]：文本元素id
6.[rotation]：旋转角度
7.[pathString]：箭头路径数据
8.[fillColor]：填充颜色
9.[locked]：是否锁定
10.[arrowtype]：箭头类型（1 | 2 | 3）
11.[arrowsize]：箭头尺寸
```

##### 无连接标志`noconnectflag`

示例：

```js
"O~0~0~gge86~M-4 -4L4 4M4 -4L-4 4~#33cc33~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，O
2.[x]：横坐标
3.[y]：纵坐标
4.[gId]：文本元素id
5.[pathString]：路径数据
6.[strokeColor]：线条颜色
7.[locked]：是否锁定
```
~~~

~~~
#### 3.特殊图元

##### 形状`SHEET`

这里的形状可以理解为自由格式的组合，可以自定义数据格式，目前原理图没有使用入口，**主要在pcb中使
用**，例如pcb文档中的**图形**，**3D模型**等都使用的`svgnode`图元进行构建。更多细节请参考`基
础图元格式 > PCB部分 > 特殊图元 > 形状svgnode`。

##### 器件`schlib`

原理图的`schlib`即原理图器件（库文件），库文件可以由其他图元组成，但不能由库本身组成库，原理图
库只能放置在原理图中。

在原理图中，`schlib`压缩数据**同其他图元一样**会放到`shape`字段下，`schlib`格式分为两部分组
成：

+ 器件的头部数据：头部数据以分隔符`~`隔开。
+ 器件的内部图元数据：图元之间将会以分隔符`#@$`隔开。

两部分数据再以分隔符号`#@$`隔开。

示例：

```json
"LIB~0~0~package`R0201`nameAlias`Value(Ω)`BOM_Supplier
Part``BOM_Supplier``Contributor`LCEDA_Lib`spicePre`R`spiceSymbolName`R_0201_US`B
OM_Manufacturer
Part``~~0~gge3fb2f834261e887c~b600ab0518154c0c9d2ed5ca96244c69~36be002046ee442da
ec3a30b0f22c9fe~0~~yes~yes#@$T~N~-5.96875~-8~0~#000080~Arial~~~~~comment~1k~1~st
art~gge167~0~#@$T~P~-5.96875~-17.15625~0~#000080~Arial~~~~~comment~R1~1~start~gg
e169~0~#@$PL~15 0 12 5~#A00000~1~0~none~gge171~0#@$PL~7 -5 2
5~#A00000~1~0~none~gge172~0#@$PL~12 5 7
-5~#A00000~1~0~none~gge173~0#@$P~show~0~2~20~0~0~gge174~0^^20~0^^M 20 0 h
-5~#800^^0~11~0~0~2~end~~~#800^^0~19~-4~0~2~start~~~#800^^0~18~0^^0~M 15 -3 L 12
0 L 15 3#@$PL~-7 5 -12 -5~#A00000~1~0~none~gge181~0#@$PL~2 5 -2
-5~#A00000~1~0~none~gge182~0#@$PL~-2 -5 -7 5~#A00000~1~0~none~gge183~0#@$PL~-12
-5 -15 0~#A00000~1~0~none~gge184~0#@$P~show~0~1~-20~0~180~gge185~0^^-20~0^^M -20
0 h 5~#800^^0~-11~0~0~1~start~~~#800^^0~-19~-4~0~1~end~~~#800^^0~-18~0^^0~M -15
3 L -12 0 L -15 -3"
```

数据格式：

```txt
#1.[器件头部数据]
1.[cmdKey]：图元标识符，LIB
2.[x]：横坐标
3.[y]：纵坐标
4.[c_para]：自定义属性
5.[rotation]：旋转角度
6.[importFlag：eagle导入标记
7.[gId]：元素id
8.[puuid]：绑定的pcb封装uuid
9.[uuid]：器件的uuid
10.[locked]：是否锁定
11.[bind_pcb_id]：绑定的pcb封装的id（弃用）
12.[convert_to_pcb]：是否更新到pcb
~~~

~~~
13.[add_into_bom]：是否加入bom表单

#2.[器件内部图元数据]：即其他基本图元的压缩格式。
```


### PCB部分


#### 1.简单图元


##### 导线`TRACK`


示例：


```js
"TRACK~1~1~~4055.5 3348 4055.5 3346 4131.5 3270 4096 3270~gge5~0"
```


数据格式：


```txt
1.[cmdKey]：图元标识符，TRACK
2.[strokeWidth]:线宽
3.[layerid]：所属层
4.[net]：网络
5.[pointArr]：坐标点数据
6.[gId]：元素id
7.[locked]：是否锁定
```


##### 矩形`RECT`


示例：


```js
"RECT~4065.5~3293.25~63~45.5~1~gge6~0~0~~~~"
```


数据格式：


```txt
1.[cmdKey]：图元标识符，RECT
2.[x]:横坐标
3.[y]：纵坐标
4.[width]：宽度
5.[height]：高度
6.[layerid]：所属层
7.[gId]：元素id
8.[locked]：是否锁定
9.[strokeWidth]:线宽
10.[fill]:填充颜色
11.[transform]:偏移数据
12.[net]：网络
13.[c_etype]:c_etype属性值（自定义的用于细分图元类型的属性）
```


##### 圆`CIRCLE`


示例：
~~~

~~~
```js
"CIRCLE~4193.5~3148~45.6426~1~1~gge5~0~~circle_gge8,circle_gge9"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，CIRCLE
2.[cx]：圆心x坐标
3.[cy]：圆心y坐标
4.[r]：半径
5.[strokeWidth]：线宽
6.[layerid]：所属层
7.[gId]：元素id
8.[locked]：是否锁定
9.[net]：网络
10.[transformarc]:由圆转换的两个半圆的id信息
```

##### 文本`TEXT`

示例：

```js
"TEXT~L~4081~3306.5~0.8~0~0~1~~8~TEXT~M 4083.55 3298.44 L 4083.55 3306.07 M 4081
3298.44 L 4086.09 3298.44 M 4088.49 3298.44 L 4088.49 3306.07 M 4088.49 3298.44
L 4093.22 3298.44 M 4088.49 3302.07 L 4091.4 3302.07 M 4088.49 3306.07 L 4093.22
3306.07 M 4095.62 3298.44 L 4100.71 3306.07 M 4100.71 3298.44 L 4095.62 3306.07
M 4105.65 3298.44 L 4105.65 3306.07 M 4103.11 3298.44 L 4108.2
3298.44~~gge13~~0~pinpart"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，TEXT
2.[type]：文本标记，可选值：L(普通文本) | N(器件名称) | P(器件编号) | PK(封装名)
3.[x]：横坐标   //弃用，参考pathStr
4.[y]：纵坐标   //弃用，参考pathStr
5.[strokeWidth]:线宽
6.[rotation]：旋转角度    //弃用，参考pathStr
7.[mirror]:是否镜像      //弃用，参考pathStr
8.[layerid]：所属层
9.[net]：网络
10.[fontSize]：文字大小
11.[text]：文本值        //弃用，参考pathStr
12.[pathStr]：路径数据
13.[display]：是否显示
14.[gId]：元素id
15.[fontFamily]：字体
16.[locked]：是否锁定
17.[c_etype]：c_etype属性值（c_etype是自定义的用于细分图元类型的属性）
```

##### 圆弧`ARC`

示例：
~~~

```js
"ARC~1~1~~M 4108.3572 3265.4999 A 53.7587 53.7587 0 1 0 4130.4018 3353.4528~~gge16~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，ARC
2.[strokeWidth]:线宽
3.[layerid]：所属层
4.[net]：网络
5.[d]：路径数据
6.[c_helper_dots]：辅助线路径数据
7.[gId]：元素id
8.[locked]：是否锁定
```

#### 2.复杂图元

##### 焊盘`PAD`

示例：

```js
"PAD~ELLIPSE~4020~3308.5~6~6~11~~1~1.8~~0~gge5~0~~Y~0~~~4020,3308.5",
```

数据格式：

```txt
1.[cmdKey]：图元标识符，PAD
2.[shape]：焊盘形状
3.[x]：横坐标
4.[y]：纵坐标
5.[width]：宽度
6.[height]：高度
7.[layerid]：所属层
8.[net]：网络
9.[number]：编号
10.[holeR]：孔直径
11.[pointArr]：坐标点数据
12.[rotation]：旋转角度
13.[gId]：元素id
14.[holeLength]：孔长度
15.[slotPointArr]：孔的坐标点数据
16.[plated]：是否金属化
17.[locked]：是否锁定
18.[pasteexpansion]：助焊扩展
19.[solderexpansion]：阻焊扩展
20.[holeCenter]：孔中心坐标
```

##### 过孔`VIA`

示例：

~~~
```js
"VIA~4030~3308.5~2.4~~0.6~gge11~0",
```

数据格式：

```txt
1.[cmdKey]：图元标识符，VIA
2.[x]：横坐标
3.[y]：纵坐标
4.[diameter]：过孔直径
5.[net]：网络
6.[holeR]：过孔内径
7.[gId]：元素id
8.[locked]：是否锁定
```

##### 通孔`HOLE`

示例：

```js
"HOLE~4041.5~3309~4~gge15~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，HOLE
2.[x]：横坐标
3.[y]：纵坐标
4.[holeR]：孔直径
5.[gId]：元素id
6.[locked]：是否锁定
```

##### 铺铜`COPPERAREA`

示例：

```js
 "COPPERAREA~1~1~GND~M 4055 3023.5 L 4055 3060.5 L 4060 3065.5 L 4108.5 3065.5 L
4115.5 3058.5 L4115.5,3025 L4114,3023.5
Z~1~solid~gge83~spoke~none~~0~~1~1~1~1~yes~0",
```

数据格式：

```txt
1.[cmdKey]：图元标识符，COPPERAREA
2.[strokeWidth]：线宽
3.[layerid]：所属层
4.[net]：网络
5.[pathStr]：路径数据
6.[clearanceWidth]：间距
7.[fillStyle]：填充样式
8.[gId]：元素id
9.[thermal]：焊盘连接方式（发散 | 直连）
~~~

~~~
10.[keepIsland]：是否保留孤岛
11.[compressData]：铺铜压缩数据
12.[locked]：是否锁定
13.[name]：名称
14.[order]：顺序
15.[gridTrackWidth]：网格线宽
16.[gridClearance]：网格间距
17.[toBoardOutline]：到边框间距
18.[fabricationImprove]：是否制造优化
19.[spoke_width]：发散线宽
```

##### 实心填充`SOLIDREGION`

示例：

```js
"SOLIDREGION~1~~M 4012 3300.5 L 4012 3317.5 L 4019.5 3325 L 4034 3325 L 4040
3319 L 4040 3308 L 4031 3299 L4019.5,3299 L4018.5,3298 Z~solid~gge20~~~~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，SOLIDREGION
2.[layerid]：所属层
3.[net]：网络
4.[pathStr]：路径数据
5.[type]：类型
6.[gId]：元素id
7.[teardrop]：泪滴
8.[targetPad]：目标焊盘
9.[targetWire]：目标导线
10.[locked]：是否锁定
```

##### 尺寸`DIMENSION`

示例：

```js
"DIMENSION~12~M4015.9632,3321.9632L4038.4632,3299.4632M......M 4035.5 3296.5
4045.5342 3306.5342~gge21~~0~straight~0.4"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，DIMENSION
2.[layerid]：所属层
3.[d]：路径数据
4.[gId]：元素id
5.[fontSize]：尺寸高度
6.[locked]：是否锁定
7.[measuring_type]：尺寸类型
8.[font_width]：尺寸宽度
```
~~~

~~~~
##### 量角器`PROTRACTOR`

示例：

```js
"PROTRACTOR~12~M4051.5 3328 L4051.5 3299.4825M4051.5 3328...... L4069.39 3309.77
~0.4~gge22~4.5~0~0"
```

数据格式：

```txt
1.[cmdKey]：图元标识符，PROTRACTOR
2.[layerid]：所属层
3.[d]：路径数据
4.[strokeWidth]：线宽
5.[gId]：元素id
6.[fontSize]：文字大小
7.[precision]：精度
8.[locked]：是否锁定
```

#### 3.特殊图元

##### 形状`SVGNODE`

这里的形状可以理解为自由格式的组合，可以自定义数据格式，**主要在pcb中使用**，例如pcb文档中的**
图形**，**3D模型**等都使用的`svgnode`图元进行构建。

比如导入的图形：

