

ALTIUM DESIGNER 软件设计及应用实战

电子发烧友运营战队整理

<http://bbs.elecfans.com/>

目录

一、 Altium Designer 这些不同版本的软件之间的区别有多大呢？	2
二、 Altium Designer 软件功能介绍	8
三、 Altium Designer PCB 电路板设计步骤与技巧介绍	14
四、 Altium-Designer-PCB 布局布线及规则设置	19
五、 方案设计中如何运用 Altium Designer 拼板？	38
六、 AD811 的应用实例	45
AD811 的应用实例	46
1、 视屏线路驱动器	46
2. 视频键控器电路	48
3. 同步脉冲分离装置电路	51
4. pic 单片机的 ad 转换	53

一、Altium Designer 这些不同版本的软件之间的区别有多大呢？

在电子产品设计领域，Altium Designer（简称 AD）几乎是无人不知无人不晓。和 Cadence Allegro、Mentor EE 等软件相比，虽然功能上有所欠缺（主要是 PCB 仿真等功能），但是具有灵活的操作方式，非常易学易用。

然而 AD 虽然易学易用，但是大多数人并没有掌握到 Altium 使用方法的精髓。从网上可以看到，目前比较多的 Altium 版本有 AD9、AD10、AD13、AD14、AD15、AD16、AD17、AD18、AD19 等（其他 protel、DXP，AD6.9 之类老掉牙的就不说了）。那么这些不同版本的软件之间的区别有多大呢？

AD19.0.6

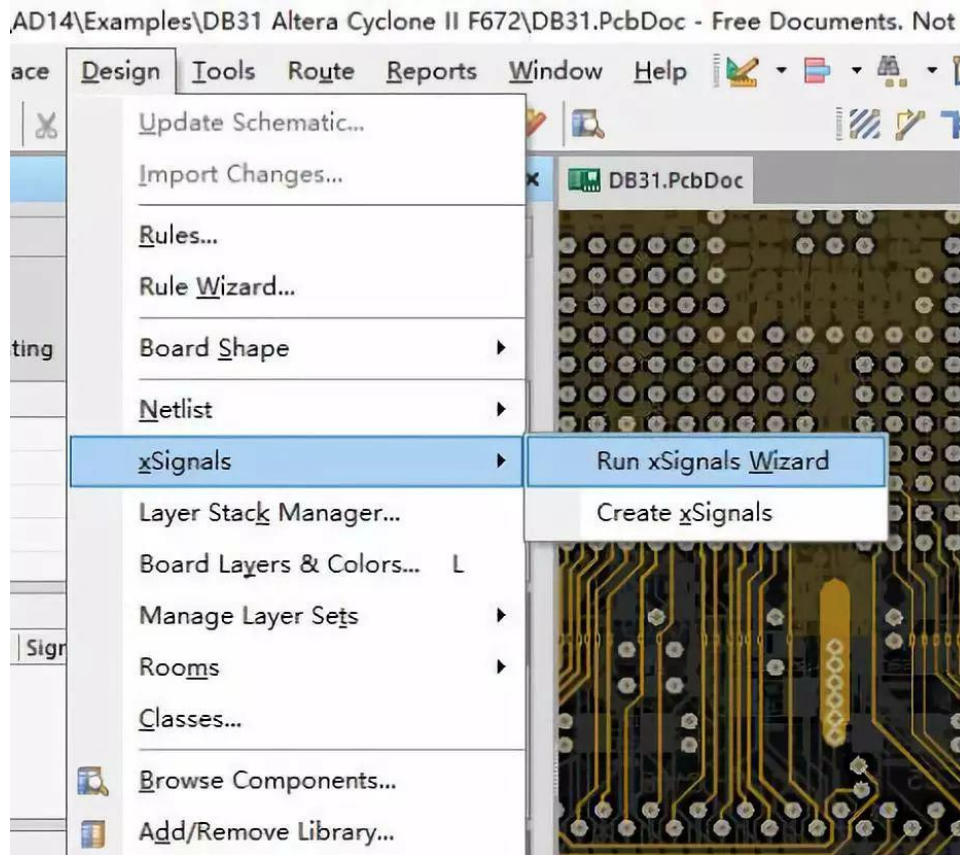
显然，软件版本越高，功能越齐全，可能存在一些未知的 bug。在 AD18 发布之前，各个版本的软件区别不是特别大。比较重要的区别是 xSignals 功能的出现（AD15 及以后的版本才有）。

简单的说，xSignals 功能类似 Cadence Allegro 的 xnet 功能。有时我们会遇到需要绕等长的高速信号线串接了电阻的情况，实际上计算等长的时候应该是以电阻两端长度的总和为准。

