一、绘制原理图

1.1 新建工程



图1.1 新建项目

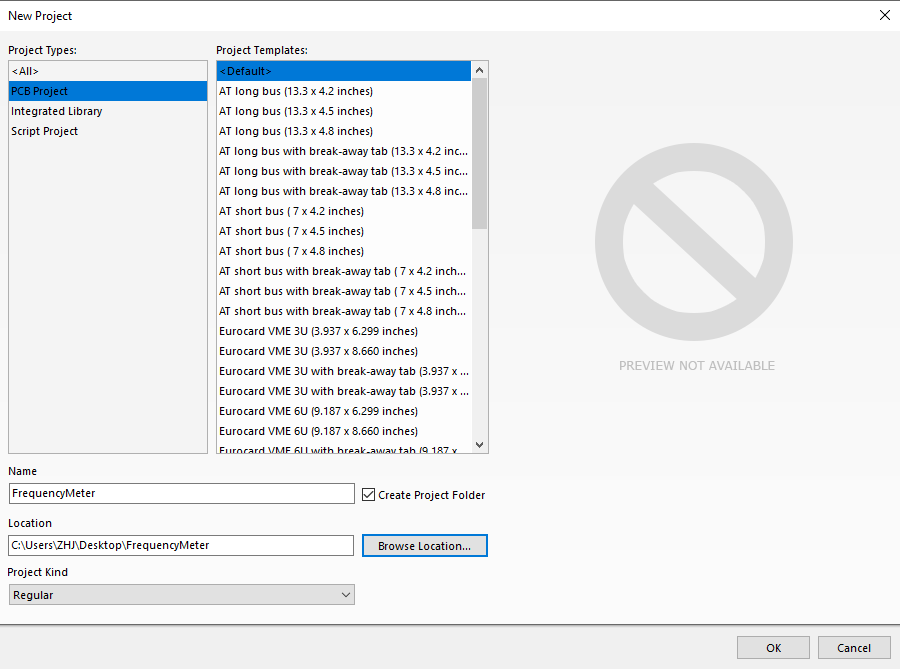


图1.2 新建工程文件

新建原理图文件

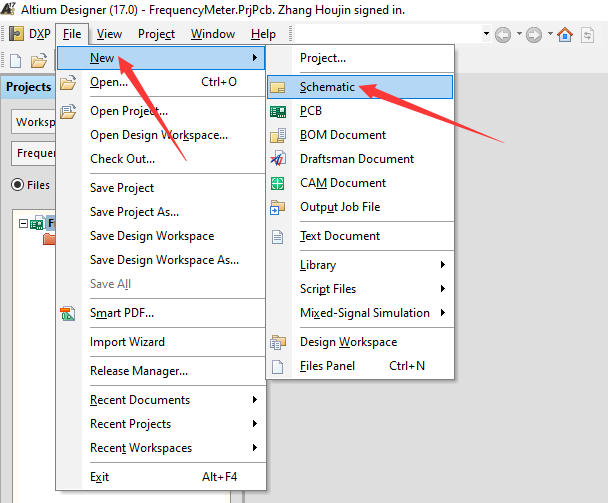


图1.3 创建原理图文件

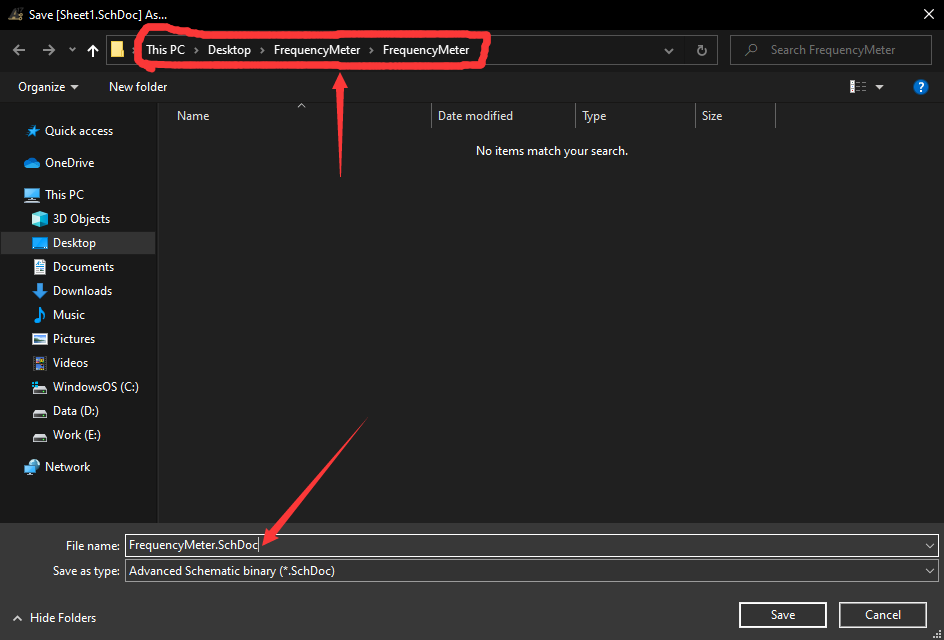


图1.4原理图路径

1.2 绘制原理图

1.2.1 主控芯片

本次项目我准备全部使用网上下载的原理图库和PCB库，不再自己设计。因此首先去网上查找相关的库文件。

主控芯片使用AT89C2051芯片，20引脚，我在网上找了一些，但没有找到非常合适的，只找到了相似的[.epw格式的元件库](https://componentsearchengine.com/AT89C2051-24PU/Microchip)。Ewp格式原件经过[格式转换](https://www.samacsys.com/altium-designer-library-instructions/)即可得到AD软件可以使用的.SchLib原理图库和.PcbLib原件库。

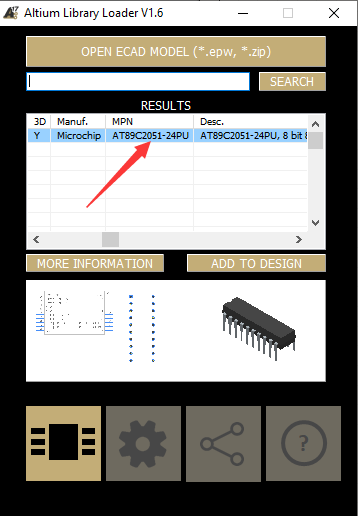


图1.5 文件格式转换

文件转换完成后，会在本地生成相应的库文件，里面包含此原件的原理图和封装。

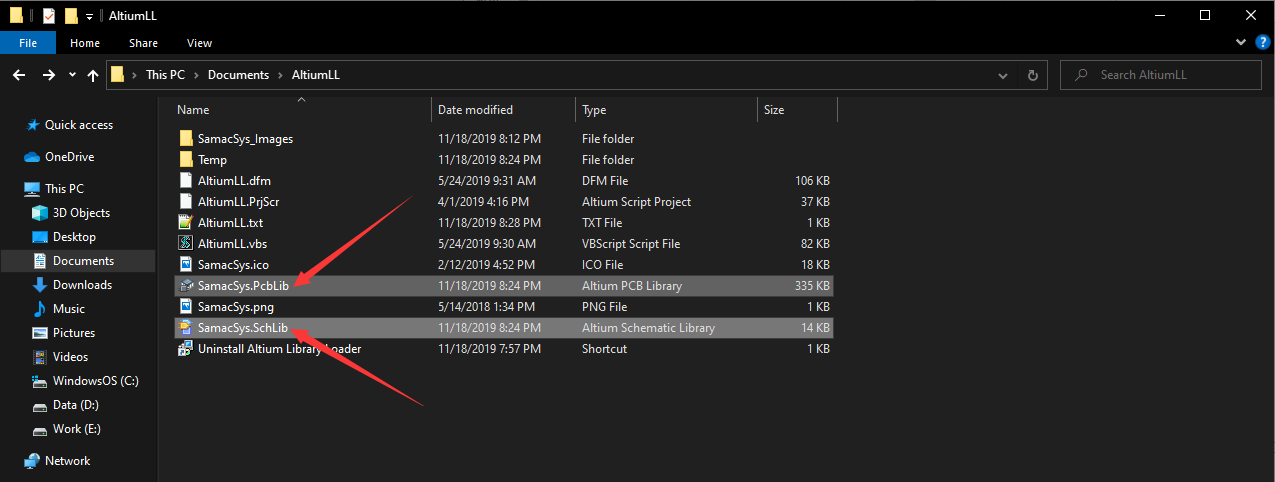
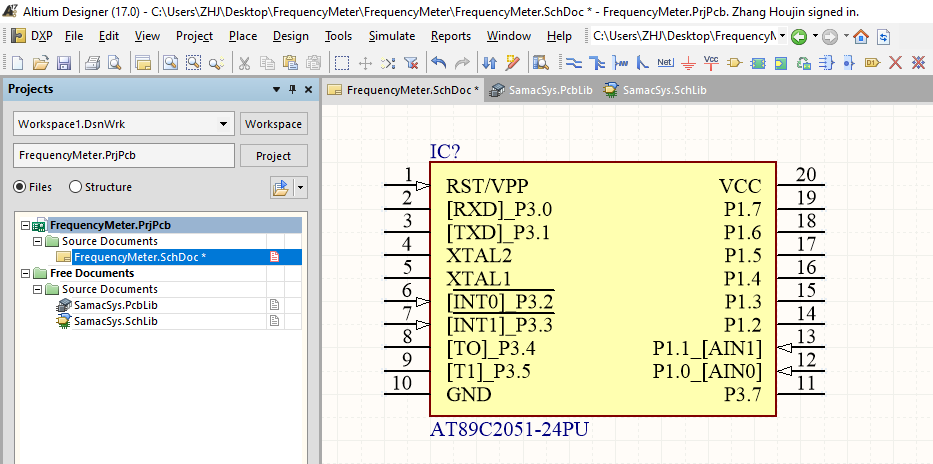


图1.6 原件封装库

现在把AT89C2051的原理图文件拖到原理图中就可以了。

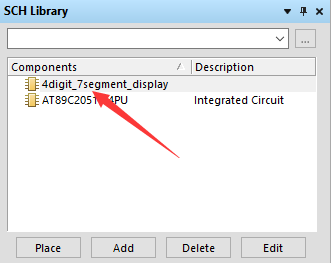


1.2.2 数码管芯片

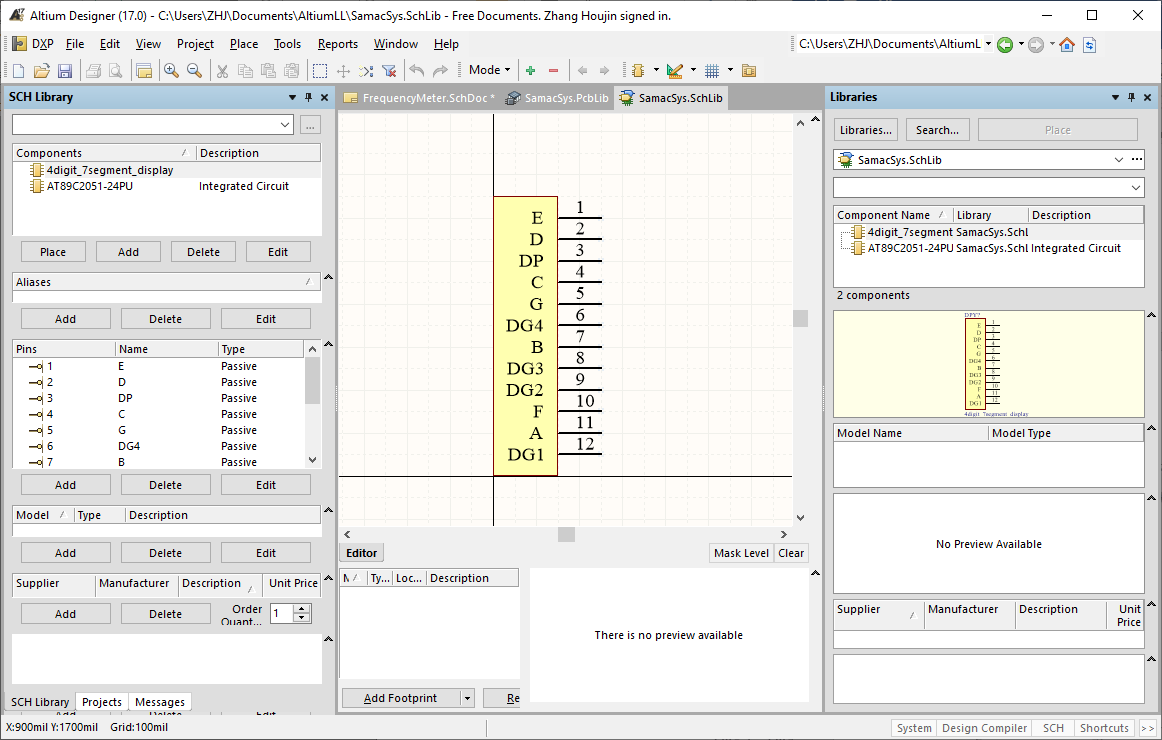
本次PCB使用到了四位一体数码管，我在网上找了很久都没找到合适的，于是决定自己画一个。先画原理图。

（1）原理图绘制

首先在原理图库中创建一个组件，命名为“4digit\_7segment\_display”。

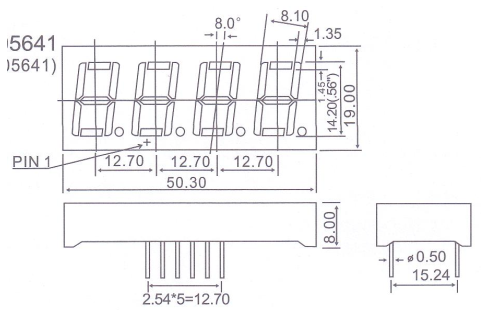


接着就是在编辑面板中画图了，画完之后长这样：

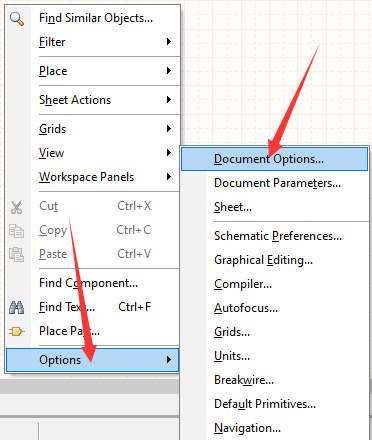


（2）封装库绘制

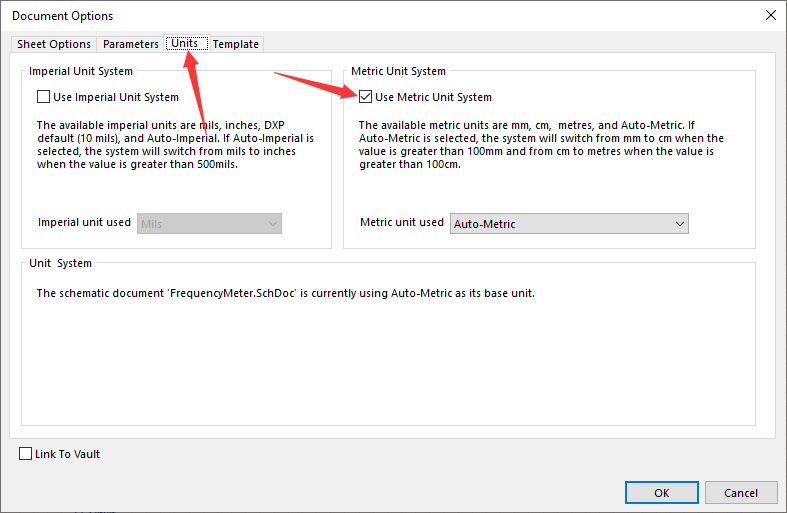
封装库的绘制参考课本9.4.2章节。首先收集相关的原件手册，在其中找的对应的封装标注。