```js
"SVGNODE~
{\"gId\":\"gge65\",\"nodeName\":\"path\",\"nodeType\":1,\"layerid\":\"1\",\"attr
s\":{\"d\":\"M 4002.49 2996.53 L 4001.2451 2998.03 L 3999.3676 2998.03 L 3997.49
2998.03 L 3997.49 2999.471 C 3997.49 3000.2635 3997.9693 3001.2082 3998.5551
3001.5703 L 3999.6203 3002.2286 L 3997.8946 3004.1354 L 3996.1689 3006.0423 L
3996.8233 3006.6967 L 3997.4777 3007.3511 L 3999.3405 3005.6653 L 4001.2032
3003.9795 L 4002.4599 3005.4938 L 4003.7166 3007.008 L 4004.6033 3006.46 C
4005.091 3006.1586 4005.49 3005.1083 4005.49 3004.126 L 4005.49 3002.34 L
4007.4981 3000.5227 L 4009.5062 2998.7054 L 4007.8431 2996.8677 C 4005.8255
2994.6383 4004.1478 2994.5324 4002.49
2996.53\",\"id\":\"gge65\",\"stroke\":\"none\",\"layerid\":\"1\"}}"
```

数据格式：

`SVGNODE`的格式由两部分组成，第一部分只有图元标识符"SVGNODE"，第二部分为形状的组成数据，数据
格式为`JSON`数据字符串。

##### 内电层`PLANEZONE`

示例：

```js
"PLANEZONE~21~GND~solid~planeL21B0#@$planeL21B0I0~M4342.5,3232 4342.5,3169
4420.5,3169 4420.5,3232zM4375.6222,3223.2077 4375.6314,3223.2077
4376.0759,3223.2015 4376.0852,3223.2012 4376.5294,3223.1869z"
~~~~

~~~markdown
```

数据格式：

压缩数据**同其他图元一样**会放到`shape`字段下，格式由两部分组成：

- 内电层的基本信息：数据以分隔符`~`隔开。
- 内电层路径数据：包含若干路径数据，数据以分隔符`~`隔开。

两部分数据再以分隔符号`#@$`隔开。

```txt
#1.[内电层基本信息]
1.[cmdKey]：图元标识符，PLANEZONE
2.[layerid]：所属层
3.[net]：网络
4.[fillStyle]：填充类型
5.[gId]：元素id

#2.[内电层路径数据]
1.[gId]：路径元素id
2.[pathStr]：路径数据
```

##### 封装`FOOTPRINT`

pcbd的`FOOTPRINT`同原理图的`schlib`类似，即PCB库文件。库文件可以由其他图元组成，但不能由库本身组成库，pcb库只能放置在PCB中。

在PCB中，`FOOTPRINT`压缩数据**同其他图元一样**会放到`shape`字段下，`FOOTPRINT`格式分为两部分组成：

- 封装的头部数据：头部数据以分隔符`~`隔开。
- 封装的内部图元数据：图元之间将会以分隔符`#@$`隔开。

两部分数据再以分隔符号`#@$`隔开。

示例：

```js
~~~

~~~
"LIB~4058.9147~3070~package`R0201`~~~gge929a3f9f08f43af9~1~b600ab0518154c0c9d2ed
5ca96244c69~1565922186~0~#@$TEXT~N~4057.97~3060~0.6~0~~3~~4.5~1k~M 4057.97
3056.28 L 4058.38 3056.07 L 4058.99 3055.46 L 4058.99 3059.75 M 4060.34 3055.46
L 4060.34 3059.75 M 4062.39 3056.89 L 4060.34 3058.94 M 4061.16 3058.12 L
4062.59 3059.75~none~gge3~~0~#@$TEXT~P~4057.97~3067~0.6~0~~3~~4.5~R1~M 4057.97
3062.46 L 4057.97 3066.75 M 4057.97 3062.46 L 4059.81 3062.46 L 4060.42 3062.66
L 4060.63 3062.87 L 4060.83 3063.28 L 4060.83 3063.69 L 4060.63 3064.1 L 4060.42
3064.3 L 4059.81 3064.5 L 4057.97 3064.5 M 4059.4 3064.5 L 4060.83 3066.75 M
4062.18 3063.28 L 4062.59 3063.07 L 4063.21 3062.46 L 4063.21
3066.75~~gge5~~0~#@$TRACK~0.3937~3~~4056.5525 3068.5039 4058.521
3068.5039~gge7~0#@$TRACK~0.3937~3~~4056.5525 3068.5039 4056.5525
3071.4961~gge8~0#@$TRACK~0.3937~3~~4056.5525 3071.4961 4058.521
3071.4961~gge9~0#@$TRACK~0.3937~3~~4059.3084 3071.4961 4061.2769
3071.4961~gge10~0#@$TRACK~0.3937~3~~4061.2769 3068.5039 4061.2769
3071.4961~gge11~0#@$TRACK~0.3937~3~~4059.3084 3068.5039 4061.2769
3068.5039~gge12~0#@$PAD~RECT~4059.997~3070~1.1811~1.5748~1~R1_2~2~0~4059.4065
3069.2126 4060.5876 3069.2126 4060.5876 3070.7874 4059.4065
3070.7874~0~gge13~0~~Y~0~0~0.4~4059.9969,3070#@$PAD~RECT~4057.832~3070~1.1811~1.
5748~1~R1_1~1~0~4057.2415 3069.2126 4058.4226 3069.2126 4058.4226 3070.7874
4057.2415 3070.7874~0~gge18~0~~Y~0~0~0.4~4057.8323,3070"
```

数据结构：

```txt
#1.[封装头部数据]
1.[cmdKey]：图元标识符，LIB
2.[x]：横坐标
3.[y]：纵坐标
4.[c_para]：自定义属性
5.[rotation]：旋转角度
6.[importFlag：eagle导入标记
7.[gId]：元素id
8.[layerid]：所属层
9.[uuid]：封装的uuid
10.[utime]：更新时间
11.[locked]：是否锁定
12.[bind_sch_id]：绑定的原理图器件的id（弃用）

#2.[封装内部图元数据]：即其他基本图元的压缩格式。
```

##### 图纸`SHEET`

原理图的图纸是一种特殊的器件，PCB中的图纸也可以理解为特殊的封装，但PCB使用的单独的图元来表示PCB图纸，所以图纸的结构也同封装结构一致，分为两部分：

- 图纸的头部数据：头部数据以分隔符`~`隔开。
- 图纸的内部图元数据：图元之间将会以分隔符`#@$`隔开。

两部分数据再以分隔符号`#@$`隔开。

数据格式：

```txt
#1.[图纸头部数据]
1.[cmdKey]：图元标识符，SHEET
2.[x]：横坐标
~~~

~~~~
3.[y]：纵坐标
4.[locked]：是否锁定
5.[layerid]：所属层
6.[gId]：元素id

#2.[图纸内部图元数据]：即其他基本图元的压缩格式。
```


# 原理图文件格式

一个原理图可以包含多个图页，每个图页都可以作为一份独立原理图文档，所以这里的原理图可以理解为**原
理图工程**，可以包含若干原理图页，在压缩格式中使用`schematics`字段来标识改工程下的所有原理图
页，类型为`JSON数组`。

示例如下：

```js
{
  "editorVersion": "6.3.0",     // 版本号
  "docType": "5",               // 文档类型
  "title": "New Project",       // 文档标题
  "description": "",            // 文档描述
  "colors": {},                 // 文档颜色配置（弃用）
  "schematics": [
    {
      "docType": "1",
      "title": "Sheet_1",
      "description": "",
      "dataStr": "{\"head\":
{\"docType\":\"1\",\"editorVersion\":\"6.3.0\",\"newgId\":true,\"c_para\":
{\"Prefix
Start\":\"1\"},\"c_spiceCmd\":null},\"canvas\":\"CA~1000~1000~#FFFFFF~yes~#CCCCC
C~5~1000~1000~line~5~pixel~5~0~0\",\"shape\":[\"LIB~0~-5~package`RES-ADJ-
TH_3386P`BOM_Supplier Part``BOM_Supplier`LCSC`BOM_Manufacturer
Part``nameAlias`Value`Contributor`LCEDA_Lib`BOM_value(Ω)`10K`spicePre`R`spiceSym
bolName`R_3386P_US`~~0~gge348eb5222b2937f0~ce6e0cc9b0684cd986edbf86d7075dfd~e0b0
d01b6e0d44dbab9db0ad79a427bc~0~~yes~yes#@$T~N~-9.15625~15.34375~0~#000080~Arial~
~~~~comment~R_3386P_US~1~start~gge103~0~#@$T~P~-9.15625~7~0~#000080~Arial~~~~~co
mment~RP1~1~start~gge105~0~#@$P~show~0~2~0~-25~90~gge107~0^^0~-25^^M 0 -25 v
10~#880000^^0~3~-13~270~2~end~~~#0000FF^^0~-1~-20~270~2~start~~~#0000FF^^0~0~-18
^^0~M -3 -15 L 0 -12 L 3 -15#@$E~11~-8~1~1~#A00000~1~0~none~gge114~0#@$PG~-2 -15
0 -11 2 -15~#880000~1~0~none~gge115~0#@$P~show~0~1~20~-5~0~gge116~0^^20~-5^^M 20
-5 h
-10~#8D2323^^0~7~-2~0~1~end~~~#8D2323^^0~14~-6~0~1~start~~~#8D2323^^0~13~-5^^0~M
10 -8 L 7 -5 L 10 -2#@$P~show~0~3~-20~-5~180~gge123~0^^-20~-5^^M -20 -5 h
10~#8D2323^^0~-6~-2~0~3~start~~~#8D2323^^0~-14~-6~0~3~end~~~#8D2323^^0~-13~-5^^0
~M -10 -2 L -7 -5 L -10 -8#@$PL~0 -15 0 -11~#880000~1~0~none~gge130~0#@$PL~9 -1
10 -5~#8D2323~1~0~none~gge131~0#@$PL~6 -9 9 -1~#8D2323~1~0~none~gge132~0#@$PL~4
-1 6 -9~#8D2323~1~0~none~gge133~0#@$PL~1 -9 4
-1~#8D2323~1~0~none~gge134~0#@$PL~-1 -1 1 -9~#8D2323~1~0~none~gge135~0#@$PL~-3
-9 -1 -1~#8D2323~1~0~none~gge136~0#@$PL~-6 -1 -3
-9~#8D2323~1~0~none~gge137~0#@$PL~-8 -9 -6 -1~#8D2323~1~0~none~gge138~0#@$PL~-10
-5 -8 -9~#8D2323~1~0~none~gge139~0\"],\"BBox\":
{\"x\":-22,\"y\":-27,\"width\":70.4,\"height\":43.7},\"colors\":{}}"
    },
    {
      "docType": "1",
      "title": "Sheet_2",
~~~~
~~~~

```
      "description": "",
      "dataStr": "{\"head\":
{\"docType\":\"1\",\"editorVersion\":\"6.3.0\",\"newgId\":true,\"c_para\":
{\"Prefix
Start\":\"1\"},\"c_spiceCmd\":null},\"canvas\":\"CA~1000~1000~#FFFFFF~yes~#CCCCC
C~5~1000~1000~line~5~pixel~5~0~0\",\"shape\":
[\"F~part_netLabel_netPort~15~0~0~gge78~~0^^15~0^^netPort1~#0000FF~-6.5~5~0~end~
1~Times New Roman~8pt~gge80^^PL~15 0 10 5 -5 5 -5 -5 10 -5 15
0~#0000FF~1~0~transparent~gge82~0\"],\"BBox\":
{\"x\":-44,\"y\":-5,\"width\":61,\"height\":12.7},\"colors\":{}}"
    }
  ]
}
```

# PCB文件格式

PCB不像原理图一样包含多页，所以PCB的文件格式和PCB库无结构上的差异。

示例：

```js
{
  "head": {
    "docType": "3",
    "editorVersion": "6.3.0",
    "newgId": true,
    "c_para": {},
    "hasIdFlag": true
  },
  "canvas":
"CA~1000~1000~#000000~yes~#FFFFFF~10~1000~1000~line~0.5~mil~1~45~~0.5~4087.5~330
8.5~0~yes",
  "shape": [
    "TRACK~1~1~~4076.5 3290.5 4076.5 3296.5 4115.5 3335.5~gge5~0",
    "PAD~ELLIPSE~4095~3304.5~6~6~11~~1~1.8~~0~gge6~0~~Y~0~~~4095,3304.5"
  ],
  "layers": [
    "1~TopLayer~#FF0000~true~true~true~",
    "2~BottomLayer~#0000FF~true~false~true~",
    "3~TopSilkLayer~#FFCC00~true~false~true~",
    "4~BottomSilkLayer~#66CC33~true~false~true~",
    "5~TopPasteMaskLayer~#808080~true~false~true~",
    "6~BottomPasteMaskLayer~#800000~true~false~true~",
    "7~TopSolderMaskLayer~#800080~true~false~true~0.3",
    "8~BottomSolderMaskLayer~#AA00FF~true~false~true~0.3",
  ],
  "objects": [
    "All~true~false",
    "Component~true~true",
    "Prefix~true~true",
    "Name~true~false",
    "Track~true~true",
    "Pad~true~true",
    "Via~true~true",
    "Hole~true~true",
    "Copper_Area~true~true",
    "Circle~true~true",
```

~~~~
    "Arc~true~true",
    "Solid_Region~true~true",
    "Text~true~true",
    "Image~true~true",
    "Rect~true~true",
    "Dimension~true~true",
    "Protractor~true~true"
  ],
  "BBox": {
    "x": 4076.5,
    "y": 3290.5,
    "width": 39,
    "height": 45
  },
  "preference": {
    "hideFootprints": "",
    "hideNets": ""
  },
  "DRCRULE": {
    "Default": {
      "trackWidth": 1,
      "clearance": 0.6,
      "viaHoleDiameter": 2.4,
      "viaHoleD": 1.2
    },
    "isRealtime": false,
    "isDrcOnRoutingOrPlaceVia": false,
    "checkObjectToCopperarea": true,
    "showDRCRangeLine": true
  },
  "netColors": {}
}
```

# 更新日志

暂无

-----------
##  Content
---



# Content

- [Common Information](https://docs.easyeda.com/en/DocumentFormat/1-Common-
Information/index.html)
    - [Example](https://docs.easyeda.com/en/DocumentFormat/1-Common-
Information/index.html)
        - [Schematic Example](https://docs.easyeda.com/en/DocumentFormat/1-
Common-Information/index.html)
        - [Schematic JSON File Source]
(https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
        - [PCB Example](https://docs.easyeda.com/en/DocumentFormat/1-Common-
Information/index.html)
        - [PCB JSON File Source](https://docs.easyeda.com/en/DocumentFormat/1-
Common-Information/index.html)
~~~~

```
    - [General File Struck](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
        - [Delimiter Mark](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
            - [~ (Tilde)](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
            - [`(Back Quote)](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
            - [^^(Double Circumflex)]
(https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
            - [#@$(Octothorpe Ampersat Dollar)]
(https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
    - [Document Type](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
    - [Canvas Coordinates](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
    - [SVG](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
    - [Q&A](https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
        - [1. How to check the json file format]
(https://docs.easyeda.com/en/DocumentFormat/1-Common-Information/index.html)
- [EasyEDA Schematic File Format](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
    - [Head](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Head information for schematic and subckt.]
(https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Head information for Schematic Symbol, Spice Symbol and Subpart]
(https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
    - [Canvas](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
    - [Shapes](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Rectangle](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Polyline](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Path](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Arc](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Pie](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Bus Entry](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Image](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Polygon](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Line](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Circle](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Bus](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Pin](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Ellipse](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Arrowhead](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Annotations](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Netlabels](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Netflags](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Wire](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Junctions](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [No Connect Flag](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [Symbol](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
    - [strokeStyle](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
    - [Q&A](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
        - [1. Why don't save the Wire, Annotion, netlabel, netflag to Shape field.](https://docs.easyeda.com/en/DocumentFormat/2-EasyEDA-Schematic-File-Format/index.html)
- [EasyEDA PCB File Format](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Head](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [Head information for PCB.](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [Head information for PCB Footprint](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Parameters Dimensions](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Canvas](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [System Color](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Layers config](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Preference](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [DRC Rule](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
    - [Shapes](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [Unit](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [TRACK](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [COPPERAREA](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [RECT](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [CIRCLE](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [SOLIDREGION](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [TEXT](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [Arc](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [PAD](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [VIA](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [HOLE](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [DIMENSION](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
        - [Footprint](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-Format/index.html)
- [EasyEDA Schematic File Object](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
    - [Rules](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [JSON Keys](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [itemOrder key](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
    - [Example](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [File](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [wire](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [Symbol](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
        - [JSON example](https://docs.easyeda.com/en/DocumentFormat/4-EasyEDA-Schematic-File-Object/index.html)
- [EasyEDA PCB File Object](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
    - [Rules](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [JSON Keys](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [itemOrder key](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
    - [Example](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [File](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [TRACK](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [SIGNALS](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [FOOTPRINT](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
        - [JSON example](https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-Object/index.html)
```

```
---

##  Common Information and Example
---

# Common Information

EasyEDA is a free, zero-install, Web and cloud-based EDA tool suite, integrating
powerful schematic capture, mixed-mode circuit simulation and PCB layout.

EasyEDA team tries to make our users happy. We provide an open ASCII file
format. With this file format, you can create a schematic or PCB using some
codes, even with Notepad. When you try to add hundreds of LEDs to a schematic or
PCB batch, you will find out that you can use codes to create an EasyEDA file,
then import it to EasyEDA. It is fun and quick.

EasyEDA's file is a JSON file, but we compress all of the shape's attributes to
a simple string, which will make the file size smaller and saving to server
faster. More importantly, with this solution you can create some very big
designs. Most browsers will crash when trying to decode the big JSON files. But
EasyEDA will provide an **API** to let you to access the EasyEDA friendly JSON
object, so you can hack the designs in the EasyEDA editor.

Ok, let's explain them with examples.

### Example

#### Schematic Example

[Schematic Example ](https://easyeda.com/file_view_simply-
schematic_puoGYgasK.htm)

![](../_images/easyeda-404_Format_schematic-example.png)


#### Schematic JSON File Source

check it via github gist [Schematic json]
(https://gist.github.com/dillonHe/0b62babdb8ab3d2ad7d3#file-schematic-json)
https://gist.github.com/dillonHe/0b62babdb8ab3d2ad7d3.js

#### PCB Example

[PCB Example](https://easyeda.com/guest/EasyEDA_File_Format-D6igsqZXL)
![](../_images/easyeda-395_Format_PCB-examle.png)

#### PCB JSON File Source

check it via github gist [PCB json]
(https://gist.github.com/dillonHe/279c55659bb7065258b6#file-pcb-json)
https://gist.github.com/dillonHe/279c55659bb7065258b6.js

### General File Struct

#### Delimiter Mark
```

```
From the above JSON source, you can find there are lots of **`**, **~**, **^^**
and **#@$**
 characters. These characters are used as **delimiter mark**. These characters
are not used frequently in design.

 ***Note: Although these characters were not picked properly at the very
beginning, we can't change these, EasyEDA already has lots of existing
designs***

##### ~ (Tilde)

 `~` is used to separate the attributes of the shapes. Taking rectangle as an
example.  `R~170~100~10~10~200~130~#99FF00~1~0~none~gge36~`, when use pure JSON
file, it should be look like below,
check it via github gist [rect json]
(https://gist.github.com/dillonHe/55151fae7c36785cdc31#file-rect-json).

https://gist.github.com/dillonHe/55151fae7c36785cdc31.js


So EasyEDA's source is small in file size and will transfer from the internet
faster.


##### `(Back Quote)

  **`** is used to separate the custom attributes.

**package`LED3MM** stands package:LED3MM

##### ^^(Double Circumflex)

**^^** is used to join segments, just used in *netFlag* , *Pin* and *pAD*.

##### #@$(Octothorpe Ampersat Dollar)

Union the characters **#@$** as a supper mark, it will be used to implode the
[shapes](./schematic.htm#Shapes) to a string, it's only used in *Symbol* and
*PCBLIB*.



### Document Type

check it via github gist [document type]
(https://gist.github.com/dillonHe/8c1a0e599540980bf7ab#file-document-type-js)
https://gist.github.com/dillonHe/8c1a0e599540980bf7ab.js

### Canvas Coordinates

The canvas is a two-dimensional grid.
The upper-left corner of the canvas has the coordinate (0,0)
![](../_images/easyeda-372_Format_coor.png)

### SVG
```

~~~
 EasyEDA uses [Scalable Vector Graphics (SVG)](http://en.wikipedia.org/wiki/SVG)
which is an XML-based vector image format for two-dimensional graphics to
realize the shapes.

### Q&A

#### 1. How to check the json file format
 Check the [EasyEDA source dialog](https://docs.easyeda.com/en/Export/Export-
EasyEDA-Source-File/index.html) out, copy the text to text area, then click the
`Apply` button. That is all.

---

##  EasyEDA Schematic File Format
---

# EasyEDA Schematic File Format

Note: Schematic, Schematic Symbol, Spice Symbol, Subpart and Subckt are used the
same file format. Please check [Schematic JSON File Source]
(https://docs.easyeda.com/en/Export/Export-EasyEDA-Source-File/index.html) out
before keeping read.

### Head

#### Head information for schematic and subckt.

```
    "head":"1~1.7.5~Author`Dillon`~TRAN`2u`2m`0`{AC`dec``0`0`{DC`0``0`0`{TF```"
```

**Format**:

1. [document type](https://docs.easyeda.com/en/DocumentFormat/1-Common-
Information/index.html) :`1`
2. document version: `1.7.5`h
3. custom attributes: **key: value** pairs, separate with **, added via **Add
Parameter**
![](../_images/easyeda-374_Format_customAttributes.png)
4. spice simulation configure store, Now can set four types `tran`, `AC`, `DC`,
`TF`, every type split with `{`. When opening the simulation dialog, these
information will be listed in like below image
![](../_images/easyeda-406_Format_simulation.png)


#### Head information for Schematic Symbol, Spice Symbol and Subpart

```


"head":"7~1.7.5~400~300~package`DIP08`nameDisplay`0`nameAlias`Model`Model`555`na
me`555`pre`U?`spicePre``Contributor`Dillon"
```

**Format**:

1. [document type](https://docs.easyeda.com/en/DocumentFormat/1-Common-
Information/index.html) :`7`
2. document version: `1.7.5`
~~~

~~~~
3. origin x position. **Reserved field, can't be changeded**
4. origin y position. **Reserved field, can't be changeded**
5. custom attributes: **key: value** pairs, separate with **`**, added via **Add new parameter**.