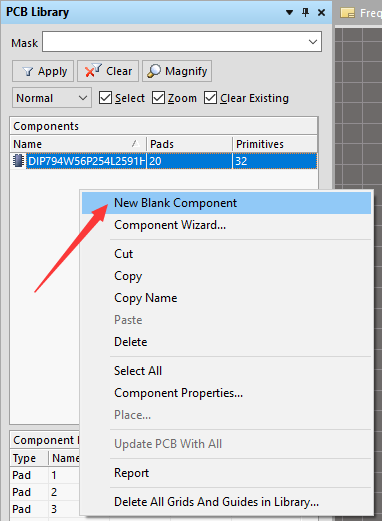
按照此封装的标注，进行原件封装的绘制。首先设置单位为公制单位。



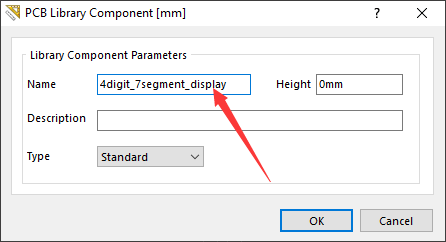
打开原理图设置界面，在units页面选择使用公制单位系统。



接下来绘制原件的外形轮廓，根据之前的原件封装，可以计算出该原件的矩形轮廓的范围。如果以该原件的中心点为坐标原点，则可以计算出矩形轮廓的四个顶点坐标：(-25.1,9.5)、(25.1,9.5)、(-25.1,-9.5)、(-25.1,-9.5)。实际绘图中，要比这些点的数值稍大一些。同样在原理图库中添加一个新的原件，命名为“4digit\_7segment\_display”。



接着重命名数码管封装

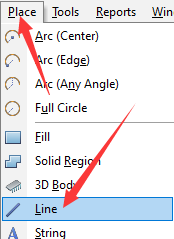


下面开始绘制数码管封装。

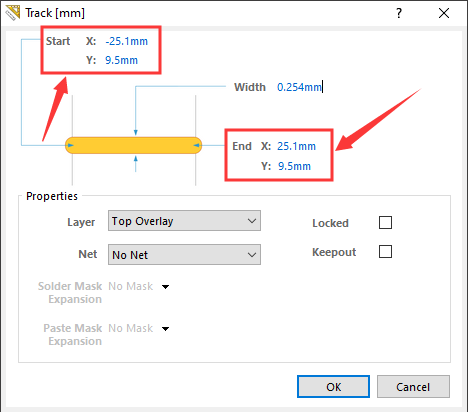
1. 切换当前层为丝印层

2. 绘制矩形边框

首先点击放置，选择直线。



之后绘制矩形上边直线，通过设置线的起始和终止位置实现该线的绘制。



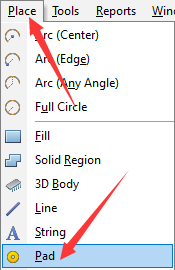
其余三根线使用同样的方法绘制，绘制完成后如下图所示。



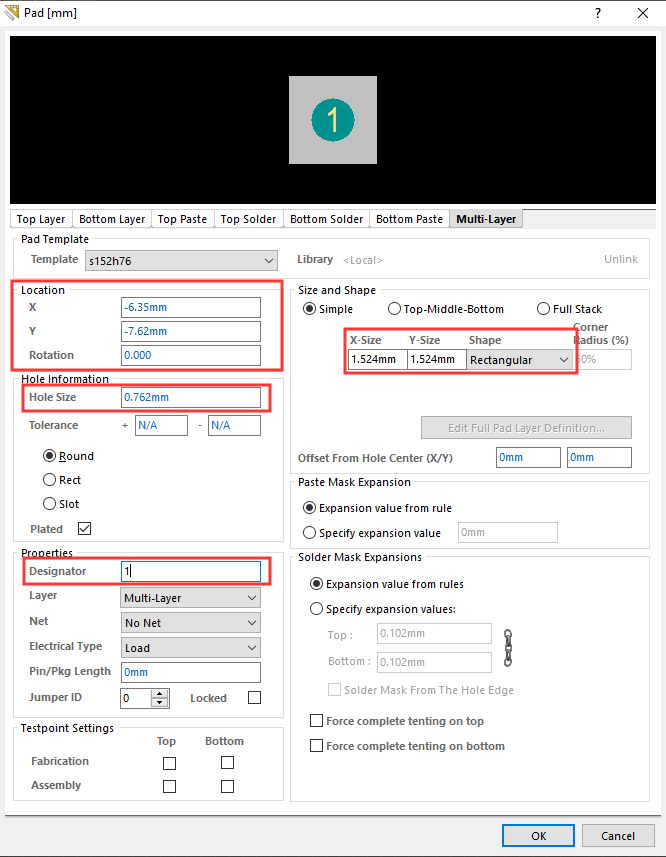
为了美观，接下来放置一个字符串表示四位一体数码管。



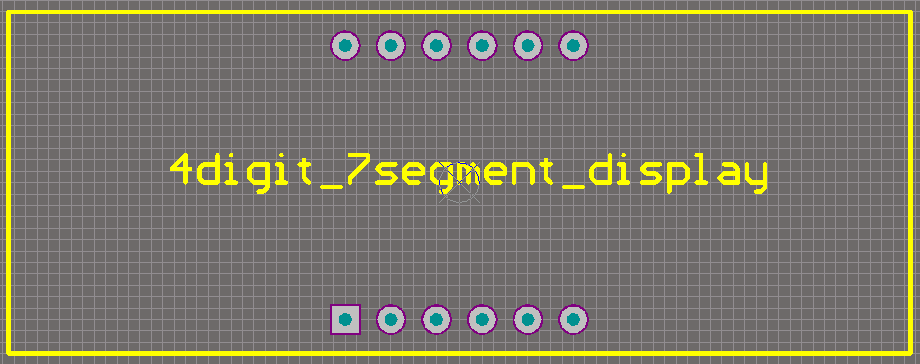
接下来放置焊盘。首先要计算焊盘的绝对位置，下面我根据课本的数据进行绘制。



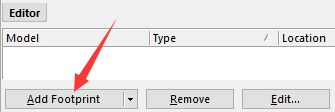
按下Tab键设置焊盘的属性，首先确定一号焊盘的属性。



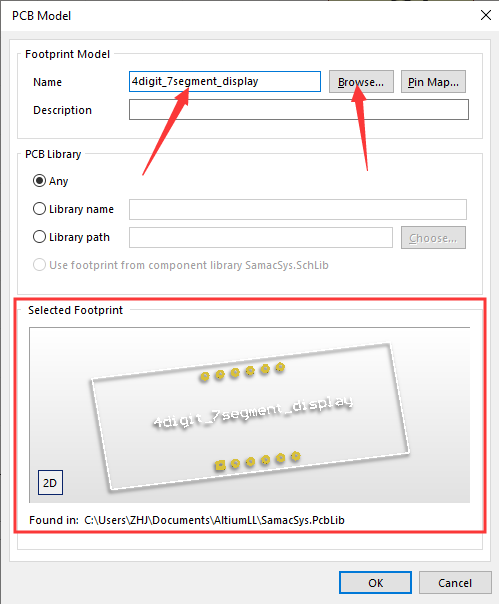
之后，按照一号焊盘的规则，放置剩余的十一个焊盘。



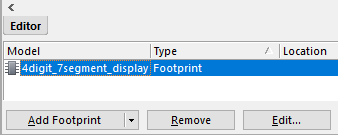
接下来保存该文件即可。下面需要将该原件的原理图与封装关联起来。先找到它的原理图，点击下方“Add Footprint”按钮添加封装文件。



找到刚才画完的封装文件，添加关联即可。



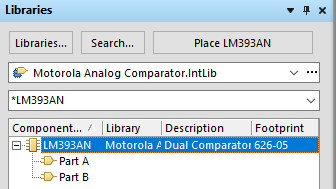
关联成功后可以看到原理图所关联的封装文件。



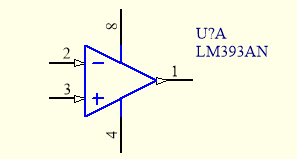
现在四位一体数码管的封装库就做好了。

1.2.3 运算放大器

本次工程中使用到的运放型号为LM393AN，在“Motorola Analog Comparator.IntLib”有该原件的原理图。下载该元件库，直接导入该原件即可。