```
package: DIP08
nameDispaly: 0 (hide it is name when placed to schematic)
nameAlias: Model
name:555
pre:U? , when place to schematic, will be marked as U1, U2. subpart will be set as `U?.1`, `U?.2` etc.
spicePre:X, `X` stands for a subckt.
sourceId:xxxxxxxxx (just for schematic Lib and spice symbol)
```

Place it to schematic canvas, it's attributes will be looked like below image. The name field is alias as Model and it is invisible.

![](../_images/easyeda-405_Format_schematicLibAttr.png)

### Canvas
```

"canvas":"CA~1200~1200~#FFFFFF~yes~#CCCCCC~10~1200~1200~line~10~pixel~5~400~300"
```

**Format**:
1. command: CA
2. view box width: 1200, View Box Width / Canvas width = scaleX
3. view box height: 1200,View Box Height / Canvas Height = scaleY
4. back ground: #FFFFFF
5. grid visible: yes/none
6. grid color: #CCCCCC
7. grid size: 10 pixel
8. canvas width: 1200 pixel
9. canvas height: 1200 pixel
10. grid style: line/dot
11. snap size: 10 pixel
12. unit: pixel(Always pixel)
13. ALT snap size:5 (Snap Size when pressing the `ALT` Key)
14. origin x position
15. origin y position

Canvas setting image

![](../_images/easyeda-403_Format_SchematicCanvas.png)

### Shapes
The shape is an array. EasyEDA store various shape in this field, they are different with a command which locate at the begin of the string.
```
    "shape":[
        "PL~210 100 260 100~#000000~2~0~none~gge58",
        "R~210~110~~~50~30~#000000~1~0~none~gge61",
        "I~90~90~271~105~0~https://easyeda.com/assets/static/images/logo-140x39.png~gge62",
        "PG~310 100 350 130 300 150 290 150 270 120~#000000~2~0~none~gge64",
~~~~

~~~~
        "PT~M230 170 C270 200 270 170 240 150 240 150 240 150 240
150~#000000~2~0~none~gge65"
    ]
```

#### Rectangle
```
    "R~650~0~20~20~230~160~#FF0000~2~1~#9966FF~gge5"
```
**Format**:

Check [ Rect element of SVG](http://www.w3.org/TR/SVG11/shapes.html#RectElement)
out.

1. command: R
2. x: 650
3. y: 0
4. rx: 20
5. ry: 20
6. width: 230
7. height: 160
8. strokeColor: #FF0000
9. strokeWidth: 2 //pixel
10. [strokeStyle](#strokeStyle): 1
11. fillColor: #9966FF
12. id: gge36
13. locked:null
Rect's attributes and image looks like bellow image:
![](../_images/easyeda-402_Format_rect.png)

#### Polyline
    "PL~610 130 780 130~#FF0000~5~0~none~gge6"

**Format**:
Check [ Polyline element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#PolylineElement) out.

1. command: PL
2. points: 610 130 780 130
3. strokeColor: #FF0000
4. strokeWidth: 5 //pixel
5. [strokeStyle](#strokeStyle): 0
6. fillColor: none
7. id: gge6
8. locked:null
Polyline's attributes and image looks like bellow image:
![](../_images/easyeda-401_Format_polyline.png)

#### Path
    "PT~M670 300 C830 370 850 230 920 300 920 300 920 300 920
300~#000000~1~0~none~gge17"
**Format**:
Check [Path element of SVG](http://www.w3.org/TR/SVG11/paths.html#PathElement)
out.

1. command: PT
2. pathString:M670 300 C830 370 850 230 920 300 920 300 920 300 920 300
3. strokeColor: #FF0000
~~~~

~~~
4. strokeWidth: 5 //pixel
5. [strokeStyle](#strokeStyle): 0
6. fillColor: none
7. id: gge6
8. locked:null
Path's attributes and image looks like bellow image:
![](../_images/easyeda-392_Format_path.png)

**bezier** is a **path** too.

#### Arc

```"A~M 1020 60 A 80 80 0 0 1 953.096
199.904~968.78,121.45,1048.785,201.457,1018.785,61.457,948.785,221.45~#FF0000~3~
0~none~gge19"```

**Format**:
**Arc** is a **Path** element, Check [Path element of SVG]
(http://www.w3.org/TR/SVG11/paths.html#PathElement) out.

1. command: A
2. pathString:M670 300 C830 370 850 230 920 300 920 300 920 300 920 300
3. helperDots: the four green dots
4. strokeColor: #FF0000
5. strokeWidth: 3 //pixel
6. [strokeStyle](#strokeStyle): 0
7. fillColor: none
8. id: gge19
9. locked:null


ARC's attributes and image looks like bellow image:
![](../_images/easyeda-382_FileFomat_Arc.png)

#### Pie
    "PI~M 970 40 L 1189.9 34.4509 A 220 180 0 0 1 923.103 215.863
Z~970,40,1190,220,1327.7106323242188,30.973068237304688,923.1032104492188,215.86
282348632812~#FF0000~3~0~#CCCCCC~gge22"
**Pie** is a **Path** element, Check [Path element of SVG]
(http://www.w3.org/TR/SVG11/paths.html#PathElement) out. Pie is similar with
Arc, the pathString has a `Z`

1. command: PI
2. pathString:M 970 40 L 1189.9 34.4509 A 220 180 0 0 1 923.103 215.863 Z
3. helperDots: the four green dots
4. strokeColor: #FF0000
5. strokeWidth: 3 //pixel
6. [strokeStyle](#strokeStyle): 0
7. fillColor: none
8. id: gge19
9. locked:null


Pie's attributes and image looks like bellow image:

![picture 4](../_images/2-EasyEDA-Schematic-File-Format-20200804064939.png)
~~~

```
#### Bus Entry
    "BE~0~660~150~670~140~gge15"

**Format**:

1. command: BE
2. rotation:0
3. start x1: 660
4. start y1: 150
5. end x1: 670
6. end y1: 140
7. id: gge15
8. locked:null

Bus Entry's attributes and image looks like bellow image:

![picture 5](../_images/2-EasyEDA-Schematic-File-Format-20200804065008.png)


#### Image
    "I~610~10~271~105~0~https://easyeda.com/assets/static/images/logo-
140x39.png~gge12"

**Format**:
Check [ Image element of SVG]
(http://www.w3.org/TR/SVG11/struct.html#ImageElement) out.

1. command: I
2. x: 610
3. y: 10
4. width: 271
5. height: 105
6. rotation: 0
7. href:https://easyeda.com/assets/static/images/logo-140x39.png
8. id: gge12
9. locked:null
Image's attributes and image looks like bellow image:

![picture 6](../_images/2-EasyEDA-Schematic-File-Format-20200804065036.png)



#### Polygon
    "PG~640 10 900 40 920 140 760 230 560 140~#FF0000~2~0~#00FF00~gge10"

**Format**:
Check [ Polygon element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#PolygonElement) out.

1. command: PG
2. points: 640 10 900 40 920 140 760 230 560 140
3. strokeColor: #FF0000
4. strokeWidth: 2 //pixel
5. [strokeStyle](#strokeStyle): 0
6. fillColor: #00FF00
7. id: gge10
8. locked:null
Polygon's attributes and image looks like bellow image:
```

```
![picture 7](../_images/2-EasyEDA-Schematic-File-Format-20200804065104.png)



#### Line
    "L~360~160~510~160~#FF0000~2~0~none~gge11"

**Format**:
Check [ Line element of SVG](http://www.w3.org/TR/SVG11/shapes.html#LineElement)
out.

1. command: L
2. x1:360
3. y1:160
4. x2:510
5. y2:160
6. strokeColor: #FF0000
7. strokeWidth: 2 //pixel
8. [strokeStyle](#strokeStyle): 0
9. fillColor: #00FF00
10. id: gge11
11. locked:null

#### Circle
    "C~710~170~105~#FF0000~2~0~#0000FF~gge12"
**Format**:
Check [ Circle  element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#CircleElement) out.

1. command: C
2. cx:720
3. cy:90
4. r:105
5. strokeColor: #FF0000
6. strokeWidth: 2 //pixel
7. [strokeStyle](#strokeStyle): 0
8. fillColor: #0000FF
9. id: gge12
10. locked:null


#### Bus
    "B~570 130 680 130 680 210~#008800~2~0~none~gge19"

Bus is similar with [Polyline](#polyline), Bus is start with `B`, polyline start
with `PL`.

#### Pin
    "P~show~0~1~670~30~~gge23^^670~30^^M 670 30 h
-20~#880000^^1~648~33~0~1~end~~11pt^^1~655~29~0~1~start~~11pt^^0~653~30^^0~M 650
27 L 647 30 L 650 33"

A Pin has seven segments, join these segments with [^^ (Double Circumflex)]
(./common.htm#Double-Circumflex) as a string like above.

1. **Pin configure** `P~show~0~1~670~30~~gge23`
    1. command: P
    2. display: show/'' (*bad design, should use yes/none*)
    3. electric: 0, can be  ['Undefined', 'Input','Output','I/O','Power']
    4. spice pin number: 1
    5. position x: 670
    6. position y: 30
    7. rotation: null, can be ['null' or 0, '90', '180', '270']
    8. id: gge23
    9. locked: null
2. **pin dot** `670~30`
    The gray dot at the end of the Pin, it is important.
    1. pin dot x: 670
    2. pin dot y: 30

3. **pin path** `M 670 30 h -20~#880000`
    1. path: M 670 30 h -20, a 20 pixel horizontal line start from **pin dot**
    2. pin color: #880000

4. **name** `1~648~33~0~1~end~~11pt`
    1. visible : 1/0 stand show or hide
    2. position x: 648
    3. position y: 33
    4. rotation: 0
    5. text: **1**
    6. text anchor: end
    7. font family: null, default is **Verdana**
    8. font size: 11pt, default is 7pt
5. **number** `1~655~29~0~1~start~~11pt`

     the same as **name** above
6. **dot** `0~653~30`

    stands for not. a circle with radius in 3 pixel
    1. visible : 0/1 hide / show
    2. circle x: 653
    3. circle y: 30
7. **clock** `0~M 650 27 L 647 30 L 650 33`
    1. visible: 0/1 hide / show
    2. clock path: M 650 27 L 647 30 L 650 33

Pin's attributes and image looks like bellow image:
![](../_images/easyeda-399_Format_pin.png)


#### Ellipse
    "E~720~90~105~65~#FF0000~2~0~#0000FF~gge12"

**Format**:
Check [ Ellipse  element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#EllipseElement) out.

1. command: E
2. cx:720
3. cy:90
4. rx:105
5. ry:65
6. strokeColor: #FF0000
7. strokeWidth: 2 //pixel
8. [strokeStyle](#strokeStyle): 0
```

```
9. fillColor: #0000FF
10. id: gge12
11. locked:null

Ellipse's attributes and image looks like bellow image:
![](../_images/easyeda-378_Format_ellipse.png)

#### Arrowhead

    "AR~part_arrowhead~1060~120~gge23~180~M 1060 120 L 1063 126 L 1055 120 L
1063 114 Z~#FF0000"

**Format**:

1. command: AR
2. part Type:part_arrowhead, not used
3. x:1060
4. y:120
5. id:gge23
6. rotation: 180
7. path String: M 1060 120 L 1063 126 L 1055 120 L 1063 114 Z
9. fillColor: #FF0000
9. locked:null

Arrow head's attributes and image looks like bellow image:
![](../_images/easyeda-383_FileFomat_Arrow.png)

#### Annotations
    "T~L~540~60~0~#0000FF~~9pt~bold~normal~~comment~Text~1~start~gge26"

Check [Text  element of SVG](http://www.w3.org/TR/SVG11/text.html#TextElement)
out.
**Format**:

1. command: T
2. mark: L // `L` = label, `N` = Name, `P` = prefix `N,P` are for [Symbol]
(#Symbol)
3. position x:540
4. position y:60
5. rotation:0
6. fill color: #0000FF
7. font family: null, default is **Verdana**
8. font size: 9pt
9. font-weight: bold
10. font style: normal
11. dominant baseline: null
12. text type: comment // **comment** or **spice** command
13. string: Text
14. visible: 1/0 show/hide (use for mark `N` or `P` )
15. text anchor: start (start middle end)
16. id:gge26
17. locked:null

Text's attributes and image looks like bellow image:
![](../_images/easyeda-415_Format_text.png)


#### Netlabels
```

```
        "N~360~100~0~#FF0000~VCC~gge32~start~362~100~Times New Roman~",

**Format**:

1. command: N
2. pin dot x: 360
3. pin dot y: 100
4. rotation: 0
5. fill color: #FF0000
6. name: VCC
7. id: gge32
8. text anchor: start (start middle end)
9. postion x: 362
10. postion y: 100
11. font family: Times New Roman
12. font size:null default is 7pt
13. locked:null


netlabel's attributes and image looks like bellow image:
![](../_images/easyeda-388_Format_netlabel.png)

#### Netflags
Netflag is very similar with netlabel


"F~part_netLabel_gnD~330~110~~gge41^^330~110^^GND~#000080~319~97~0~start~0~Times
New Roman~9pt^^PL~330 120 330 110~#000000~1~0~none~gge44^^PL~320 120 339
120~#000000~1~0~none~gge45^^PL~324 122 337 122~#000000~1~0~none~gge46^^PL~326
124 333 124~#000000~1~0~none~gge47^^PL~329 126 331 126~#000000~1~0~none~gge48",

A Netflag  has several segments, join these segments with [^^(Double
Circumflex)](./common.htm#Double-Circumflex) as a string like above.

1. **configure** `P~show~0~1~670~30~~gge23`
    1. command: F
    2. part id: part\_netLabel\_gnD
    3. position x: 330
    4. position y: 110
    5. rotation: null [0, 90, 180, 270]
    6. id: gge41,
    7. locked: null

2. **pin dot** `670~30`
    The gray dot at the end of the Pin, it is important.
    1. pin dot x: 330
    2. pin dot y: 140

3. **mark string** `GND~#000080~319~97~0~start~0~Times New Roman~9pt`
    1. net flag string: GND
    2. color: #000080
    3. position x: 319
    4. position y: 97
    5. rotation: 0 [0, 90, 180, 270]
    6. text anchor: start (start middle end)
    7. visible: 1/0 show/hide the net flag string
    8. font family: Times New Roman
    9. font size:null default is 7pt
4. **shapes**

    All other items are [shapes](#shapes).

netflag's attributes and image looks like bellow image:
![](../_images/easyeda-387_Format_netflag.png)

#### Wire

        "W~570 130 680 130 680 210~#008800~2~0~none~gge19"


`Wire` is similar with [Polyline](#polyline), `Wire` is start with `W`,
[polyline](#polyline) start with `PL`.

#### Junctions

        "J~420~140~2.5~#CC0000~gge18",


**Format**:

1. command: J
2. pin dot x: 420
3. pin dot y: 140
4. junction circle radius: 2.5 pixel
5. fill color: #CC0000
6. id: gge18
7. locked:null


Junction's attributes and image looks like bellow image:
![](../_images/easyeda-385_Format_junction.png)

#### No Connect Flag

        "O~960~410~gge5~M956,406 L964,414 M964,406 L956,414~#FF0000"


**Format**:

1. command: O
2. pin dot x: 960
3. pin dot y: 410
4. id: gge5
5. pathStr: M956,406 L964,414 M964,406 L956,414
6. color: #FF0000
7. locked:null


No Connect Flag's attributes and image looks like bellow image:

![picture 8](../_images/2-EasyEDA-Schematic-File-Format-20200804065137.png)



#### Symbol
```

```
"LIB~220~140~package`C1`nameAlias`Value(F)`Value(F)`1u`spicePre`C`spiceSymbolNam
e`Capacitor`~~0~gge66#@$T~N~214~129~0~#000080~Arial~~~~~comment~1u~1~start~gge68
#@$T~P~214~120~0~#000080~Arial~~~~~comment~C1~1~start~gge69#@$PL~218 148 218
132~#A00000~1~0~none~gge70#@$P~show~0~1~200~120~180~gge71^^200~140^^M 210 140 h
-10~#800^^0~214~140~0~1~start~~^^0~206~136~0~1~end~~^^^^#@$PL~230 140 222
140~#A00000~1~0~none~gge72#@$PL~222 132 222
148~#A00000~1~0~none~gge73#@$P~show~0~2~210~120~0~gge74^^240~140^^M 230 140 h
10~#800^^0~226~140~0~2~end~~^^0~234~136~0~2~start~~^^^^#@$PL~218 140 210
140~#A00000~1~0~none~gge75"

A Symbol  has several shapes, join these shapes with [#@$(Octothorpe Ampersat
Dollar) ](./common.htm#Octothorpe-Ampersat-Dollar) as a string like above.

1. **configure** <code>LIB~270~140~package\`DO35-
7\`nameAlias\`Model\`Model\`1N4001\`spicePre\`D\`spiceSymbolName\`Diode\`~~0~gge
116</code>

    1. command: LIB
    2. position x: 270
    3. position y: 140
    4. [custom attributes]: *package\`DO35-
7\`nameAlias\`Model\`Model\`1N4001\`spicePre\`D\`spiceSymbolName\`Diode\`*
    5. rotation: 0, can be ['null' or 0, '90', '180', '270']
    6. import flag: 0 just  used in import from eagle
    7. id: gge116
    8. locked: null

2. **shapes**

    All other items are [shapes](#shapes).


### strokeStyle
- 0 : solid
- 1 : dashed
- 2: dotted

### Q&A


#### 1. Why don't save the Wire, Annotion, netlabel, netflag to Shape field.
These items will be used to create netlist, save them to separate field will
make you spent more less time to do this. We don't need to traversal all the
shapes.



---

##  EasyEDA PCB File Format
---

# EasyEDA PCB File Format
Note: PCB and PCB Footprint are used the same file format. Please check [PCB
JSON File Source](https://docs.easyeda.com/en/DocumentFormat/3-EasyEDA-PCB-File-
Format/index.html) out before keeping read.
```

~~~~
### Head

#### Head information for PCB.

    "head":"3~1.7.5~Author`Dillon`"

**Format**:

1. [document type](https://gist.github.com/dillonHe/8c1a0e599540980bf7ab#file-
document-type-js) :`3`
2. document version: `1.7.5`
3. custom attributes: **key: value** pairs, separate with **`**, added via **Add
new parameter**
  ![](../_images/easyeda-374_Format_customAttributes.png)



#### Head information for PCB Footprint
    "4~1.7.5~400~300~`pre`U?`Contributor`Dillon"


**Format**:

1. [document type]() :`4`
2. document version: `1.7.5`
3. origin x position. **Reserved field, can't be changeded**
4. origin y position. **Reserved field, can't be changeded**
5. custom attributes: **key: value** pairs, separate with **`**, added via **Add
new parameter**.

```
pre:U? , when place to PCB, will be marked as U1, U2.
Contributor:Dillon
sourceId:xxxxxxxxx (just for Footprint)
```

### Parameters Dimensions

EasyEDA support millimeter, inch and millimeter, but when these items are stored
to a file, all of them will be expressed as 10X mil. Taking line lengths or
widths for examples, stroke width equal 1, stands 10mil.



### Canvas

"CA~2400~2400~#000000~yes~#FFFFFF~10~1200~1200~line~1~mil~1~45~visible~0.5~400~3
00"

**Format**:

1. command: CA
2. view box width: 2400(24000 mil), View Box Width / Canvas width = scaleX = 2
3. view box height: 2400(24000 mil),View Box Height / Canvas Height = scaleY = 2
4. back ground: #000000
5. grid visible: yes/none
6. grid color: #FFFFFF
~~~~

~~~
7. grid size: 10(100 mil)
8. canvas width: 1200 (12000 mil)
9. canvas height: 1200 (12000 mil)
10. grid style: line/dot
11. snap size: 1 (10 mil)
12. unit: mil(inch, mil, mm)
13. routing width: 1 (10mil)
14. routing angle: 45 degree( 45 90 free)
15. copper area: visible/invisible
16. ALT snap size: 0.5 ( 5 mil Snap Size when pressing the `ALT` Key)
17. origin x position
18. origin y position

Canvas setting image

![](../_images/easyeda-393_Format_PCBCanvas.png)

### System Color
    "#000000~#FFFFFF~#FFFFFF~#000000~#FFFFFF"


**Format**:

1. future use: #000000
2. future use: #FFFFFF
3. future use: #FFFFFF
4. hole Color: #000000
5. DRC error: #FFFFFF

### Layers config
layers is an array, each layer is an item of the layers.
```
     "layers":[
        "1~TopLayer~#FF0000~true~true~true",
        "2~BottomLayer~#0000FF~true~false~true",
        "3~TopSilkLayer~#FFFF00~true~false~true",
        "4~BottomSilkLayer~#808000~true~false~true",
        "5~TopPasterLayer~#808080~false~false~false",
        "6~BottomPasterLayer~#800000~false~false~false",
        "7~TopSolderLayer~#800080~false~false~false",
        "8~BottomSolderLayer~#AA00FF~false~false~false",
        "9~Ratlines~#6464FF~true~false~true",
        "10~BoardOutline~#FF00FF~true~false~true",
        "11~Multi-Layer~#C0C0C0~true~false~true",
        "12~Document~#FFFFFF~true~false~true",
        "21~Inner1~#800000~false~false~false",
        "22~Inner2~#008000~false~false~false",
        "23~Inner3~#00FF00~false~false~false",
        "24~Inner4~#000080~false~false~false"
    ]
```
**Format**:

1. layer id: 1
2. layer name: TopLayer
3. layer color: #FF0000
4. visible: true, hints the objects in this layer show or hide
5. active: false. active layer
~~~

~~~
6. config: true. if be set false, you can't see it on the layer toolbar.


### Preference
```
    "preference":{
        "hideFootprints":"gge118~gge221~gge227~gge233",
        "hideNets":"BSYNC~DREQ~GPIO0~MICP~GND"
    }
```
`hideFootprints` : when the id of the footprints in here, you can't see them on
canvas.
`hideNets` : when the net name in here, you can't see them on canvas, you can
hide the ratline at here too. There are some guys would like to hide then GND
ratline, then use copper area to connect all the GND pad.

### DRC Rule
```
     "DRCRULE":{
        "trackWidth":0.7,
        "track2Track":0.7,
        "pad2Pad":0.8,
        "track2Pad":0.8,
        "hole2Hole":1,
        "holeSize":1.6
    }
```
`trackWidth`: 0.7 (7 mil)  track width
`track2Track`: 0.7 (7 mil) track to track distance
`pad2Pad`:   0.8(8 mil) pad to pad distance
`track2Pad`: 0.8(8 mil) track to pad distance
`hole2Hole`: 1(10 mil) hole to hole distance
`holeSize`:  1.6(16 mil) hole diameter

This is a simple DRC, more later.

### Shapes

The shape is an array. EasyEDA store various shape in this field, they are
different with a command which locate at the begin of the string.
```
    "shape":[
        "TRACK~1~1~S$19~311 175 351 175 352 174~gge18",
        "PAD~ELLIPSE~329~185~6~6~11~~1~1.8~~0~gge20",
        "VIA~329~202~3.2~~0.8~gge23",
        "COPPERAREA~2px~1~GND~349 247 492 261 457 314 339
329~1~solid~gge27~spoke~none~[]",
        "SOLIDREGION~1~~350 146 483 146 447 228 371 220~solid~gge26"
    ]
```
#### Unit
 EasyEDA takes **10 mil** as a basic factor, when a stroke width is 1, we can
take it as 1\*10mil = 10mil, is 2, we can take it as 2\*10mil = 20mil



#### TRACK
    "TRACK~1~1~S$8~311 175 351 175 352 174~gge18"
~~~

~~~
**Format**:
Check [ Polyline element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#PolylineElement) out.

1. command: TRACK
2. stroke Width: 1 (10 mil)
3. layer id: 1 (TopLayer)
4. net: "S$8"
5. points: 311 175 351 175 352 174
6. id : gge18
7. locked: null

TRACK's attributes and image looks like bellow image:
![](../_images/easyeda-416_Format_Track.png)


#### COPPERAREA
```
    "COPPERAREA~2px~1~GND~349 247 492 261 457 314 339
329~1~solid~gge27~spoke~yes~[[\"M339,329 349,247 492,261 457,314z\"]]"
```
**Format**:



1. command: COPPERAREA
2. stroke Width: 2 (20 mil)
3. layer id: 1 (TopLayer)
4. net: GND
5. points: 349 247 492 261 457 314 339 329
6. clearance Width : 1 (10 mil)
7. fill style: solid/none
8. id: gge27
9. thermal: spoke/direct
10. keep island: none/yes
11. copper zone: [[\"M339,329 349,247 492,261 457,314z\"]] rings and holes
12. locked: null

COPPERAREA's attributes and image looks like bellow image:
![](../_images/easyeda-373_Format_Copper-Area.png)


#### RECT
    "RECT~406~220~105~52~1~gge32"

**Format**:

Check [ Rect element of SVG](http://www.w3.org/TR/SVG11/shapes.html#RectElement)
out.