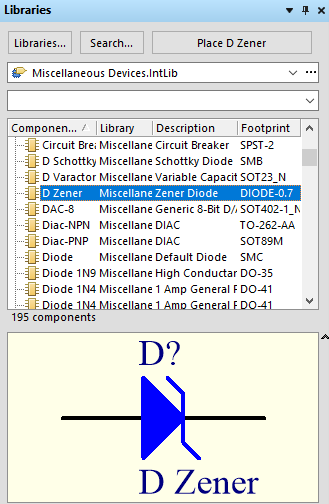


运放被分为两个部分，本次将会使用partA部分。



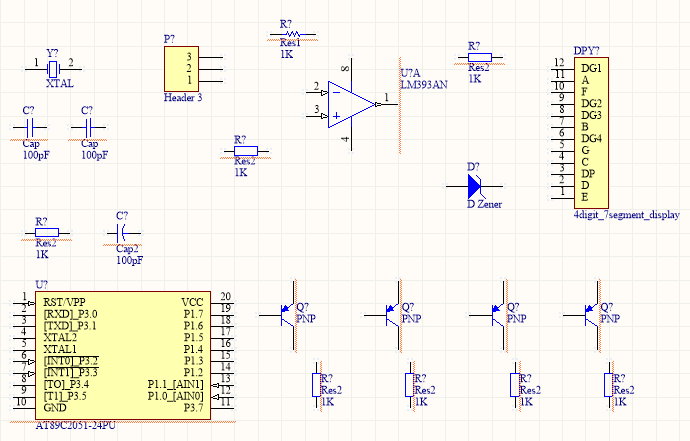
1.2.4 稳压二极管

课本原理图中的稳压二极管使用的是1N4730，这里我使用了“Miscellaneous Devices.IntLib”元件库中的齐纳二极管代替。



1.2.5 其他原件

其他原件算是比较常规的原件了，所以不再详细说明。按照原理图找到相应的原件并放置到面板上即可。

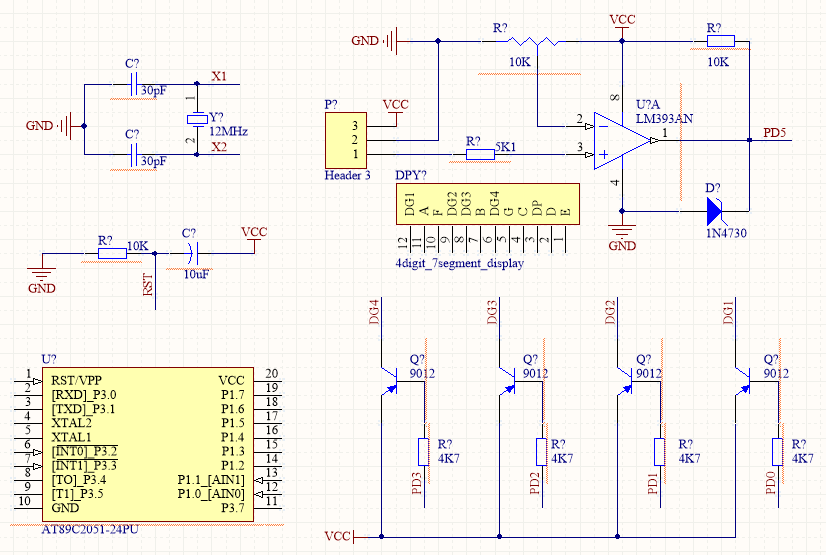


1.2.6 原理图连线

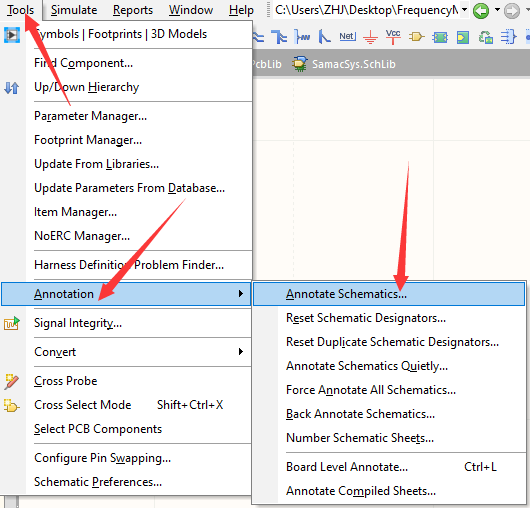
按照课本的原理图示例，连接各个原件。



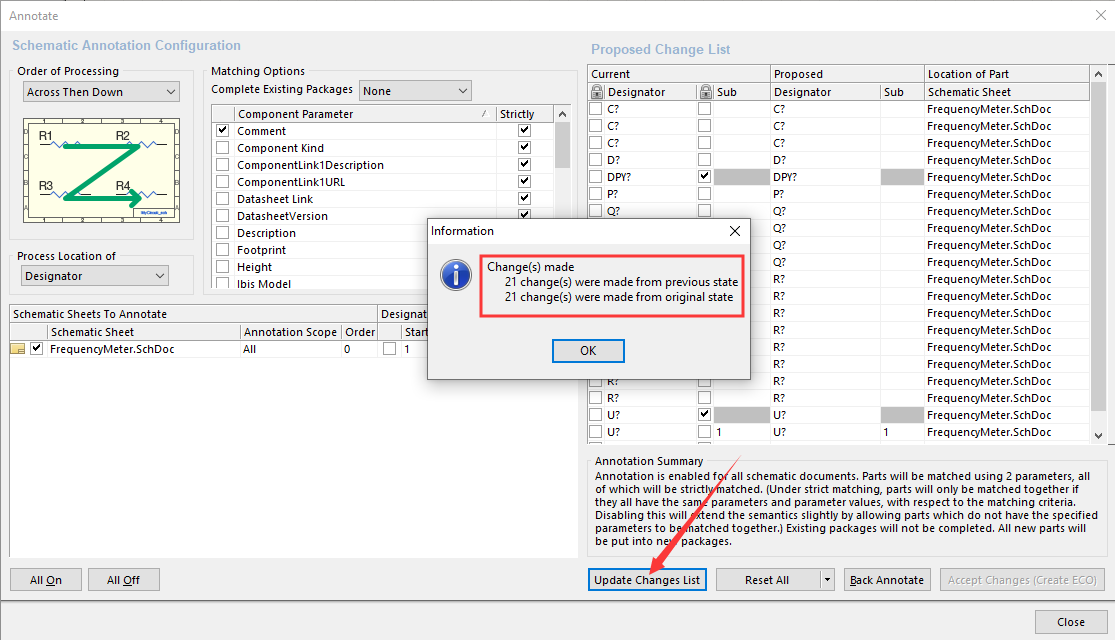
接下来进行原件描述的改写。按照课本的示例，改写各元件的描述。



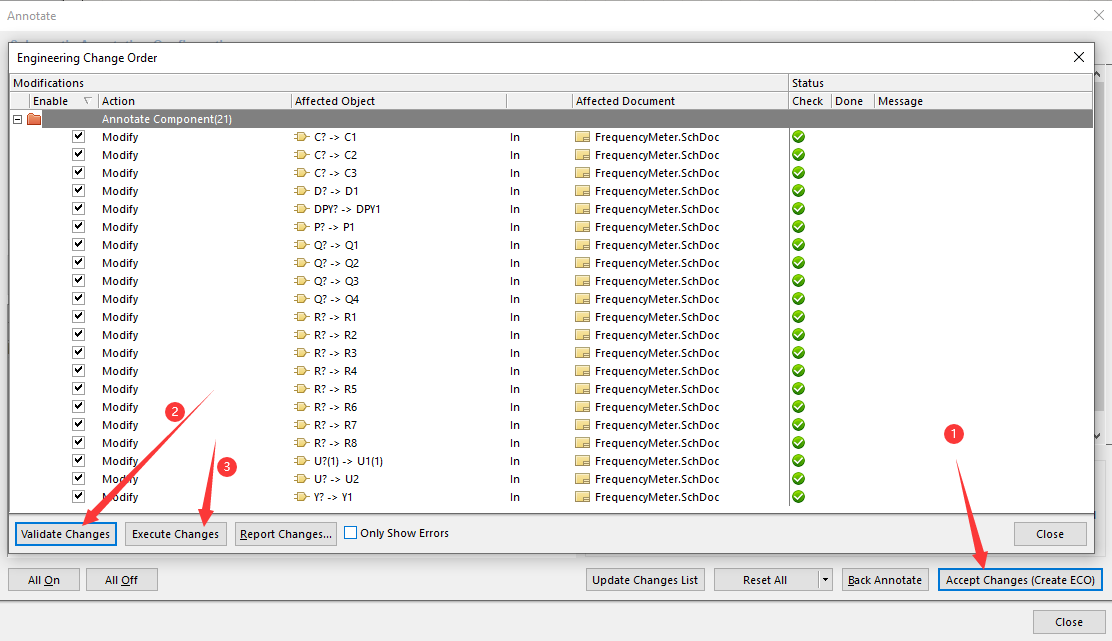
接下来需要生成每个元件的网络标号，使用AD的工具自动生成即可。



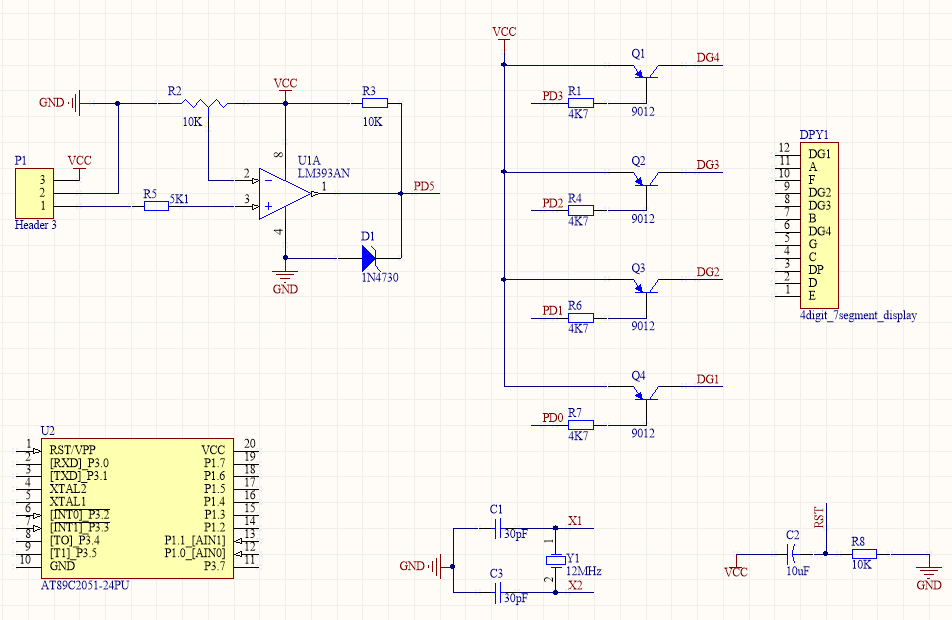
打开网络标号生成界面，设置好标号排布方式，点击更新列表。



之后点击接受这些改变，会弹出预改变原件的对话框，点击验证更新以检查元件封装是否正确。如果没有问题，点击执行更改按钮即可完成更新。

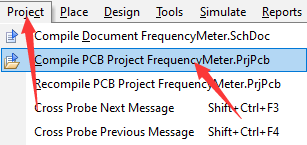


现在，原理图已经完成标号了。



1.2.7 编译检查

原理图绘制完成后，需要首先进行编译检查。



编译完成后，软件会在message界面给出很多错误和警告信息（如果有问题的话）。下面就需要针对这些错误信息对原理图进行修改。

我这里一共有47处错误信息，首先看第一项：

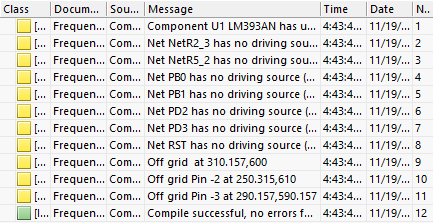


这是那个运算放大器，他警告说组件B没有使用。因为这个运放是LM393AN原件的一部分，我们只用了这一部分的原件，所以该警告可以忽略。

第二项报错：

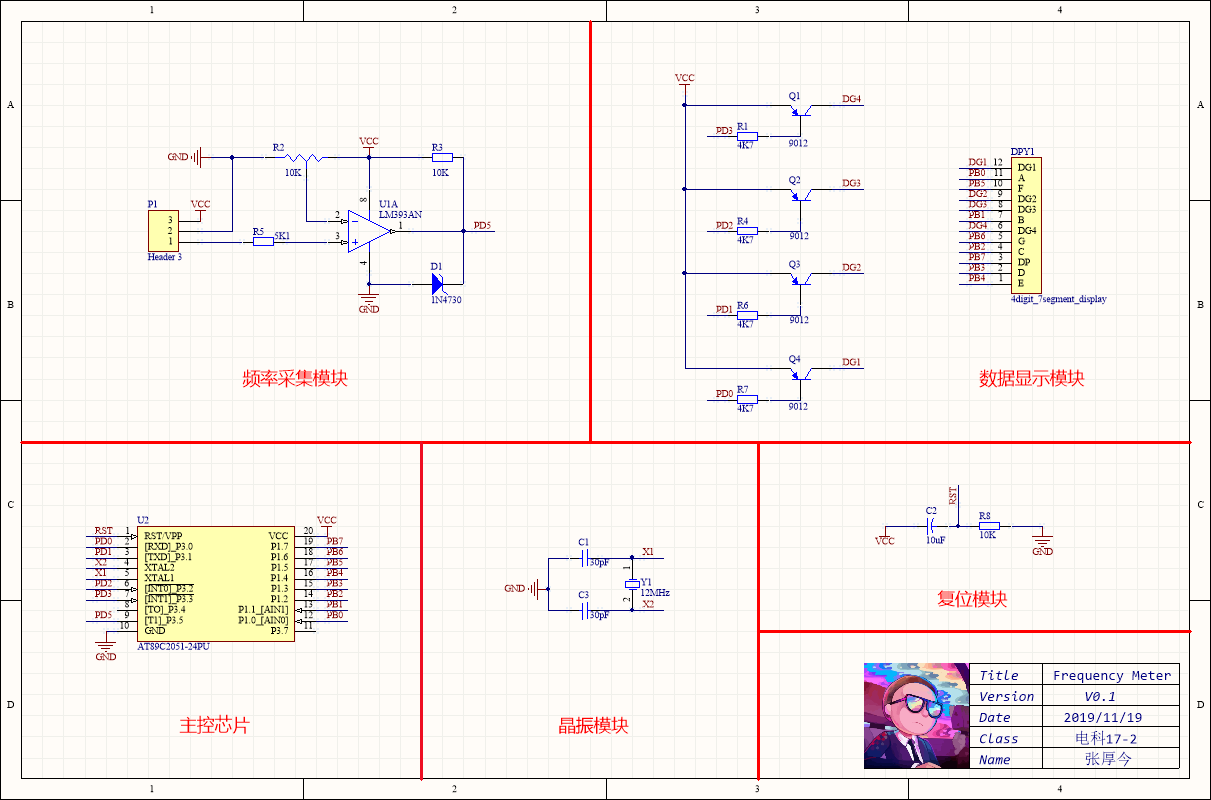


错误二提示，网络标号为DG1的网络，只有一个引脚。去原理图中看一下，发现果然有问题：我忘画主控芯片和数码管的网络标号了。马上修改原理图，重新编译后，错误信息已经没有了，剩下了29个警告信息。再经过我的修改，现在剩下了11个警告信息。大部分警告信息是因为没有驱动源或者原件没有对齐网格导致的，对原理图功能没有影响，所以这些警告可以忽略。



1.2.7 原理图修饰

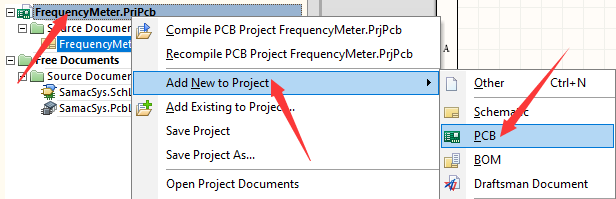
原理图绘制完成后，需要对其进行相关问题说明和修饰。首先创建图纸模板，将必要的信息填写到模板中。之后将原理图各部分模块进行区分，标注并说明各模块内容。



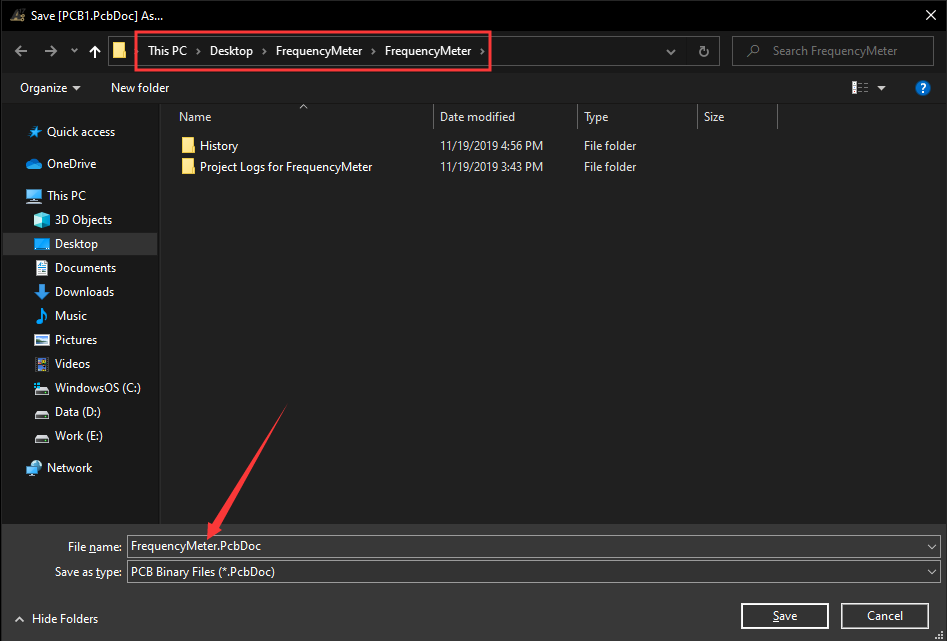
二、绘制PCB

2.1 新建PCB文件

首先在本工程目录下创建新的PCB文件。

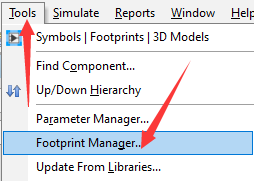


另存PCB文件，同时重命名为FrequencyMeter.PcbDoc

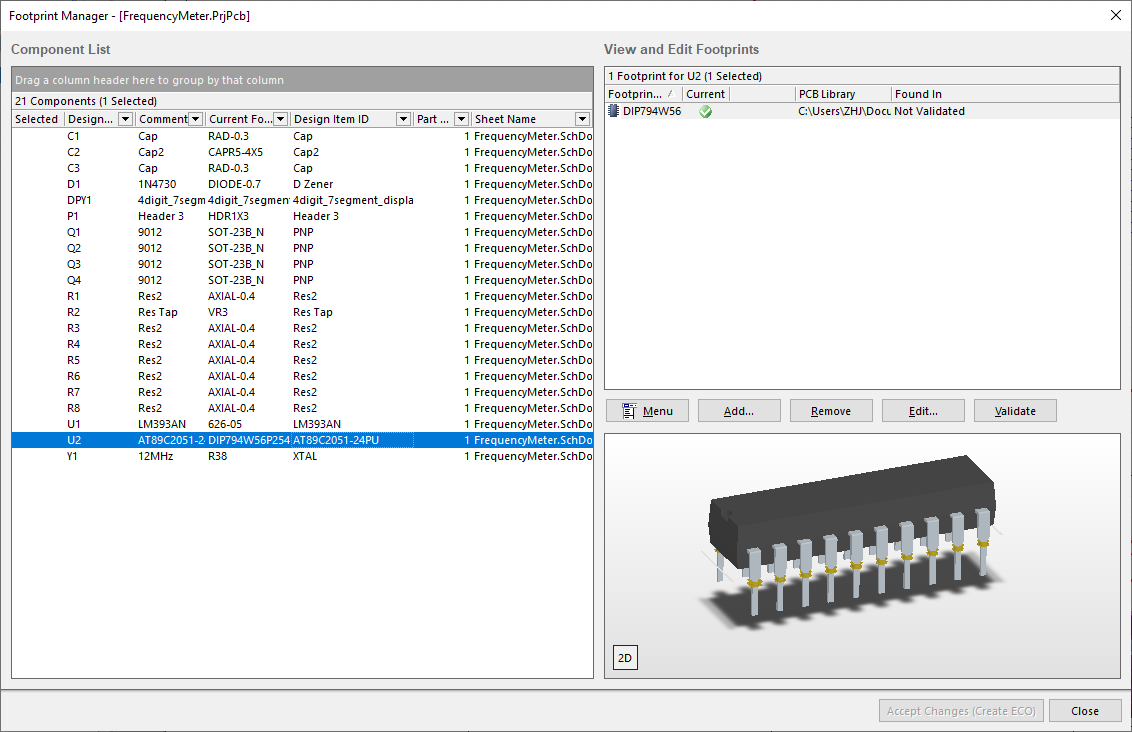


2.2 绘制PCB的准备

绘制原理图前，需要查看当前的封装是否合适。在原理图编辑界面查看工具菜单下的封装管理界面。

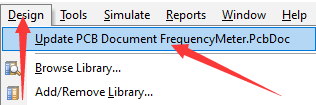


进入封装管理界面，查看各个元件的封装。

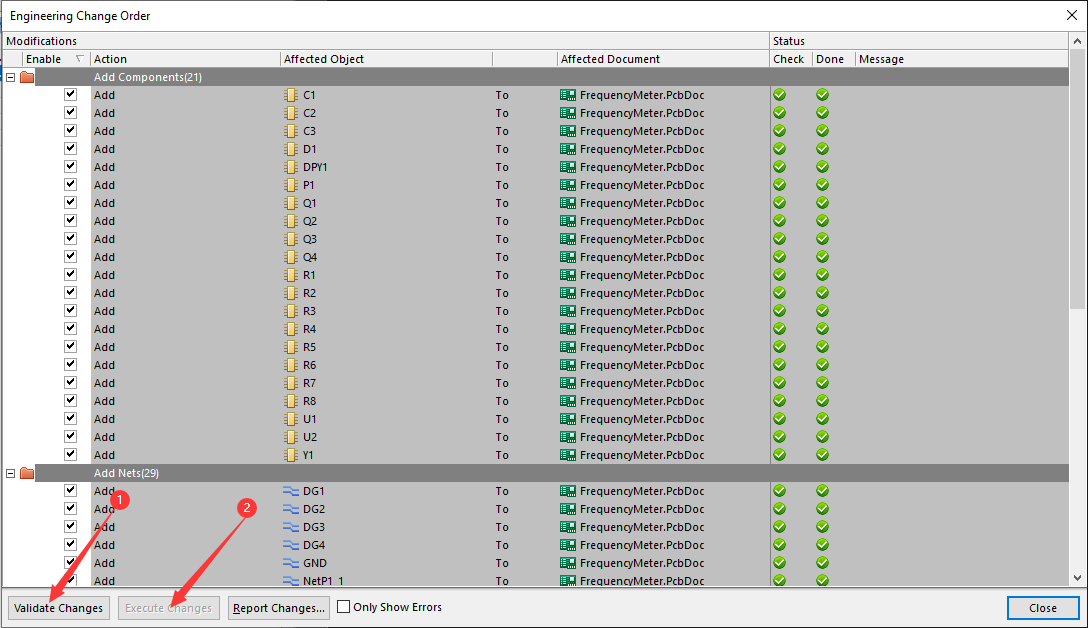


2.3 导入PCB

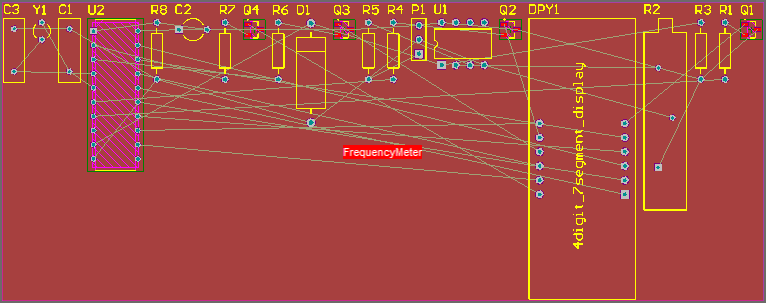
现在根据原理图将原件的封装导入OCB文件中。在原理图界面的设计菜单中，选择更新原理图至PCB文件。



之后会弹出预更改窗口，点击验证改变以验证封装的正确性。之后点击执行更改，将封装导入PCB文件中。

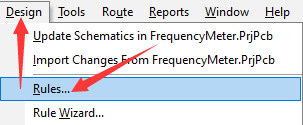


这样，元件的封装就可以导入到PCB文件中了。



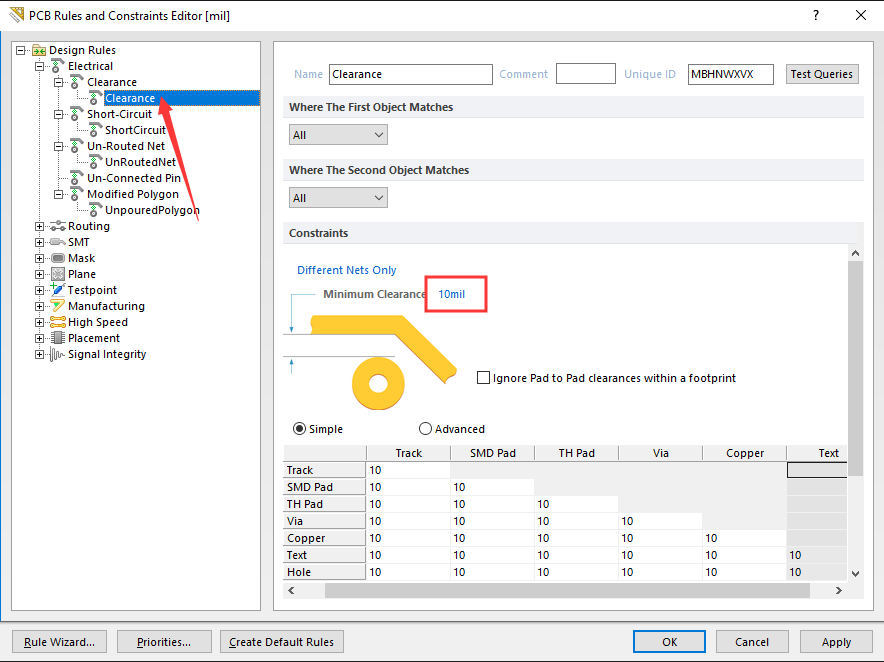
2.3 设置规则

PCB绘制过程最重要的就是规则的制定，下面需要进行PCB布线规则的改写。在PCB绘制界面打开规则设置。



（1）电器规则

电器规则包括安全距离、短路规则、未布线网络、未连接的引脚等规则。首先设置安全距离，将安全距离设置为10mil。任何网络线之间都不能小于10mil的安全距离，否则就会报错。



其他规则设置为：不允许短路、不检查未连接网络，默认即可。

（2）布线规则

布线规则包括走线宽度、布线拓扑、布线优先级、布线工作层、布线拐角模式、布线过孔类型、布线扇形控制、差分线设计，下面逐步进行设置。

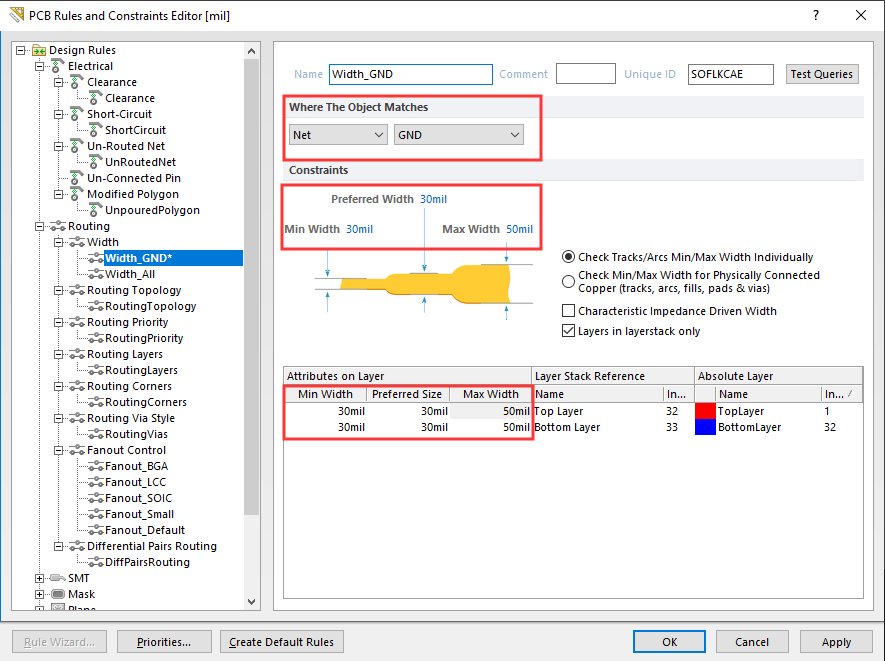
1. 走线宽度

根据课本的建议，信号线的宽度在10~15mil左右，电源线的宽度大约30mil~50mil，那么我就设计两套规则：“默认线宽”在10~50mil，推荐线宽设置为12mil作为信号线默认线宽；“电源线线宽”设置为30~50mil，推荐线宽设置为30mil作为电源线的默认线宽。

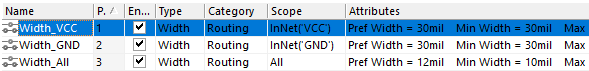
首先修改默认线宽：



之后添加一个地线的线宽规则：



这里设置默认地线宽度为30mil，线宽范围为30mil~50mil。电源线VCC以相同的规则进行设置。下面要设置规则的优先级，将电源线和地线的优先级设置为高优先级，全局布线规则设置为低优先级。

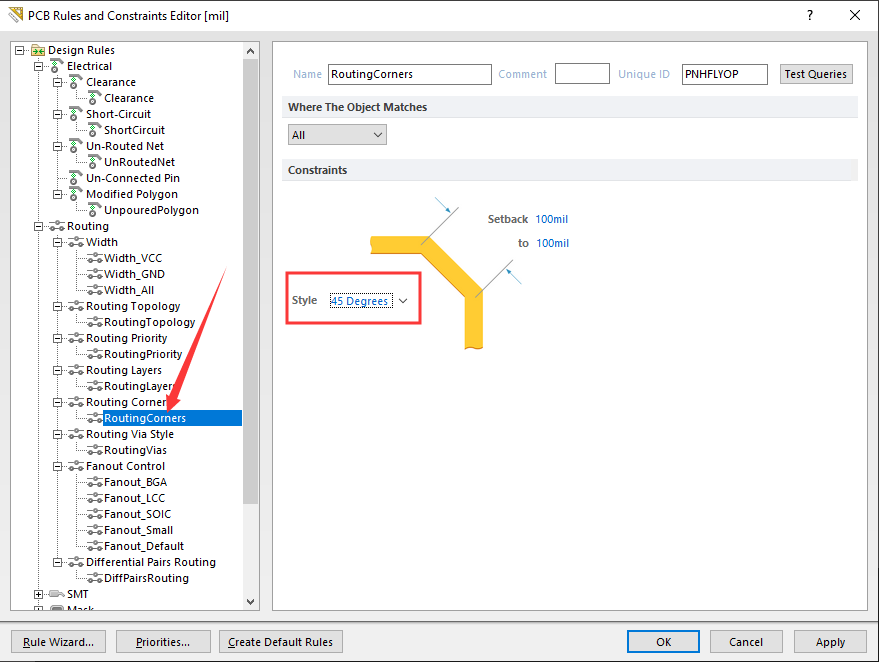


2）拓扑结构

拓扑结构的设置对于自动布线有重要的作用，但是对于手动布线没有太大作用。拓扑结构设置为默认的最短方式。

3）布线拐角模式

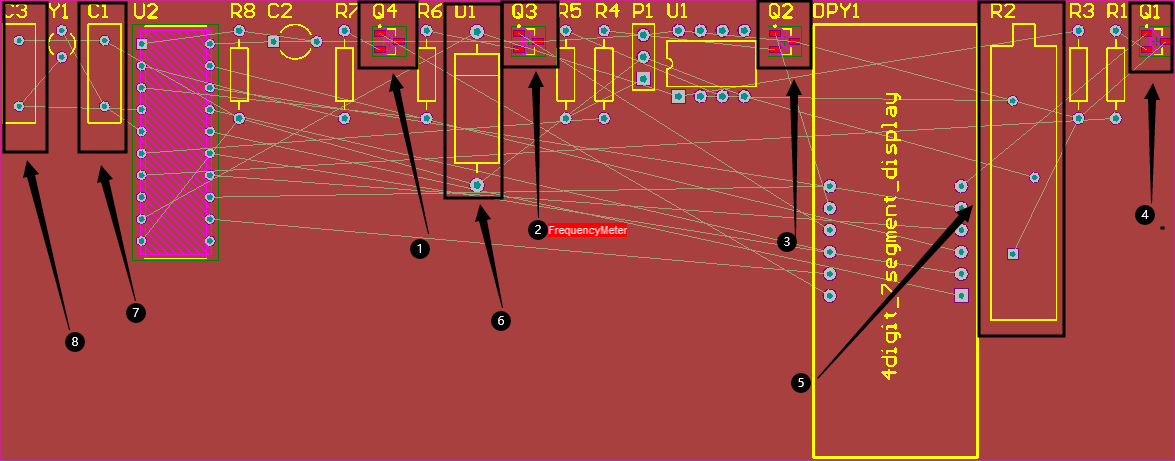
布线拐角模式是在布线拐角时默认的拐弯方式，系统默认的方式是45度拐角，还有90°拐角以及圆角的方式。我设置的是45°默认拐角。



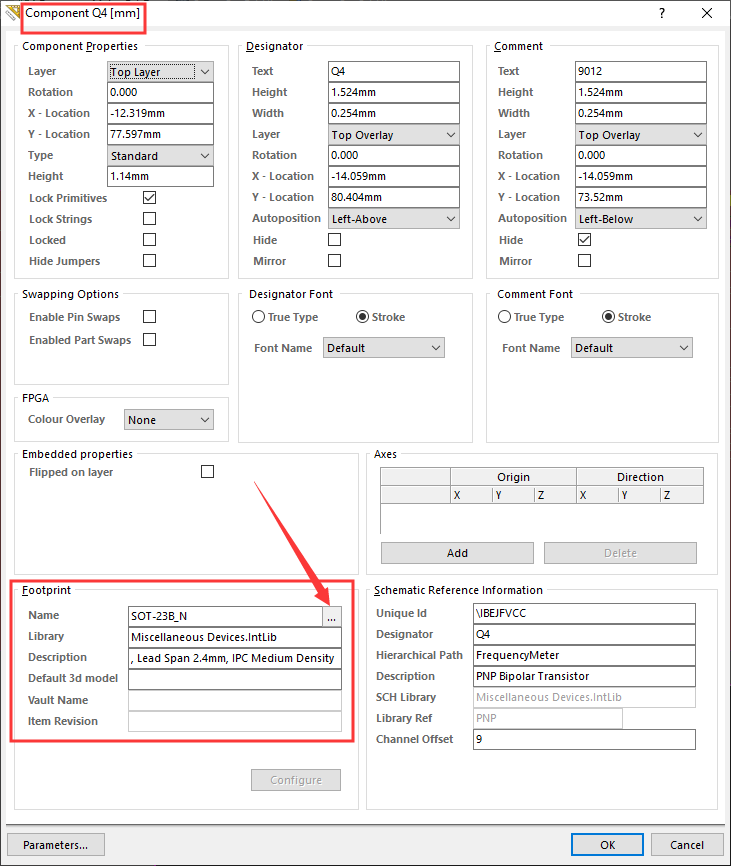
至于其他的规则，因为我没有特殊的要求，所以就设置为默认的选项就可以了。

2.4 原件封装检查

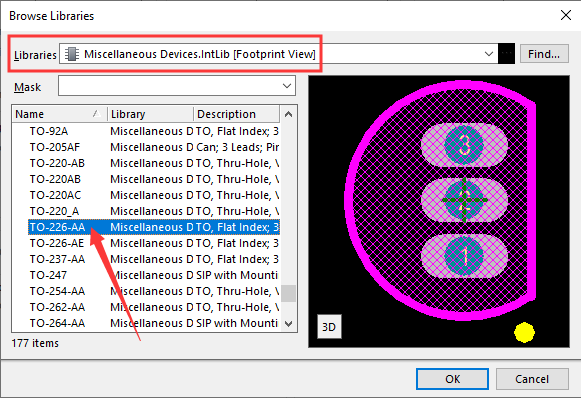
在进行PCB绘制之前，需要首先进行原件的封装检查。以课本的封装为标准，核对我的PCB封装是否正确。经过检查发现，我的四个三极管、一个电位器、两个电容和一个稳压二极管的封装是与课本上不同的。一共八个封装有问题，所以现在就需要改一下封装。