1. command: RECT
2. x: 406
3. y: 220
4. width: 105
5. height: 52
6. layer id:1
7. id: gge36
8. locked:null
Rect's attributes and image looks like bellow image:
~~~

```
![](../_images/easyeda-397_Format_PCB402_Format_rect.png)


#### CIRCLE
    "CIRCLE~363~273~42~1~3~gge33"


**Format**:
Check [ Circle element of SVG]
(http://www.w3.org/TR/SVG11/shapes.html#CircleElement) out.

1. command: CIRCLE
2. cx:363 (3630 mil)
3. cy:273
4. r:42 (420 mil)
5. stroke width: 1 (10mil)
6. layer id: 3 (Top silk layer)
7. id: gge33
8. locked:null

CIRCLE's attributes and image looks like bellow image:
![](../_images/easyeda-394_Format_PCBcircle.png)

#### SOLIDREGION
    "SOLIDREGION~1~GND~322 256 376 317 447 250 353 231~solid~gge34"


**Format**:

1. command: SOLIDREGION
2. layer id: 1 (Toplayer)
3. net: GND
4. points:322 256 376 317 447 250 353 231
5. type: solid/cutout/npth
6. id: gge34
7. locked:null

SOLIDREGION's attributes and image looks like bellow image:
![](../_images/easyeda-407_Format_SolidRegion.png)


#### TEXT
    "TEXT~L~351~252~0.8~0~none~1~~8~TEXT~M 352.55 250.64 L 352.55 258.27 M 350
250.64 L 355.09 250.64 M 357.49 250.64 L 357.49 258.27 M 357.49 250.64 L 362.22
250.64 M 357.49 254.27 L 360.4 254.27 M 357.49 258.27 L 362.22 258.27 M 364.62
250.64 L 369.71 258.27 M 369.71 250.64 L 364.62 258.27 M 374.65 250.64 L 374.65
258.27 M 372.11 250.64 L 377.2 250.64~~gge35"


**Format**:

1. command: TEXT
2. type: L/P (L = label, P = prefix)
3. position x: 351 (3510 mil)
4. position y: 252 (2520 mil)
5. stroke width: 0.8 (8 mil)
6. rotation: 0
7. mirror : none ( not user now)
```

```
8. layer id: 1 (Toplayer)
9. net: ''
10. font size: 8 (80 mil in height)
11. string: TEXT
12. text path: M 352.55 250.64 L 352.55 258.27 M 350 250.64 L 355.09 250.64 M
357.49 250.64 L 357.49 258.27 M 357.49 250.64 L 362.22 250.64 M 357.49 254.27 L
360.4 254.27 M 357.49 258.27 L 362.22 258.27 M 364.62 250.64 L 369.71 258.27 M
369.71 250.64 L 364.62 258.27 M 374.65 250.64 L 374.65 258.27 M 372.11 250.64 L
377.2 250.64
13. display: '' (none = hide, other = show)
14. id: gge35
15. locked: null


TEXT's attributes and image looks like bellow image:
![](../_images/easyeda-397_Format_PCBtext.png)

#### Arc

    "ARC~1~1~S$19~M329,274 A26.95,26.95 0 0 1 370,309~~gge50"

**Format**:
**Arc** is a **Path** element, Check [Path element of SVG]
(http://www.w3.org/TR/SVG11/paths.html#PathElement) out.

1. command: ARC
2. stroke width: 1 (10 mil)
3. layer id: 1 (Toplayer)
4. net: S$19
5. path string: M329,274 A26.95,26.95 0 0 1 370,309
6. helper dots: the four green dots, no need in PCB, keep it blank
7. id: gge19
8. locked:null


ARC's attributes and image looks like bellow image:
![](../_images/easyeda-PCB382_FileFomat_Arc.png)

#### PAD

    "PAD~OVAL~814~371~6~16~11~~1~1.8~814 366 814 376~0~gge5~11~814 374.7 814
367.3~N"

**Format**:

1. command: PAD
2. shape: ELLIPSE/RECT/OVAL/POLYGON
3. center x: 814
4. center y: 371
5. width: 6 (60 mil)
6. height: 16 (160 mil)
7. layer id: 11 (All)
8. net: ''
9. number: 1
10. hole radius: 1.8 (18 mil)
11. points: '' (ELLIPSE = '', RECT = outline points)
12. rotation: 0 [0 - 360]
13. id: gge19
```

~~~
14. Hole(Length): 11 (110mil)
15. Hole Points: 814 374.7 814 367.3 // slot hole from to point
16. Plated:Y/N
14. locked:null

PAD's attributes and image looks like bellow image:
![](../_images/easyeda-390_Format_PAD.png)



#### VIA

    "VIA~432~215~3.2~~0.8~gge5"

**Format**:

1. command: VIA
2. center x: 432
3. center y: 215
4. diameter: 3.2
5. net : ''
6. hole radius: 0.8 (8 mil)
7. id: gge5
8. locked:null


VIA's attributes and image looks like bellow image:
![](../_images/easyeda-417_Format_VIA.png)

#### HOLE
    "HOLE~284~255~4~gge5"

**Format**:

1. command: HOLE
2. center x: 284
3. center y: 255
4. diameter: 4
5. id: gge5
6. locked:null


HOLE's attributes and image looks like bellow image:
![](../_images/easyeda-382_Format_Hole.png)

#### DIMENSION
```
    "DIMENSION~3~M 301 217 L 442 217 M 306 220 L 301 217 L 306 214 M 437 220 L
442 217 L 437 214 M 369.5 209.82 L 370.05 209.55 L 370.86 208.73 L 370.86 214.45
M 372.94 213.09 L 372.66 213.36 L 372.94 213.64 L 373.21 213.36 L 372.94 213.09
M 377.74 208.73 L 375.01 212.55 L 379.1 212.55 M 377.74 208.73 L 377.74 214.45 M
380.9 209.82 L 381.45 209.55 L 382.26 208.73 L 382.26 214.45 M 384.06 208.73 L
384.06 210.64 M 386.25 208.73 L 386.25 210.64~gge8"
```
**Format**:

1. command: DIMENSION
2. layer id: 3 (Top Silk layer)
~~~

~~~
3. path: M 301 217 L 442 217 M 306 220 L 301 217 L 306 214 M 437 220 L 442 217 L
437 214 M 369.5 209.82 L 370.05 209.55 L 370.86 208.73 L 370.86 214.45 M 372.94
213.09 L 372.66 213.36 L 372.94 213.64 L 373.21 213.36 L 372.94 213.09 M 377.74
208.73 L 375.01 212.55 L 379.1 212.55 M 377.74 208.73 L 377.74 214.45 M 380.9
209.82 L 381.45 209.55 L 382.26 208.73 L 382.26 214.45 M 384.06 208.73 L 384.06
210.64 M 386.25 208.73 L 386.25 210.64
4. id: gge5
5. locked:null


DIMENSION's attributes and image looks like bellow image:
![](../_images/easyeda-376_Format_Dimension.png)
DIMENSION just allows to change it layer id, if you don't accept this DIMENSION,
delete it and redraw again.

#### Footprint
```
    "LIB~245~240~package`CK17-B`~~~gge15~1#@$TEXT~P~295~219.5~0.7~0~~3~~4.5~C1~M
298.07 218.07L297.86 217.66 L297.45 217.25 L297.05 217.05 L296.23 217.05 L295.82
217.25 L295.41 217.66 L295.2 218.07 L295 218.68 L295 219.7 L295.2 220.32 L295.41
220.73 L295.82 221.14 L296.23 221.34 L297.05 221.34 L297.45 221.14 L297.86
220.73 L298.07 220.32 M 299.42 217.86L299.83 217.66 L300.44 217.05 L300.44
221.34 ~~gge16#@$TRACK~0.9~3~~257.5 224.5 332.5 224.5 332.5 255.5 257.5 255.5
257.5
224.5~gge17#@$PAD~ELLIPSE~245~240~9.4~9.4~11~~1~2.25~~0~gge18#@$PAD~ELLIPSE~345~
240~9.4~9.4~11~~2~2.25~~0~gge19"
```

A Footprint has several shapes, join these shapes with [#@$(Octothorpe Ampersat
Dollar) ]()as a string like above.

1. **configure** <code>LIB~245~240~package\`CK17-B\`~0~~gge15~1</code>
    1. command: LIB
    2. position x: 270
    3. position y: 140
    4. [custom attributes](common.htm#Back-Quote): *package\`CK17-B\`*
    5. rotation: 0, can be [0 - 360 ]
    6. import flag: '', just used in import from eagle
    7. id: gge115
    8. locked: null

2. **shapes**

    All other items are [shapes](#shapes).

Footprints' image looks like bellow image:
![](../_images/easyeda-396_Format_PCBlibs.png)

---

##  EasyEDA Schematic File Object
---

# EasyEDA Schematic File Object

Note: Schematic, Schematic Symbol, Spice Symbol, Subpart and Subckt use the same
format.
~~~

~~~~
*EasyEDA Schematic File Object* is a JSON Object which allows you to hack your
designs via other languages, such as Javascript, Python, PHP, C, C++. The
interesting thing is that your can control/modify your design in EasyEDA editor
via Javascript language.

### Rules

#### JSON Keys
 Every EasyEDA graph unit has an unique key, such as  "**wire**", "**Symbol**",
"**junction**", "**bus**", "**busentry**", "**netlabel**", "**netflag**",
"**pin**", "**polyline**", "**path**", "**arc**", "**rect**", "**polygon**",
"**arrowhead**", "**ellipse**", "**image**"

#### itemOrder key
   Because an object in EasyEDA  is an unordered set of name/value pairs in
[JSON format](http://json.org/), but EasyEDA's graphs are ordered. We need an
array to store the order of these objects. Every schematic lib has an itemOrder
key and the whole JSON object has an itemOrder key.

### Example

#### File

![](../_images/easyeda-379_Format_fileJson.png)
Open [Schematic Example ](https://easyeda.com/guest/EasyEDA_File_Format-
D6igsqZXL)

#### wire
```
    "wire":{
        "gge48":{
            "gId":"gge48",
            "strokeColor":"#008800",
            "strokeWidth":"1",
            "strokeStyle":0,
            "fillColor":"none",
            "pointArr":[
                {
                    "x":290,
                    "y":430
                },
                {
                    "x":370,
                    "y":430
                },
                {
                    "x":370,
                    "y":490
                }
            ]
        }
        ..........
    }
```
All wires will be stored to **wire** key, their id will be taken as the key such
as `gge48`.

#### Symbol
~~~~

~~~
  All schematic components will be stored to **Symbol**, their id will be taken
as the key such as `gge7`. Schematic component JSON is a little bit complicated,
it has lots of other **JSON Keys**, such as `polyline`, `image`, `path` etc.

Note: please check the other shapes format via below JSON example

#### JSON example
check the complete JSON object via github gist [Schematic Json object]
(https://gist.github.com/dillonHe/fe0bb029c51603077ad9)
https://gist.github.com/dillonHe/fe0bb029c51603077ad9.js

---

##  EasyEDA PCB File Object
---

# EasyEDA PCB File Object

Note: PCB and Package use the same format.

*EasyEDA PCB File Object* is a JSON Object which allows you to hack your designs
via another language, such as Javascript, Python, PHP, C, C++. The interesting
thing is that your can control/modify your design in EasyEDA editor via
Javascript. So you can use codes to create your own outline.

### Rules

#### JSON Keys
 Every EasyEDA graph unit has an unique key, such as  "TRACK", "PAD", "VIA",
"TEXT", "DIMENSION", "FOOTPRINT", "ARC", "RECT", "CIRCLE", "HOLE", "COPPERAREA",
"SOLIDREGION", "DRCRULE", "FABRICATION"

#### itemOrder key
   Because of an object in EasyEDA is an unordered set of name/value pairs in
[JSON format](http://json.org/), but EasyEDA's graphs are ordered. We need an
array to store the order of these objects. Every package has an itemOrder key
and the whole JSON object has an itemOrder key.

### Example

#### File

![](../_images/easyeda-380_Format_filePCBJson.png)
Open [PCB Example ](https://easyeda.com/guest/EasyEDA_File_Format-D6igsqZXL)

#### TRACK
```
    "TRACK":{
        "gge6":{
            "gId":"gge6",
            "layerid":"1",
            "net":"S$7",
            "pointArr":[
                {
                    "x":357,
                    "y":171
                },
                {
~~~

~~~~
                    "x":456,
                    "y":171
                }
            ],
            "strokeWidth":1
        }
        ......
    },
```
All tracks will be stored to **TRACK** key, their id will be taken as the key
such as `gge6`.

#### SIGNALS

EasyEDA groups all of the objects with the same net name in one array.

#### FOOTPRINT

  All packages will be stored to **FOOTPRINT**, their id will be taken as the
key such as `gge7`. PCB package JSON is little bit complicated, it has lots of
other **JSON Keys**, such as `TRACK`, `ARC`, `RECT` etc.

Note: please check the other shapes format via below JSON example

#### JSON example

Check the complete JSON object via github gist [PCB Json object]
(https://gist.github.com/071d4680dcdbf6bf9dd6.git)
https://gist.github.com/dillonHe/071d4680dcdbf6bf9dd6.js




# EasyEDA Editor API

## EasyEDA API Plug

 Before reading this capture, please check [EasyEDA Document Format]
(https://docs.easyeda.com/en/DocumentFormat/0-EasyEDA-File-Format-
Index/index.html) first.