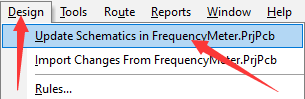
现在开始修改原件的封装。以三极管Q4为例，双击该封装，打开元件设置。在footprint选项中点击按钮修改封装样式。



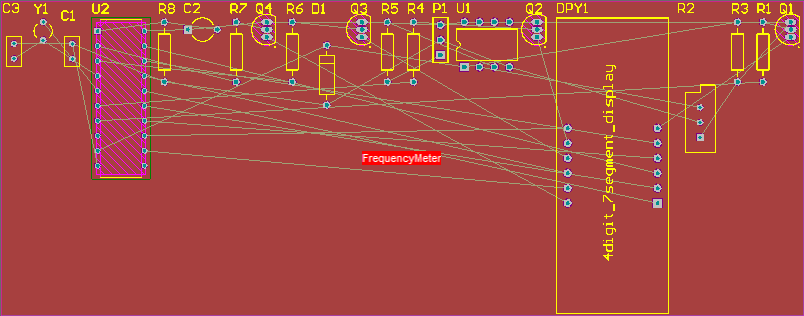
打开封装窗口，选择与课本相同的封装样式。



我选择的是TO-226-AA封装的三极管。下面将剩余的七个封装修改过来即可。改过来之后，需要更新一下PCB工程。



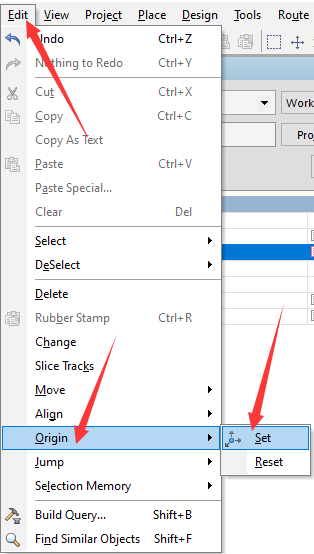
更改之后，原理图上原件对应的封装就更新成刚刚设置的封装了。所有元件封装之后，如下图所示。



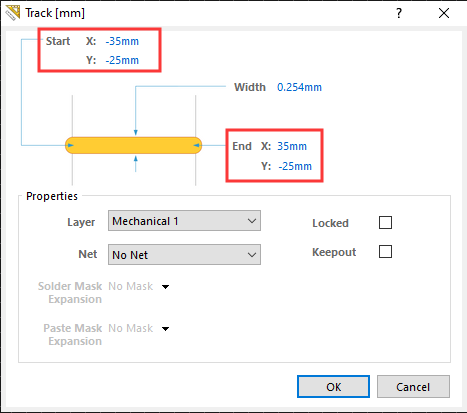
2.5 规划轮廓

规则设置好了之后，需要首先设计PCB的整体轮廓。首先将PCB的规格单位设置为公制单位，然后画出一个70mm×50mm的矩形框，作为PCB的外部轮廓。

先重新设置PCB的原点：

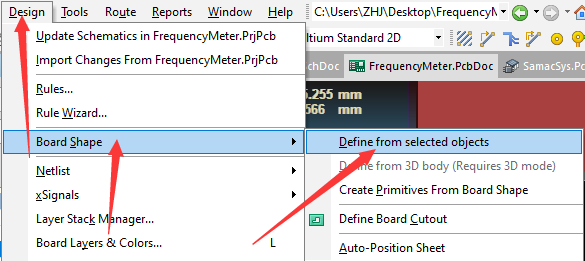


在PCB中放置原点，然后在机械层画一个70\*50的矩形。绘制方法就是先随便画一条线段，然后设置线的起始位置和终止位置。四条线可以使用相似的方式画出来。

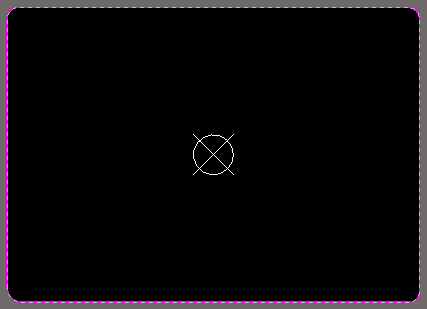


为了更加美观，我将矩形的四个角都做了倒角处理。

现在重新定义板子形状：



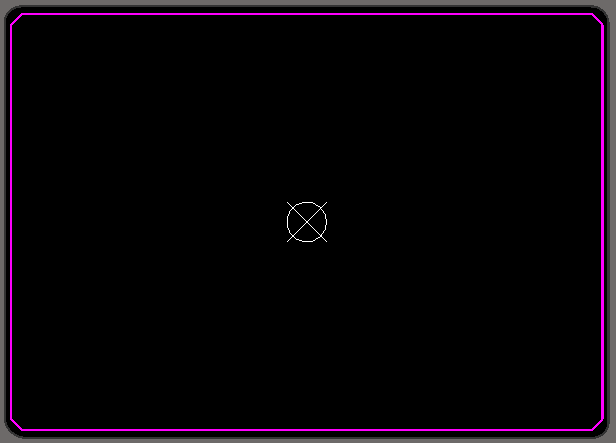
定义完成后，板子形状就变成了经过倒角的矩形。



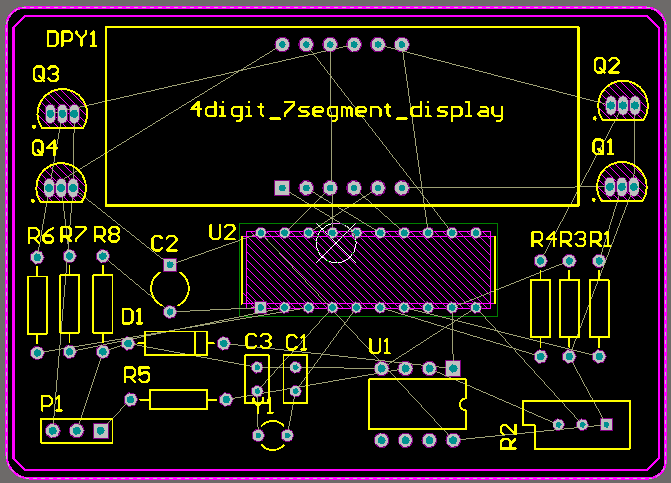
2.6 元件布局

PCB轮廓定义完成后，就可以开始进行元件布局了。按照原件尺寸进行布局，这里还需要考虑原件的大小形状以及相对位置，避免元件之间发生碰撞，还需要考虑是否便于焊接，是否符合使用习惯等等因素。下面我按照课本的示例开始布局。