### Why Need API

After route the PCB, you found out that you need to enlarge all tracks size a
bit little, How?
After route the PCB, you found out that all Vias' hole size is too small, How to
fix this?
How to create a board outline using code?
EasyEDA API will let you control your designs in an easy way.

### How to use API

#### How to find the plug entrance

You can click ** > Advanced > Extensions Setting** on the top toolbar image as
below.

#### Extensions Setting
~~~~

~~~~
You can enable or disable the default extensions, after enable, please
**reload** the EasyEDA editor. We will give you a file about how to create an
extensions soon.
![picture 2](../_images/1-How-to-Use-API-20200804064501.png)



If you enable the **Extension Name** Extension, you will find a button on the
tool bar like bellow image:
![picture 3](../_images/1-How-to-Use-API-20200804064527.png)


You can check our **github** codes of this API via
[https://github.com/dillonHe/EasyEDA-Documents/tree/master/API/example/theme]
(https://github.com/dillonHe/EasyEDA-Documents/tree/master/API/example/theme),
check the **manifest.json** and **main.js** out, you will find out how to create
an extension.

**How to install an extension**

1. Click the Load Extension button
2. Click the select file button
3. Select **All** the files.
4. Type a name
5. Click the load button.
6. **Close** EasyEDA editor and open it again.
![](../_images/easyeda-174_API_Extensions_InstallExtension.png)

#### Scripts

If you just need some simple functions, you don't need to create an extension.
You just need to create a single Javascipt file and keep it in this list.

1. You can select the `Hello World`, then click the `Run` button, the response
as below image.
2. You can select some items, then try `Move Selected Objects`.
3. You can install your own scripts, then they will show on **User Scripts**.
![](../_images/easyeda-175_API_Scripts_RunHelloWorld.png)

#### Run Script code

In some case, you just need to run the function one time, such as create a user
define board outline in codes, changing the Track width, change the hole size
etc. You can use this way.
![](../_images/easyeda-176_API_ScriptSample.png)

**example 1 Art**
You can open an empty schematic and copy [this example javascript codes]
(https://raw.githubusercontent.com/dillonHe/EasyEDA-
Documents/master/API/example/schematicShapes.js) to the text box to run a test.
After clicking the `Run` button, you will see bellow art image.
![](../_images/easyeda-177_API_ScriptSample_Art.png)

```
testInsertShape();
function testInsertShape(){
    api('insertShape', [
~~~~

```
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: "M520 500 C480 460 550 430 480 410",
            strokeColor: "#000000",
            strokeStyle: 0,
            strokeWidth: "1"
        }
    ]);
    api('insertShape', [
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeLotusFlower(500,550,3,80,40),
            strokeColor: "#000000",
            strokeStyle: 0,
            strokeWidth: "1"
        },
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeLotusFlower(700,500,6,70,30),
            strokeColor: "#ff00ff",
            strokeStyle: 0,
            strokeWidth: "1"
        },
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeFlower2(660,670,4, 14,-Math.PI/4, 63.246,-0.32175,
84.85,Math.PI/4),
            strokeColor: "#cccc00",
            strokeStyle: 0,
            strokeWidth: "2"
        }
    ]);
}
/** Lotus shape path */
function shapeLotusFlower(cx,cy,n,r,r2){
    var pathD = '', angle, angle2, x, y, x2, y2;
    function p(x,y){
        return (x+cx)+' '+(y+cy);
    }
    for(var i=0; i<n; i++){
        angle = i * Math.PI / n;
        angle2 = (i / n + 0.5) * Math.PI;
        x = r * Math.cos(angle);
        y = r * Math.sin(angle);
        x2 = r2 * Math.cos(angle2);
        y2 = r2 * Math.sin(angle2);
        pathD += 'M'+p(x,y)
            +'C'+p(x2,y2)+' '+p(-x2,-y2)+' '+p(-x,-y)
            +'C'+p(x2,y2)+' '+p(-x2,-y2)+' '+p(x,y);
    }
    return pathD;
}
/** Petal shape path */
function shapeFlower2(cx,cy,n,r1,a1,r2,a2,r3,a3){
```

~~~~
    var pathD = '', angle, angle2, angle3, angle4, x, y, x2, y2;
    function p(r,thi){
        return (r * Math.cos(thi) + cx)+' '+(r * Math.sin(thi) + cy);
    }
    function polar(r,thi){
        return {r:r,thi:thi};
    }
    for(var i=0; i<n; i++){
        angle = i>0 ? angle4 : a1 + i * 2 * Math.PI / n;
        angle2 = a2 + i * 2 * Math.PI / n;
        angle3 = a3 + i * 2 * Math.PI / n;
        angle4 = a1 + (i+1) * 2 * Math.PI / n;
        pathD += (i>0 ? '' : 'M'+p(r1,angle))
            +'C'+p(r2,angle2)+' '+p(r3,angle3)+' '+p(r1,angle4);
    }
    return pathD;
}
```
**example 2 Change track width and via hole size**
You can open a **PCB** and copy [this example javascript codes]
(https://raw.githubusercontent.com/dillonHe/EasyEDA-
Documents/master/API/example/modifyTrackVia.js) to the text box to run a test.
After that, All tracks will be 10mil.

```

//you can get the EasyEDA json objects like
https://gist.github.com/071d4680dcdbf6bf9dd6.git
//try to pen a pcb, then run bellow codes.

var json = api('getSource', {type: "json"}),
    id;

for(id in json.TRACK){
    if(json.TRACK.hasOwnProperty(id)){
        json.TRACK[id].strokeWidth = api('edit.unitConvert',
{type:'mil2pixel',value:10}); // 10mil
    }
}

for(id in json.VIA){
    if(json.VIA.hasOwnProperty(id)){
        json.VIA[id].holeR = api('edit.unitConvert',
{type:'mil2pixel',value:10}); // 10mil
    }
}
api('applySource', {source: json, createNew: true});
```


---



•##  EasyEDA Coordinate System & Unit
---



### EasyEDA Coordinate System
~~~~

~~~
EasyEDA's editor is based [SVG]
(http://en.wikipedia.org/wiki/Scalable_Vector_Graphics), SVG viewport,
(Coordinates increase **left-to-right** and **top-to-bottom**, the same as
EasyEDA ). But SVG's origin is fixed at the left top corn, and EasyEDA's origin
can be modified at the any place.
![](../_images/easyeda-178_API_Coordinate.png)

Be careful this, they are different from **Cartesian coordinate system**

### Unit

There are two kinds of unit in our editor, SVG Canvas unit and real world
EasyEDA unit. SVG Canvas unit is **Pixel**. The real world EasyEDA unit in
schematic is also **Pixel**, but in PCB, there are **mm**, **mil** and **inch**.
We use bellow map to convert Canvas to real world.
- 1 pixel = 10 mil
- 1 pixel = 0.254mm
- 1 pixel = 0.01inch

There are API for these convert.

-   mm2pixel: convert 10mm to pixel

```
        var result = api('unitConvert', {type:'mm2pixel',value:10});
```

-   mil2pixel: convert 10mil to pixel

```
        var result = api('unitConvert', {type:'mil2pixel',value:10});
```

There are other convert method, such as `inch2pixel`, `pixel2mm`, `pixel2mil`
and `pixel2inch`.

  All EasyEDA's value is based pixel, if you can keep in mind that 1 pixel equal
10mil or 0.254 mm, you don't need to use any convert function.
For example, if you want to change a Track to 20mil, so you just need to use 2.



---



##  EasyEDA API List
---



# API List

### Get EasyEDA Source

1. get EasyEDA JSON objects, type is `json`, you can check  [PCB Json object]
(https://docs.easyeda.com/en/DocumentFormat/5-EasyEDA-PCB-File-
Object/index.html) out to know more.

        var result = api('getSource', {type:'json'});
~~~

~~~
2. get [EasyEDA compress string](https://docs.easyeda.com/en/DocumentFormat/3-
EasyEDA-PCB-File-Format/index.html), EasyEDA save this string to our database,
it is a bit little hard to read and understand, but it is small in size. EasyEDA
save this string to our database.

        var result = api('getSource', {type:'compress'});

3. Get SVG string

        var result = api('getSource', {type:'svg'});

Check the [Get EasyEDA source example codes]
(https://raw.githubusercontent.com/dillonHe/EasyEDA-
Documents/master/API/example/modifyTrackVia.js).

### Apply Source

 After you can use your codes to hack EasyEDA's source, then you need to apply
the source to EasyEDA's editor. You can

1. Apply as compress string

//will open a new editor and convert compressStr to EasyEDA file.
```
        api('applySource', {source:'compressStr', createNew: true});
```
2. Apply as Json object.

//will modify the active file and convert json object to EasyEDA file.
```
        api('applySource', {source: json, createNew: !true});
```

Check the [Apply Source example codes]
(https://raw.githubusercontent.com/dillonHe/EasyEDA-
Documents/master/API/example/modifyTrackVia.js).

### Get Shape

 If you want to get an EasyEDA json object by **id**, you can try to use bellow
code.

          var obj = api('getShape', {id:'gge13'})

### Delete Shapes

  Removing shapes by follow code

### Update Shape

 If you want to modify an EasyEDA object, you can use this API.

Change the net to GND and the shape to ELLIPSE:
```
     api('updateShape', {
        "shapeType": "PAD",
        "jsonCache": {
~~~

~~~
        "gId": "gge5",
        "net": "GND"
        "shape": "ELLIPSE"
        });
```

 shapeType and gId are must provided.
1. Schematic
`shapeType`, `schlib`, `rect`, `polyline`, `polygon`, `wire`, `bus`, `image`,
`circle`, `ellipse`, `line`, `path`, `arc`, `annotation`, `junction`,
`netlabel`, `busentry`, `arrowhead`, `noconnectflag`, `pin`, `netflag`
2. PCB
`shapeType`, `FOOTPRINT`, `TRACK`, `COPPERAREA`, `SOLIDREGION`, `RECT`,
`CIRCLE`, `TEXT`, `ARC`, `DIMENSION`, `PAD`, `VIA`, `HOLE`


### Create Shape

 If you want to create EasyEDA shape by codes, you can try. We will provide more
information about this API soon, now we just provide examples. You will find out
how to do.

with shortUrl:
@example
```
api('createShape', {shapeType:'schlib', shortUrl:'nxlVIGgQO', from:'system',
title:'556_DIL14', x:400, y:300});
api('createShape', {shapeType:'FOOTPRINT', shortUrl:'RrkewO6Oi', from:'system',
title:'ARDUINO_PRO_MINI', x:400, y:300});
```
with jsonCache object:

@example
```
api('createShape', {
  "shapeType": "PAD",
  "jsonCache": {
    "gId": "gge5",
    "layerid": "11",
    "shape": "ELLIPSE",
    "x": 382,
    "y": 208,
    "net": "",
    "width": 6,
    "height": 6,
    "number": "1",
    "holeR": 1.8,
    "pointArr": [],
    "rotation": "0"
  }
});
```
@example
```
api('createShape', {
    "shapeType": "polygon",
    "stroke": "#000000",
    "stroke-width": "1",
~~~

~~~
    "stroke-style": "dashed",
    "fill": "none",
    "points": [
        {"x": 390, "y": 580},
        {"x": 450, "y": 450},
        {"x": 520, "y": 580},
        {"x": 610, "y": 490}
    ]
});
```

@example
```
api('createShape', {
    "shapeType": "arrowhead",
    "x": 300,
    "y": 300,
    "color": "#339933",
    "size": "3",
    "rotation": 0
});
```

@example
```
var ts = ["no_connect_flag", "arrowhead", "busentry", "netLabel_GNd",
"netLabel_GnD", "netLabel_gnD", "netLabel_Bar", "netLabel_VEE", "netLabel_-5V",
"netLabel_+5V", "netLabel_VCC", "netLabel_volProbe", "netLabel_netPort",
"netLabel_text", "pin", "annotation"];
for(var i=0;i<ts.length;i++){
    api('createShape', {
        "shapeType": ts[i],
        "x": 300 + i%5*50,
        "y": 300 + (i/5|0)*50
    });
}
```
with cached or pre-defined libs:
@example
```
api('createShape', {"shapeType": "pcblib", from:'GeneralPackages',
title:'C0402', x:400, y:300});
@example
api('createShape', {"shapeType": "schlib", from:'EasyEDALibs', title:'HDR2X2',
x:400, y:300});
```

@example 4
```
api('createShape', {
    "shapeType": "schlib",
    "gId": "gge6",
    "head": {},
    "itemOrder": [],
    "annotation": {
        "gge8": api('createShape', 'annotation', {}),
        "gge9": api('createShape', 'annotation', {})
    },
~~~

~~~
    "pin": {
        "gge11": api('createShape', 'pin', {}),
        "gge14": api('createShape', 'pin', {})
    },
    "polyline": {
        "gge10": api('createShape', 'polyline', {}),
        "gge12": api('createShape', 'polyline', {})
    }
});
```


@example 5
```
api('createShape', {
    "shapeType": "schlib",
    "gId": "gge6",
    "head": {},
    "children": [
        api('createShape', 'polyline', {}),
        api('createShape', 'polyline', {}),
        api('createShape', 'pin', {}),
        api('createShape', 'pin', {}),
        api('createShape', 'annotation', {}),
        api('createShape', 'annotation', {})
    ]
});
```
@example 6
```
api('createShape', {
    "shapeType": "schlib",
    "gId": "gge6",
    "head": {},
    "children": api('createShape', [
        ['polyline', {}],
        ['polyline', {}],
        ['pin', {}],
        ['pin', {}],
        ['annotation', {}],
        ['annotation', {}]
    ])
});
```
### UI

   If you want to create an extension, not just a run one time script, maybe
need toolbar button. You can check the [example]
(https://github.com/dillonHe/EasyEDA-Documents/tree/master/API/example/theme)
before you read.