在布局之前，先在禁止布线层画一个框，以便日后在布线时不再添加。

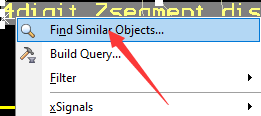


下面开始进行原件布局。

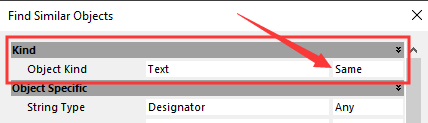


初步的PCB布局如上图所示。

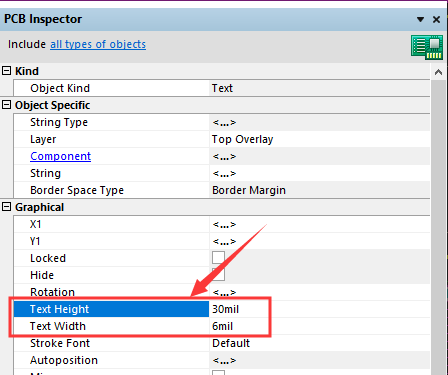
下面首先来调整一下丝印的大小和位置。



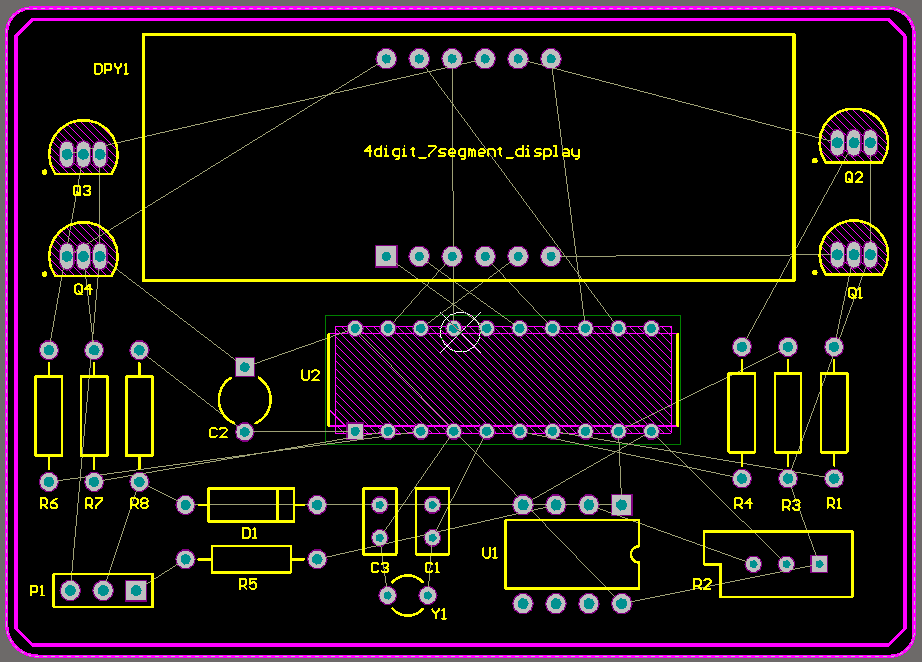
查找所有丝印，调整他们的大小。



这样所有丝印的文字就已经被选中了，现在修改丝印文字大小为6mil×30mil



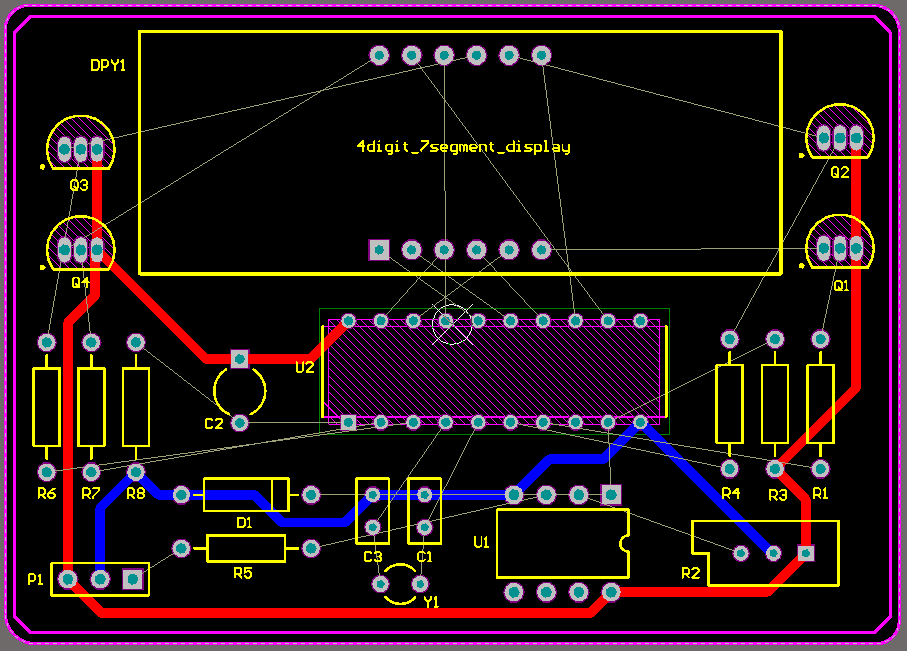
现在所有丝印文字都变成6mil×30mil了。重新调整一下丝印布局即可。



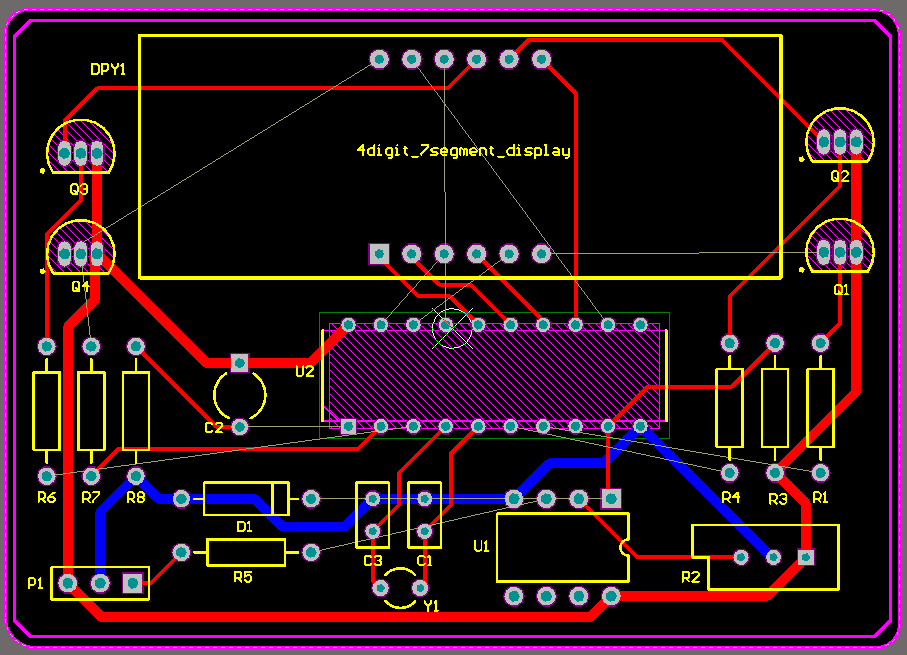
上面图中是丝印修改完成后的PCB布局。

2.7 原件连线

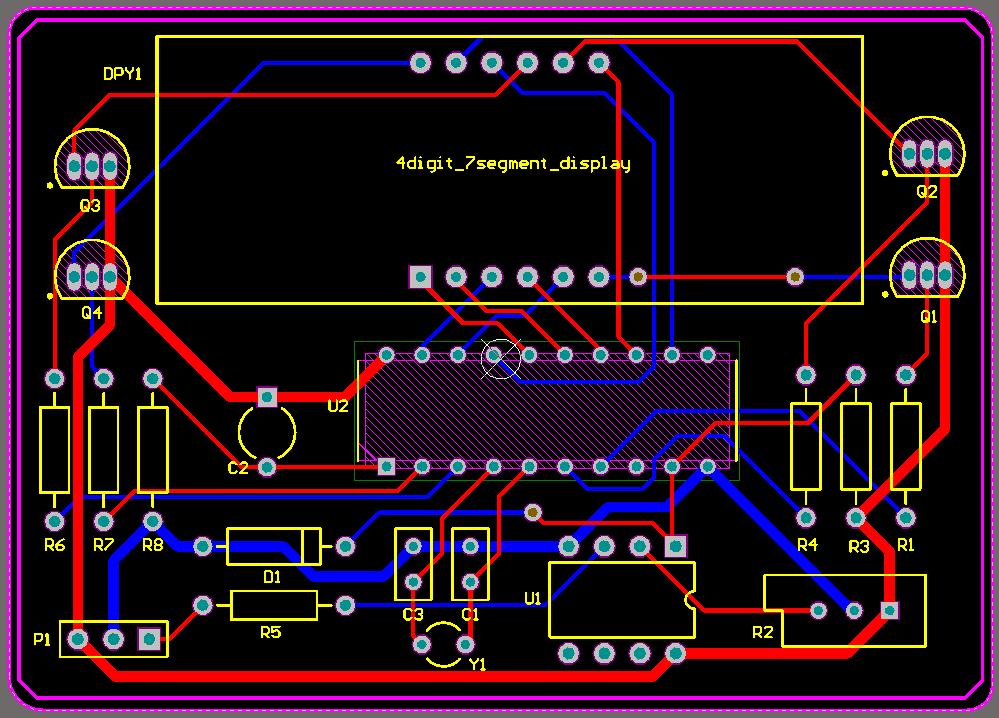
首先绘制VCC和GND的连线。



然后在顶层绘制其它信号线。



顶层信号线布局如上图所示，接下来在底层继续绘制还未连接的信号线。



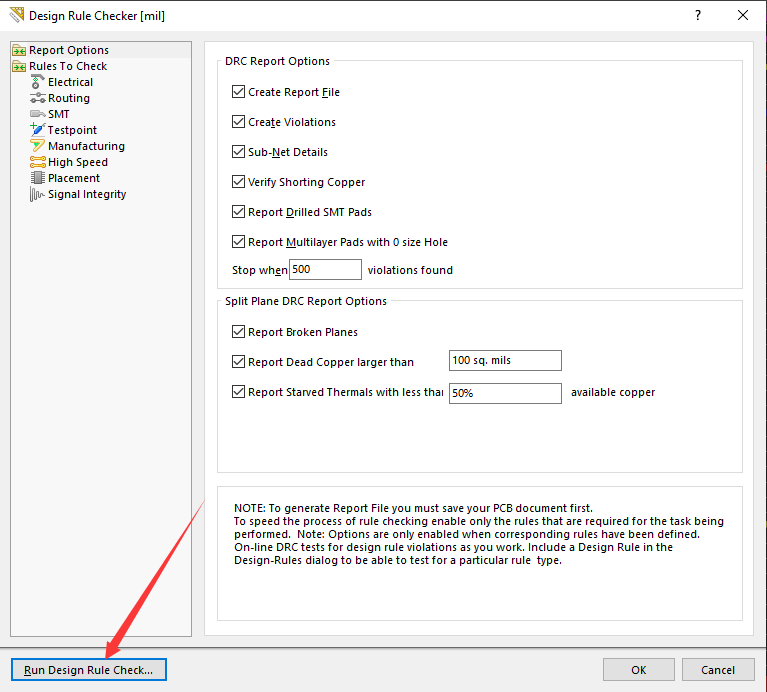
在连接底层信号线时，有三处使用了过孔。最终信号线布局如上图所示。

2.8 DRC检查

PCB原件连接完成后，首先进行一次DRC检查。



打开DRC检查窗口



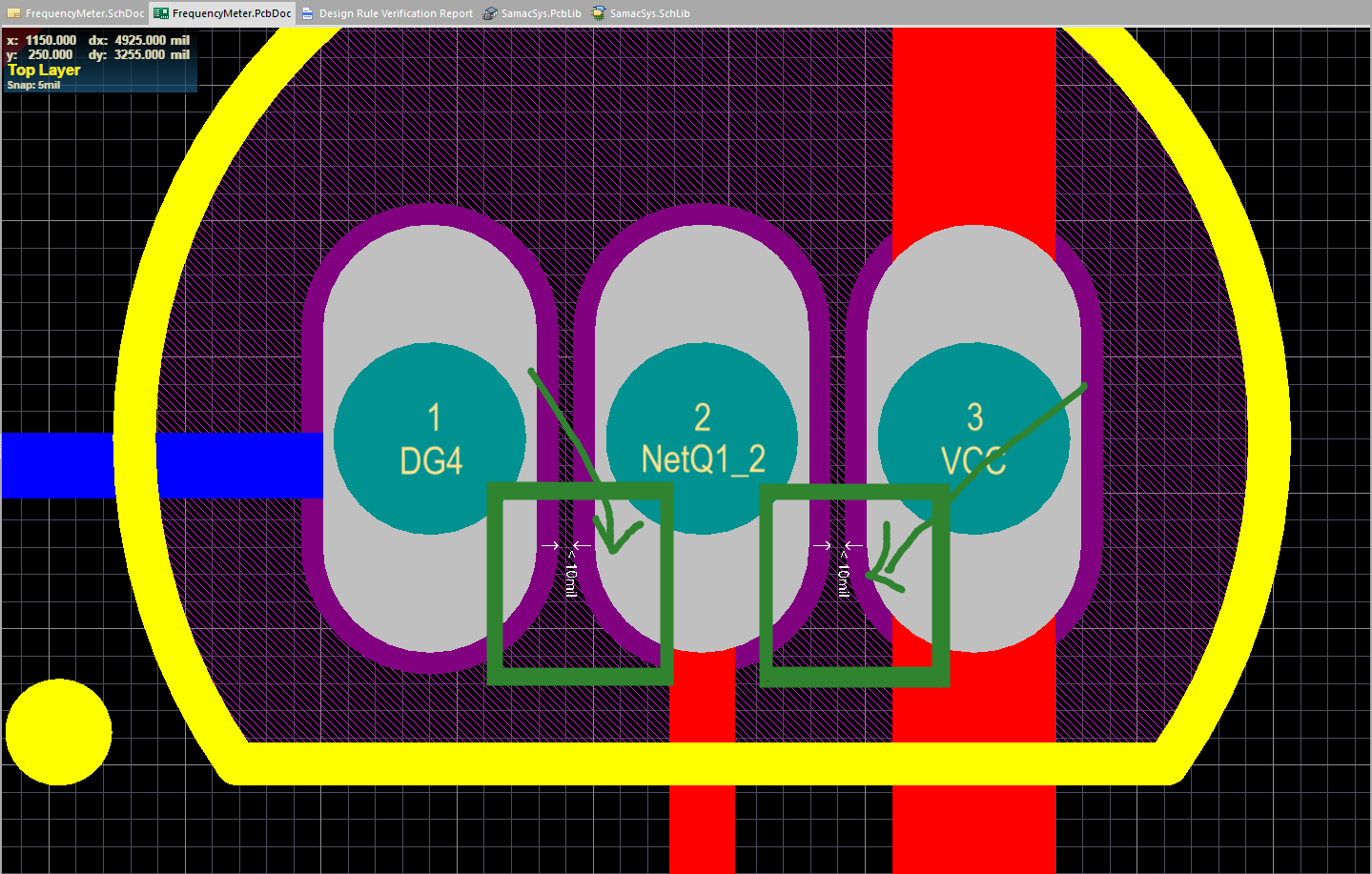
运行设计规则检查后，我的PCB被检查出有24处违反规则。



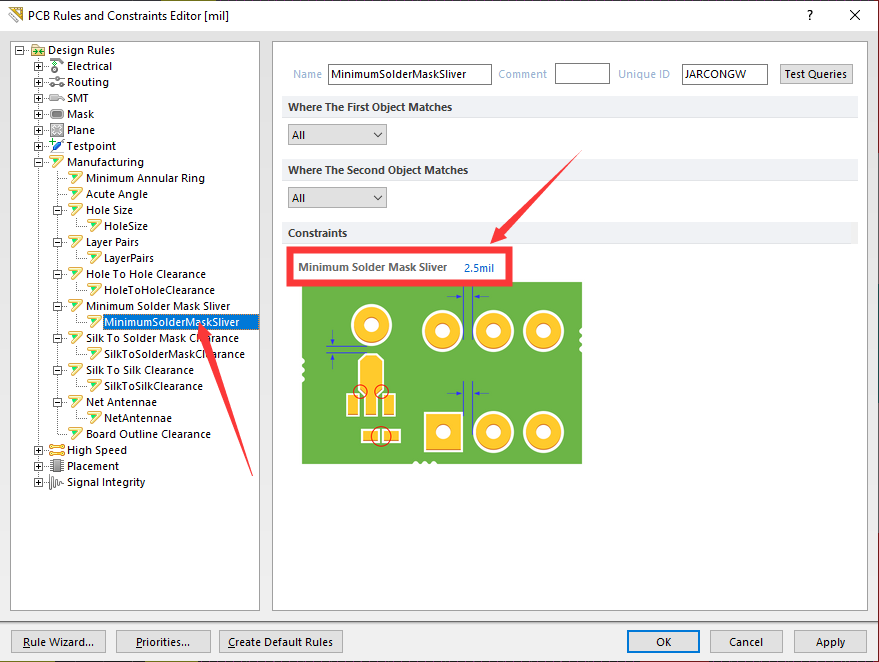
现在逐一排查这些错误，先看第一个。



双击该错误，将会跳转到PCB对应的位置。



这个报错的意思是说：这两个阻焊层之间的间距是2.63mil，小于最小规则间距10mil，所以报错。这个地方应该不算大问题，可以直接忽略。为了使其不在报错，可以修改DRC检查的规则。