#### Create Toolbar Button



@example create a button:
```
     api('createToolbarButton', {
      icon:'extensions/theme/icon.svg',
      title:'Theme Colors...',
~~~

~~~
      fordoctype:'sch,schlib',
      cmd:"extension-theme-setting"
     });
```

@example toolbar with menu:
```
api('createToolbarButton', {
 icon:'extensions/theme/icon.svg',
 title:'Theme Colors...',
 fordoctype:'sch,schlib',
 "menu" : [
     {"text":"White on Black", "cmd":"extension-theme-WhiteOnBlack"},
     {"text":"Black on White", "cmd":"extension-theme-BlackOnWhite"},
     {"text":"Custom Colors...", "cmd":"extension-theme-setting"}
 ]
});
```

#### Create Extension Menu


@example
```
api('createExtensionMenu', [
 {
     "text":"Theme Colors...",
     "fordoctype": "sch,schlib",
     "cmd": "extension-theme-white"
 }
]);
```

### Create Dialog
  check the [example](https://github.com/dillonHe/EasyEDA-
Documents/tree/master/API/example/theme)

### Command List

#### Clone

// clone gge2 gge3 and return their new ids.
```
      var newIds = api('clone', {ids:["gge2","gge3"]})
```

#### Delete
```
      api('delete', {ids:["gge2","gge3"]});
```
#### Rotate

// rotate ids to 90 degree
```
    api('rotate', {ids:["gge2","gge3"],degree:90});
```

#### Rotate Left
~~~

~~~
//anticlockwise
```

    api('rotate_left', {ids:["gge2","gge3"]});
```


#### Rotate Right


//clockwise
```

    api('rotate_right', {ids:["gge2","gge3"]});
```


#### Fliph
```

    api('fliph', {ids:["gge2","gge3"]});
```


#### Flipv
```

    api('flipv', {ids:["gge2","gge3"]});
```


#### Align Left


    api('align_left', {ids:["gge2","gge3"]});


#### Align Right


    api('align_right', {ids:["gge2","gge3"]});


#### Align Top


    api('align_top', {ids:["gge2","gge3"]});


#### Align Bottom


    api('align_bottom', {ids:["gge2","gge3"]});


### Selection


 Change or get selection states of EasyEDA objects in editor.


#### Select


// gge2 and gge3 will be marked as selected.
```

      api('select', {ids:["gge2","gge3"]});
```
#### Select None


//no objects will be selected.
```

      api('selectNone');
```
#### Get Selected Ids


     var ids = api('getSelectedIds');
~~~

~~~~
### Move

You can use [Update Shape](#UpdateShape) to change the shapes position, but the
Move method is better in this case.

#### Move Objects

Move shapes in relative coordinates, like move the shapes in arrow keys.

//Move gge2 and gge3 from left to right in 20pixel or 200mil step
//from top to bottom in 20pixel or 200mil step.
```
    api('moveObjs', {objs:[{gId:"gge2"},{gId:"gge3"}], addX: 20, addY: 20});
```

//Move gge2 and gge3 from right to left in 20pixel or 200mil step
```
    api('moveObjs', {objs:["gge2","gge3"], addX:-20});
```

//Move selected objects from left to right in 20pixel or 200mil step
```
    api('moveObjs', {addX:20});
```

#### Move Objects To

How to move a `VIA` or `junction` to position `{x:'10mil', y:'10mil'}` ?, Move
shapes to absolute coordinates.

//Move gge2 and gge3 to Canvas postion 20,20, the real coordinates are dedpend
the origin.
```
     api('moveObjsTo', {objs:[{gId:"gge2"},{gId:"gge3"}], x:20, y:20});
```

//move gge2 and gge3 to 10mm, 10mm coordinates
```
     api('moveObjsTo', {objs:["gge2","gge3"], x: api('coordConvert',
{type:'real2canvas',x: '10mm'}), y: api('coordConvert', {type:'real2canvas',y:
'10mm'})});
```

//Move selected objects to Canvas postion 20,20, the real coordinates are
dedpend the origin.
```
     api('moveObjsTo', {x:20, y:20});
```

It is very easy to understand to move a PAD, VIA, Junction to absolution
coordinates. But what are the effects of moving TRACK, FOOTPRINT, netlabel to
some where. Just try to play the codes, you will find out the regular pattern.

### SetOriginXY

EasyEDA's canvas origin is 0,0, you can't change it. But the real coordinates
can be mapped to any where.
~~~~

~~~
//set the real origin point to canvas x = 400, y = 300. X,Y is pixel all the
time.
```
    var result = api('setOriginXY', {x:400,y:300});
```

### Coordinate Convert

You can use mm or mil or inch as units, but when you apply the Parameters to SVG
graph, you must use coordinate convert.

//convert the canvas x 400 to real postion, the value is depent your units and
origin point.
```
    var result = api('coordConvert', {type:'canvas2real',x:400})
```

//the default units is your canvas units, but you can add a units like 300mm.
//if your PCB's units is mil, then you will get the canvas coordinate
400mil,300mm.
```
     var result = api('coordConvert', {type:'real2canvas',x:400,y:'300mm'});
```

  If you set the origin to **0,0**. It is very easy to map the coordinate in
your mind, you don't need to use API to convert. the canvas coordinate
**100,100** equal the real coordinate **1000mil, 1000mil** or **1inch, 1inch**
or **393.7mm, 393.7mm**

### Value Convert

 How to set the pad's hole size to 20mm? How to set the Track width to 20mil?

//the default units is your canvas units, but you can add a units like mm, mil,
inch, even pixel.
```
     var result = api('valConvert', {type:'real2canvas',val:400});
     result = api('valConvert', {type:'real2canvas',val:'400mm'})
```

//convert the 400 pixel to real value, the value is depent your units , if the
unit is mil, the result should be 4000
```
result = api('valConvert', {type:'canvas2real',val:400})
```

  If you can keep in mind 1pixel in canvas equal 10mil, so you don't need this
API, you can do it in raw way. For example,
  If you want to update the track size to 20mil, you can do.
```
     api('updateShape', {
        "shapeType": "TRACK",
        "jsonCache": {
            "gId": "gge5",
            "strokeWidth": 2
        }
        });
~~~

~~~~
```
  Or
```
     api('updateShape', {
        "shapeType": "TRACK",
        "jsonCache": {
            "gId": "gge5",
            "strokeWidth":  api('valConvert', {type:'real2canvas',val:'20mil'})
        }
     });
```

### Get SVG Arc Path

 SVG [Arc path Parameter](http://www.w3.org/TR/SVG11/paths.html#PathElement) is
very complex,  We provide a API to convert human read ARC parameter to SVG path.
```
     var result = api('getSvgArcPathByCRA', {cx:0, cy:0, rx:90, ry:90,
startAngle:0.1, endAngle:0.7, sweepFlag:1});
```

result should be `M89.55037487502231 8.985007498214534A90 90 0 0 1
68.83579685560396 57.97959185139219`



---



•##  EasyEDA API Examples
---



# Examples

Check [Github example](https://github.com/dillonHe/EasyEDA-
Documents/tree/master/API/example)

Enjoy it, if you have any questions, do let us know.

modifyTrackVia.js:

```


//you can get the EasyEDA json objects like
https://gist.github.com/071d4680dcdbf6bf9dd6.git
//try to pen a pcb, then run bellow codes.

var json = api('getSource', {type: "json"}),
    id;

for(id in json.TRACK){
    if(json.TRACK.hasOwnProperty(id)){
        json.TRACK[id].strokeWidth = api('edit.unitConvert',
{type:'mil2pixel',value:10}); // 10mil
    }
}

for(id in json.VIA){
~~~~

~~~
    if(json.VIA.hasOwnProperty(id)){
        json.VIA[id].holeR = api('edit.unitConvert',
{type:'mil2pixel',value:10}); // 10mil
    }
}
api('applySource', {source: json, createNew: true});

```


schematicShapes.js:
```
testInsertShape();

function testInsertShape(){
    api('insertShape', [
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: "M520 500 C480 460 550 430 480 410",
            strokeColor: "#000000",
            strokeStyle: 0,
            strokeWidth: "1"
        }
    ]);
    api('insertShape', [
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeLotusFlower(500,550,3,80,40),
            strokeColor: "#000000",
            strokeStyle: 0,
            strokeWidth: "1"
        },
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeLotusFlower(700,500,6,70,30),
            strokeColor: "#ff00ff",
            strokeStyle: 0,
            strokeWidth: "1"
        },
        {
            shapeTypeName: "path",
            fillColor: "none",
            pathString: shapeFlower2(660,670,4, 14,-Math.PI/4, 63.246,-0.32175,
84.85,Math.PI/4),
            strokeColor: "#cccc00",
            strokeStyle: 0,
            strokeWidth: "2"
        }
    ]);
}

/** Lotus shape path */
function shapeLotusFlower(cx,cy,n,r,r2){
    var pathD = '', angle, angle2, x, y, x2, y2;
    function p(x,y){
        return (x+cx)+' '+(y+cy);
~~~

~~~
    }
    for(var i=0; i<n; i++){
        angle = i * Math.PI / n;
        angle2 = (i / n + 0.5) * Math.PI;
        x = r * Math.cos(angle);
        y = r * Math.sin(angle);
        x2 = r2 * Math.cos(angle2);
        y2 = r2 * Math.sin(angle2);
        pathD += 'M'+p(x,y)
            +'C'+p(x2,y2)+' '+p(-x2,-y2)+' '+p(-x,-y)
            +'C'+p(x2,y2)+' '+p(-x2,-y2)+' '+p(x,y);
    }
    return pathD;
}
/** Petal shape path */
function shapeFlower2(cx,cy,n,r1,a1,r2,a2,r3,a3){
    var pathD = '', angle, angle2, angle3, angle4, x, y, x2, y2;
    function p(r,thi){
        return (r * Math.cos(thi) + cx)+' '+(r * Math.sin(thi) + cy);
    }
    function polar(r,thi){
        return {r:r,thi:thi};
    }
    for(var i=0; i<n; i++){
        angle = i>0 ? angle4 : a1 + i * 2 * Math.PI / n;
        angle2 = a2 + i * 2 * Math.PI / n;
        angle3 = a3 + i * 2 * Math.PI / n;
        angle4 = a1 + (i+1) * 2 * Math.PI / n;
        pathD += (i>0 ? '' : 'M'+p(r1,angle))
            +'C'+p(r2,angle2)+' '+p(r3,angle3)+' '+p(r1,angle4);
    }
    return pathD;
}
```



---
~~~