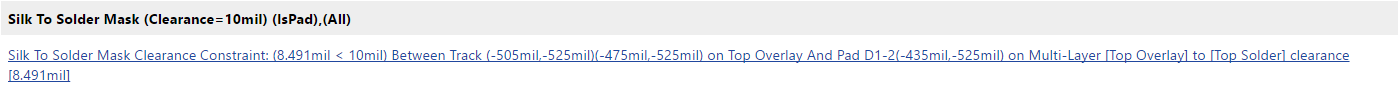


因为之前看到阻焊之间的距离是2.63mil，所以我将规则的最小距离设置为2.5mil。更改完成后再次进行DRC检查。

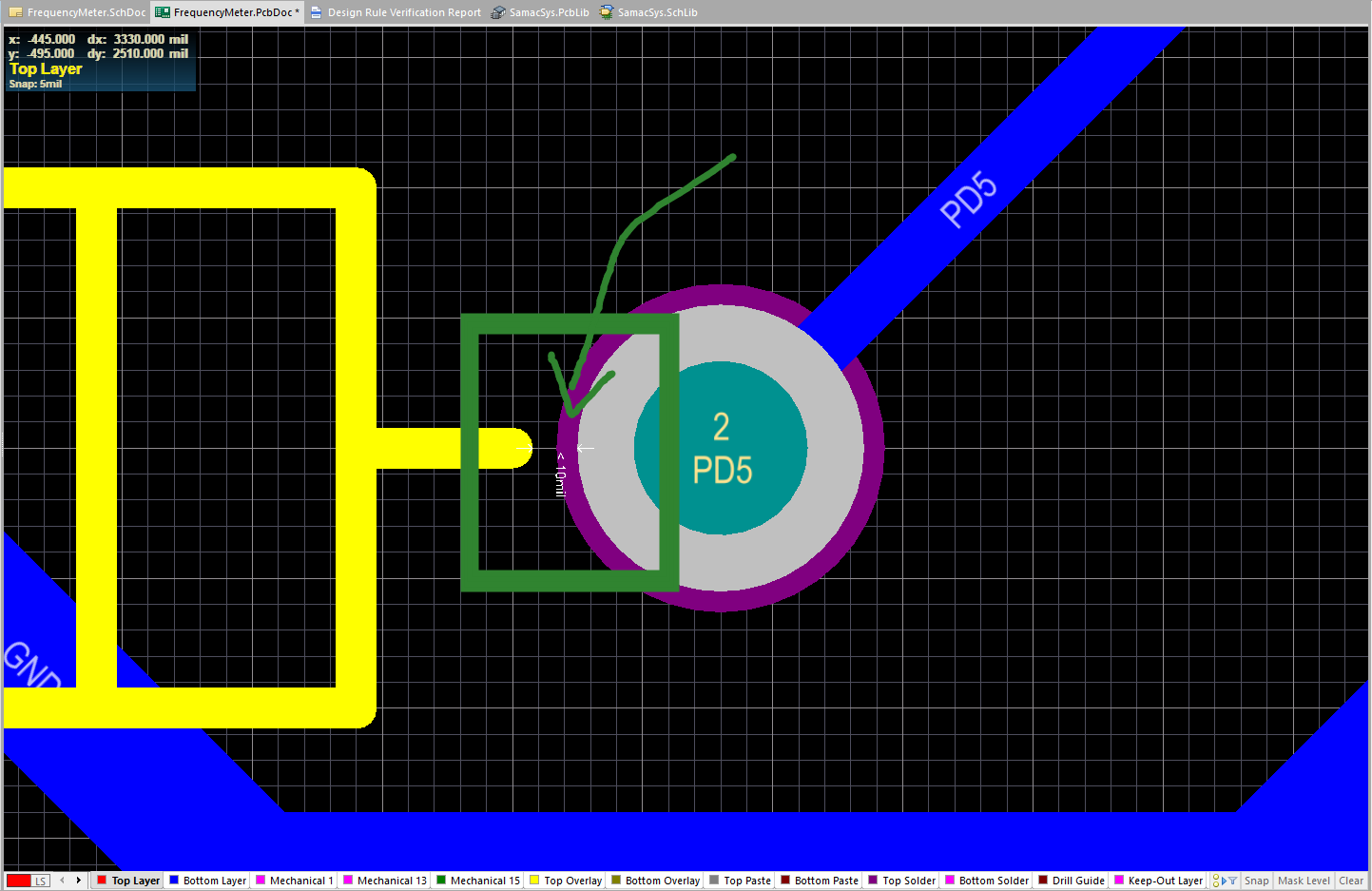


可以看到，错误已经减少到了16处。下面继续进行PCB调试。

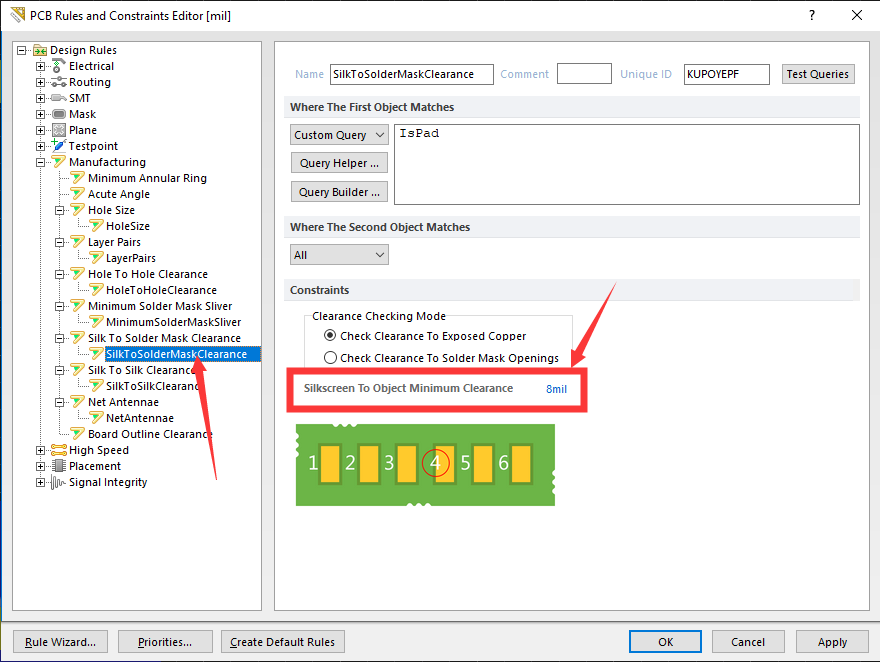
以这16处错误中的一处为例，其报错信息如下。



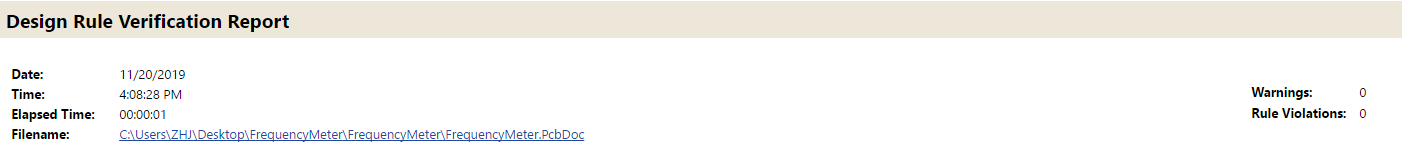
在PCB中的错误定位如下：



可以看到，错误原因是：丝印到阻焊之间的距离小于规则的最小间距。于是继续修改PCB规则。



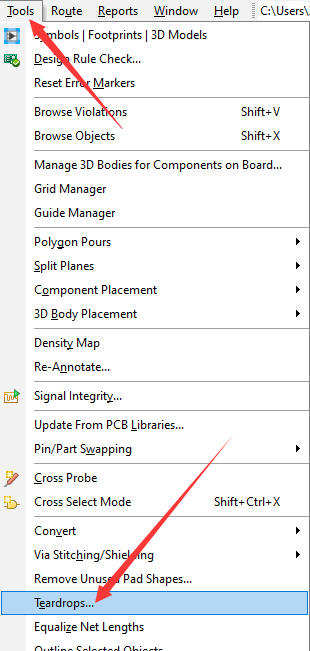
我将最小规则间距设置为8mil，然后再次进行DRC检查。



好了，现在DRC检查通过了。

2.9 铺铜和补泪滴

为了减小PCB信号之间的干扰，更好地发挥PCB性能，需要给PCB进行覆铜处理。在铺铜之前，先进行一个补泪滴操作，目的是使焊盘和信号线之间能够更好地连接。

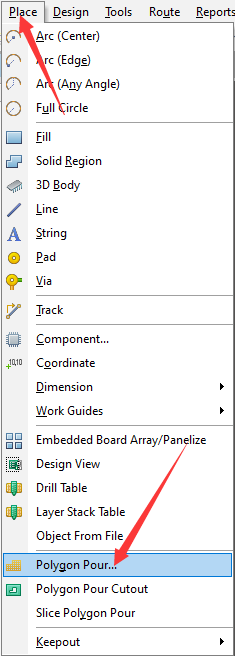


打开泪滴菜单

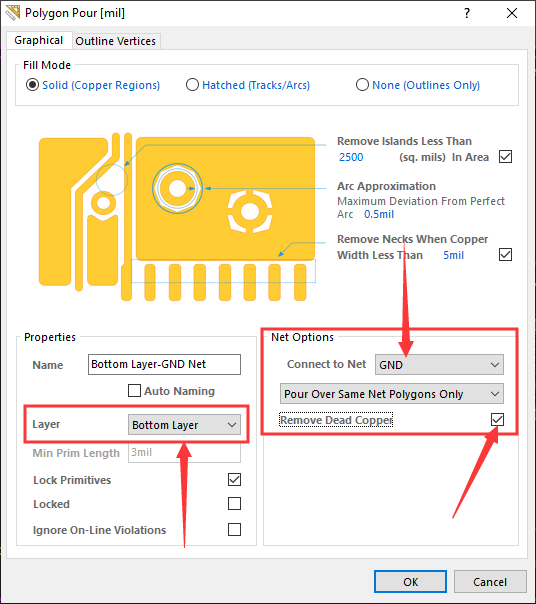


选择默认的选项即可。补泪滴完成后，界面会自动生成泪滴的信息报表。

补泪滴结束后，进行覆铜操作。



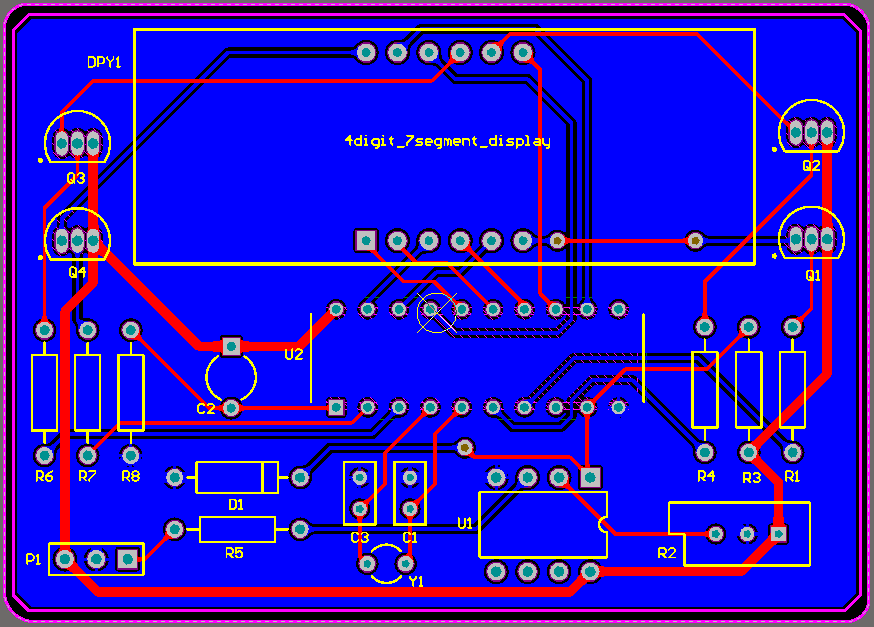
打开覆铜界面



选择在底层覆铜，连接到GND网络，同时去除死铜。

接下来就是选择覆铜区域，直接选择整块版就可以了。

覆铜结束后的效果如下图所示：

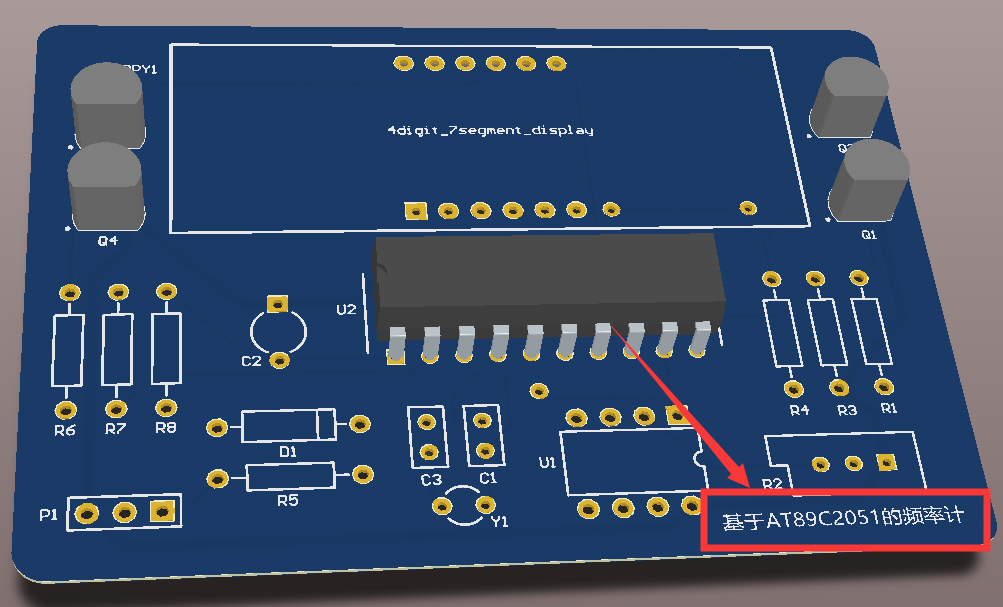


因为只有底层有覆铜，所以整块板子呈现大片蓝色。顶层的红线是VCC和一些顶层的信号线。

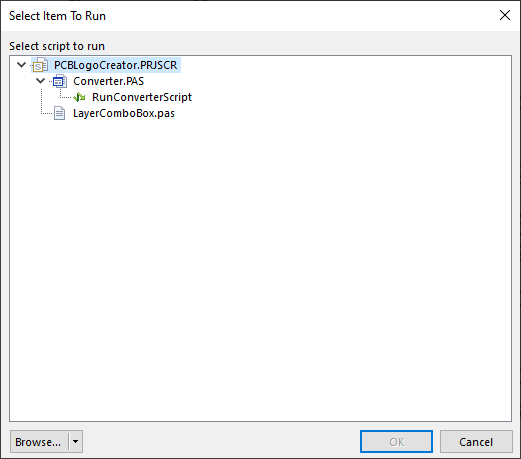
保险起见，需要再次进行一次DRC检查，确保PCB覆铜之后没有问题。我进行DRC检查后没有问题。

2.10 添加丝印

在PCB上添加丝印可以帮助我们了解该PCB的功能、设计时间等信息。首先找到一块合适的位置，添加一些字符串表明该PCB的功能信息以及制作信息。我在PCB的右下角添加了PCB名称。

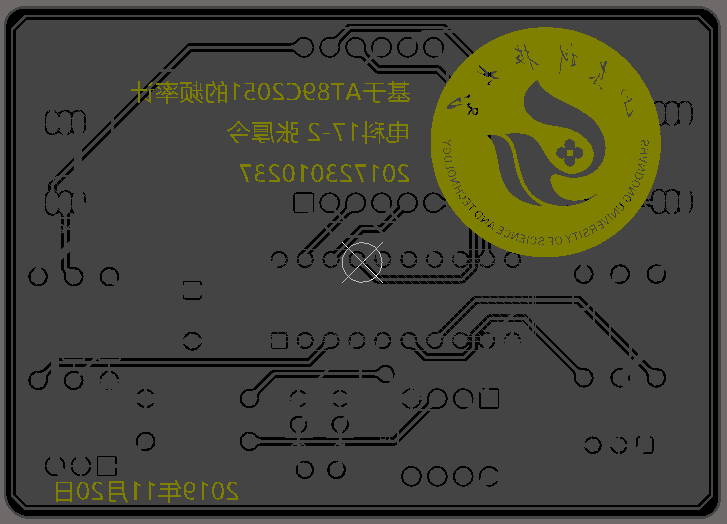


下面我想往PCB中添加一个图片，需要运行一个AD脚本，创建图标。





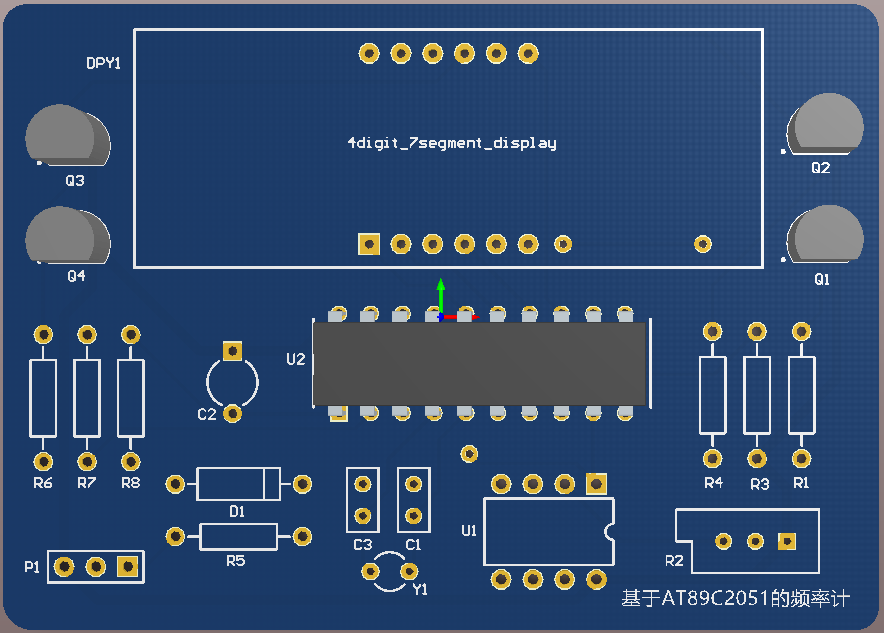
图片转换完成后，在底层添加该logo，同时添加我的个人信息。



注意这里的字符串和图片都需要是经过镜像的，因为这是我们以前面的视角观察底层的视图。实际的背面效果如下图所示：



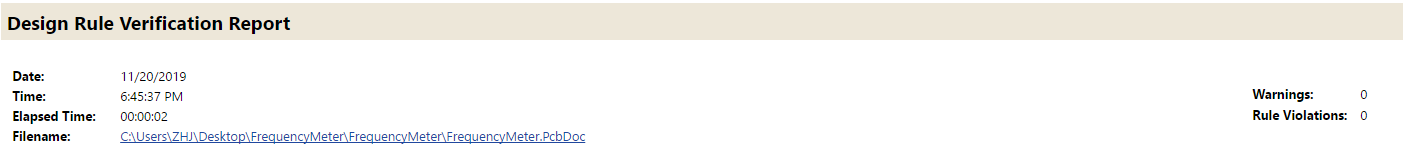
实际的前面效果：



这里我进行了一次DRC检查，发现有95处错误。错误原因是那个图片logo放到焊盘上面了，丝印与焊盘之间的间距太小，所以报错了。为了避免图片丝印覆盖焊盘，我将图片的大小和位置进行了调整，最终得到以下设计：



再进行DRC检查，不再报错。



2.11 生成工程文件

首先利用smart PDF生成PDF文件。

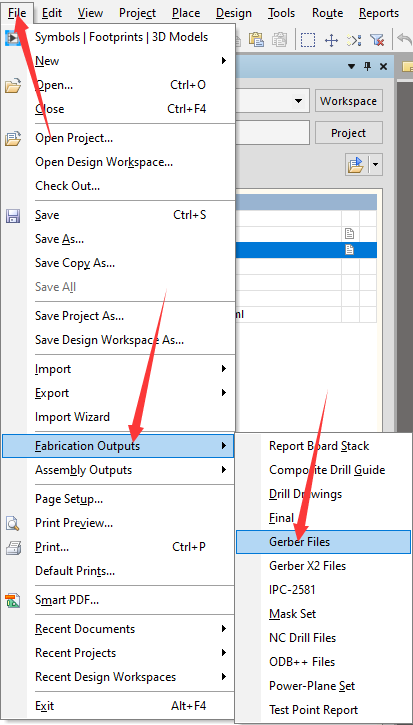


生成的PDF包含了每个元件的详细信息，但是有些部分没有显示出来。虽然不影响阅读，但效果还是不太好。

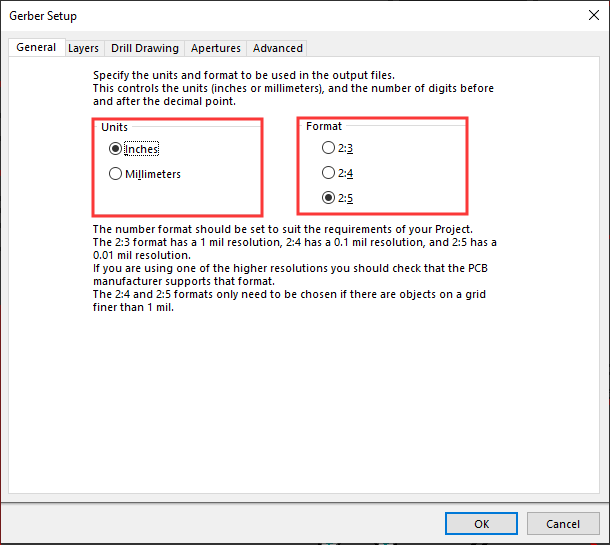
三、打板

经过多次检查，我认为板子已经没有问题了，于是准备生成文件打板。通常需要给工厂发送Gerber文件。Gerber文件是通过PCB文件生成的，工厂可以通过该文件直接进行生产。下面来生成Gerber文件。

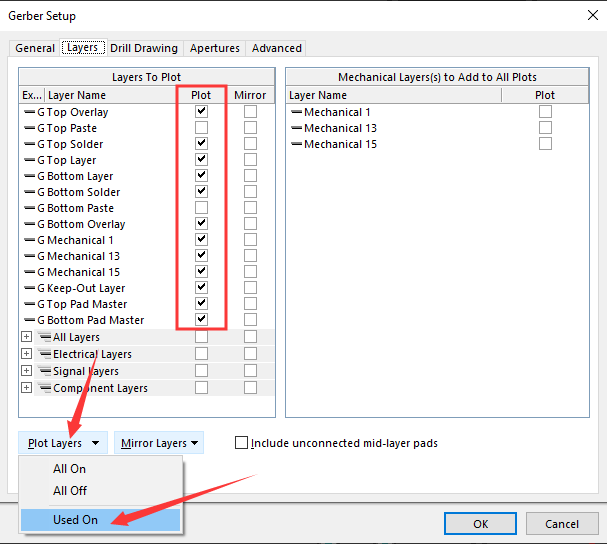
3.1 Gerber文件生成



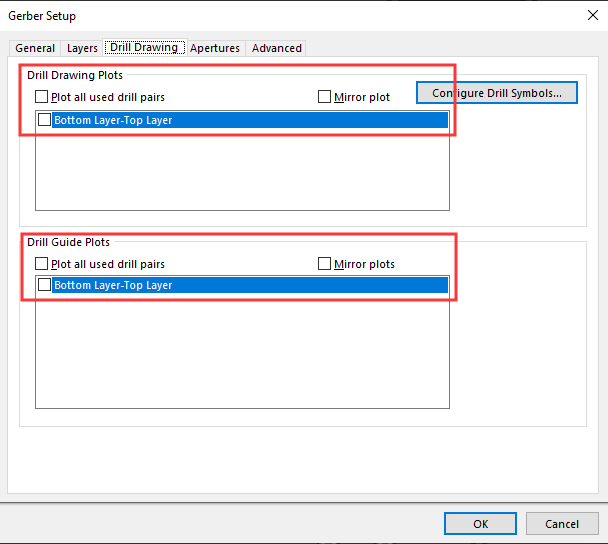
首先设置文件的单位和精度。



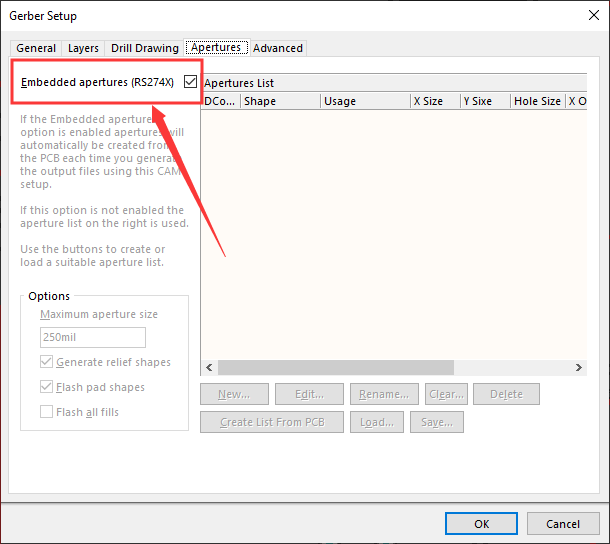
之后设置版层输出。



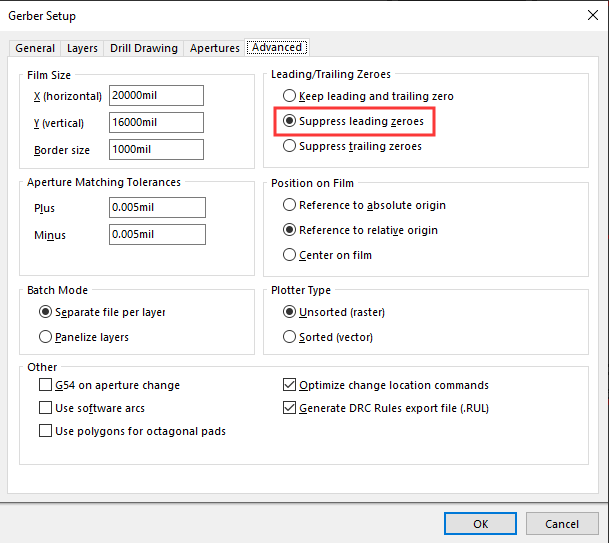
这里只输出用到的版层，同时机械层都不选。接着设置钻孔图层。



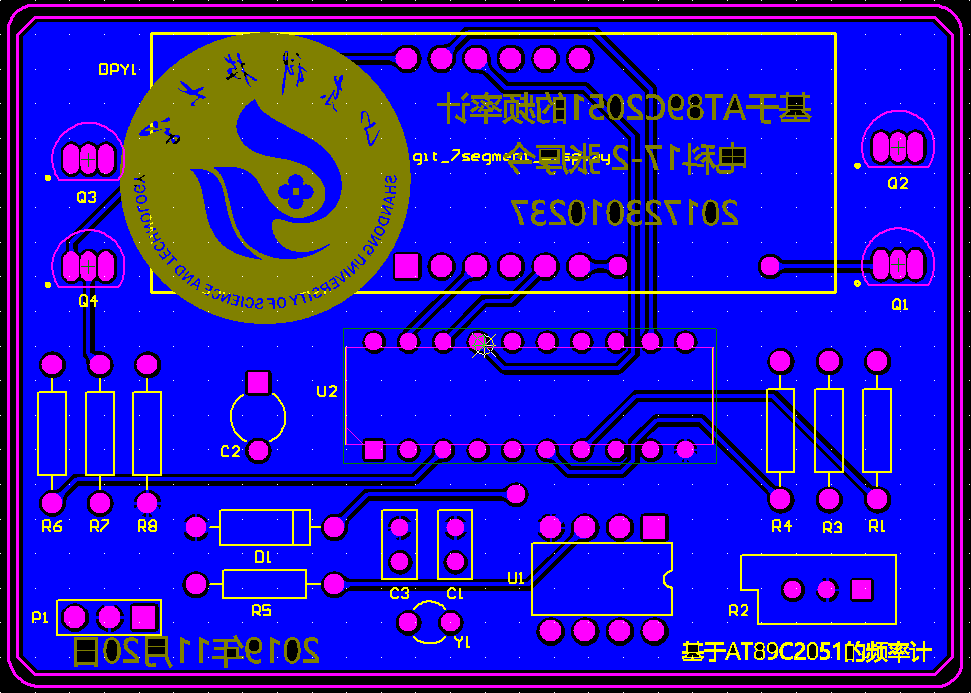
这里全都不选。然后设置光圈。



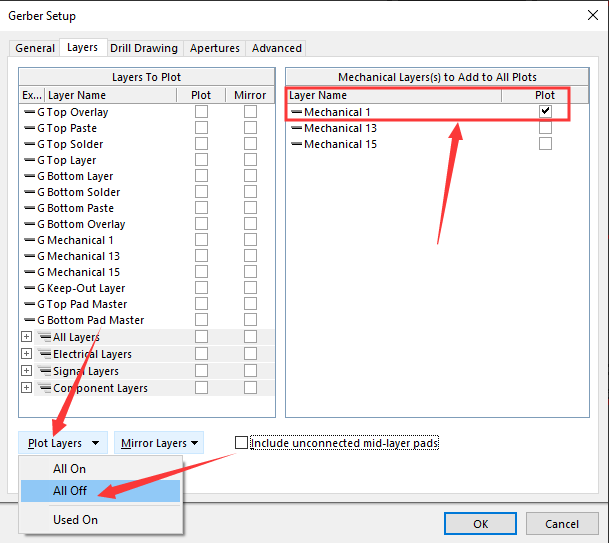
这里的嵌入式孔径选项要打勾，之后进入高级设置。



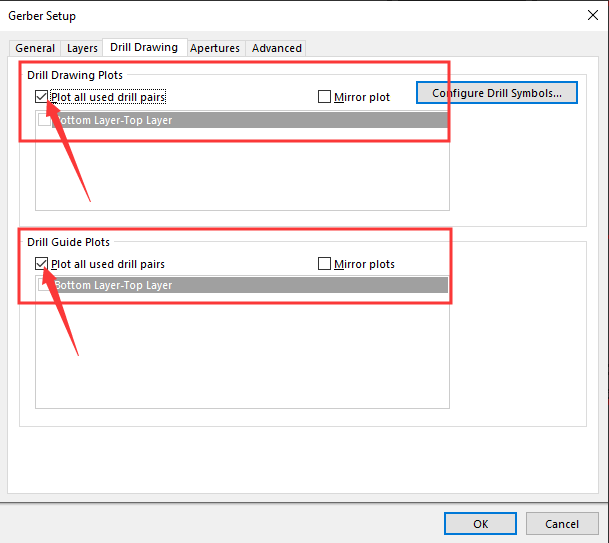
高级设置直接使用默认即可。点击确定生成相应的文件。此时会弹出生成的预览文件，该文件不必保存，直接关闭即可。



接下来再生成一些文件，同样是打开Gerber文件菜单，选择图层。



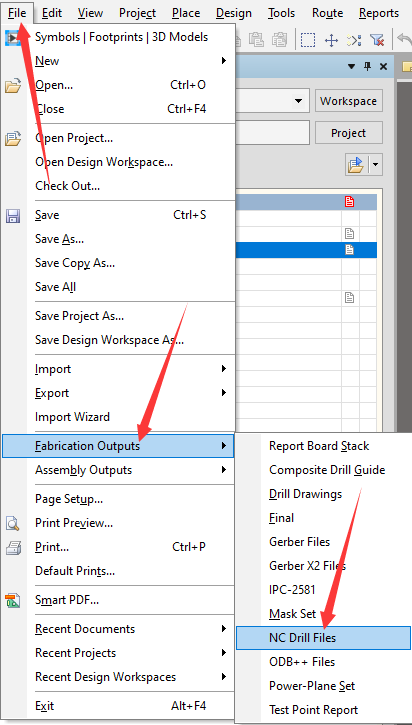
将版层全部关闭，再将机械层1打开。然后设置钻孔属性。



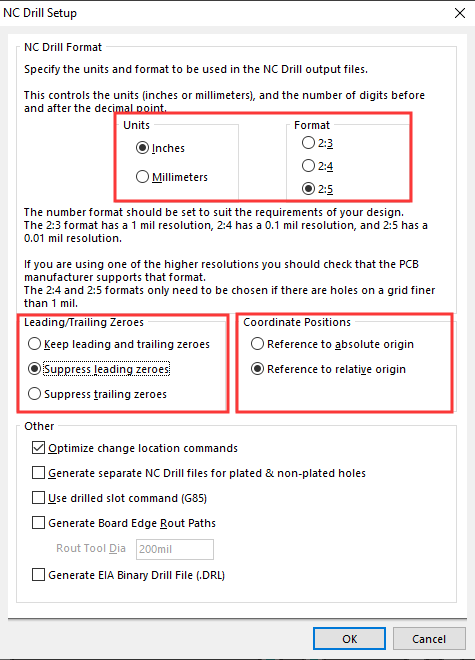
这里要把这两处都选中，表示输出我们使用的所有孔。后面就按照默认设置，再次生成Gerber文件。



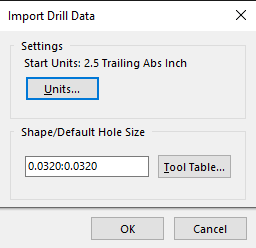
这次生成的文件表示打孔的位置。该文件不包含孔洞的大小和形状，只起到定位的作用。下面生成钻孔文件。

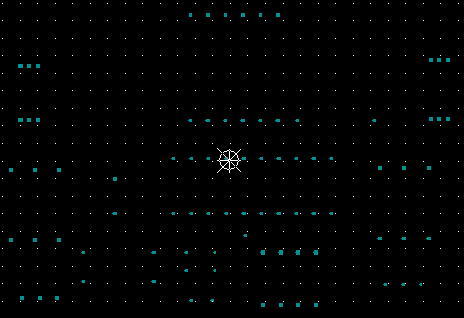


钻孔文件生成的属性，按照刚才生成Gerber文件的属性填写。



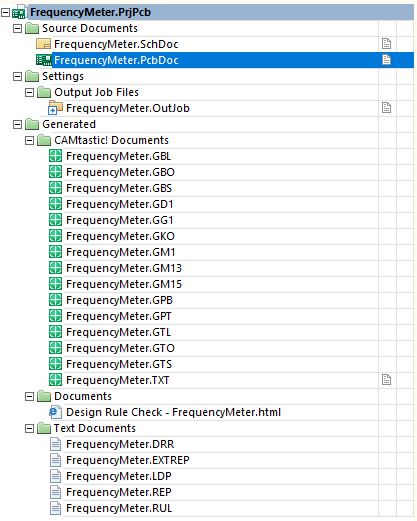
点击确定后，需要导入钻孔数据，直接点击确认，使用默认属性即可。





好的，现在Gerber文件就已经生成完毕了。

现在整个工程的目录结构如下图所示。



现在，Gerber文件就已经导出到了指定文件夹中。