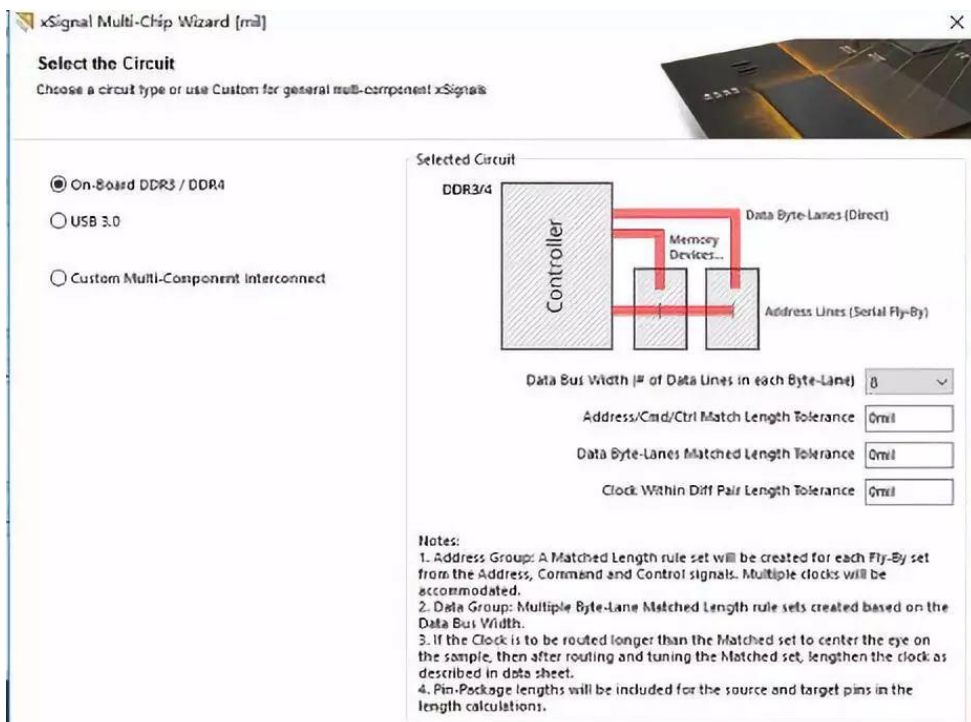
在没有 xSignals 功能之前，一般的解决办法是在 PCB 上将串联电阻两端改成相同网络，这样就可以正常的用软件进行等长了，绕完线之后再网络改回去即可。

xSignals 功能出现之后，可以将串接有电阻的信号线设置成 xSignals，软件会自动识别这种网络，等长绕线的时候会自动将电阻两端的长度相加，这样就节省了手动改网络的时间，提高了效率。我们可以手动创建

xSignals，或者用向导来创建需要的网络分组。



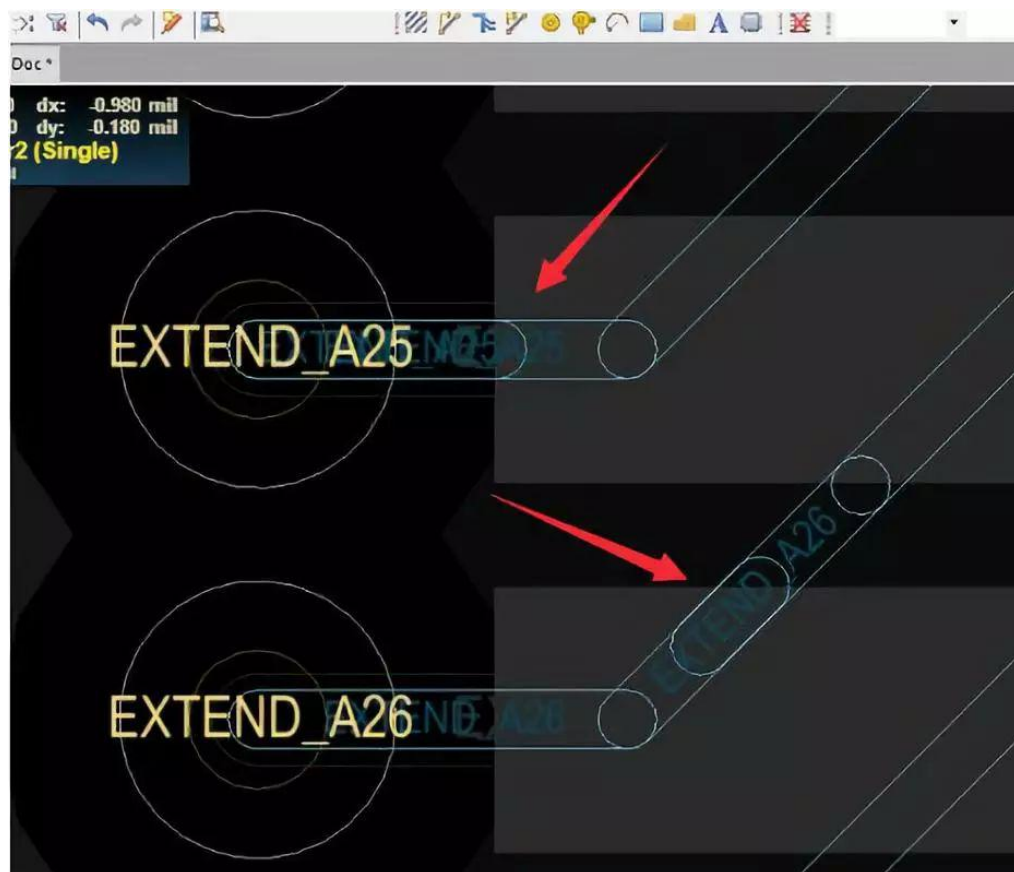
xSignals 功能菜单选项



xsignal 向导

另一个重要的区别是信号长度算法的改进，提高了高速信号线的等长精度。在 AD17 之前版本的软件里，计算信号线布线长度的算法并不严格。具体就是当 PCB 布线的线段有重叠部分时，软件会将重叠部分也算进信号线的总长度。

如果忽视这个问题，当有些线头遗留在过孔、焊盘，或者有线段重合时，这些重叠的长度会导致等长结果有很大的误差。因此用老版本 AD 绕等长时，在等长之前，需要先将 PCB 信号线设置成透明模式，在透明模式下可以看到重叠的线段，将他们删除，然后等长才能保证绕线结果准确。

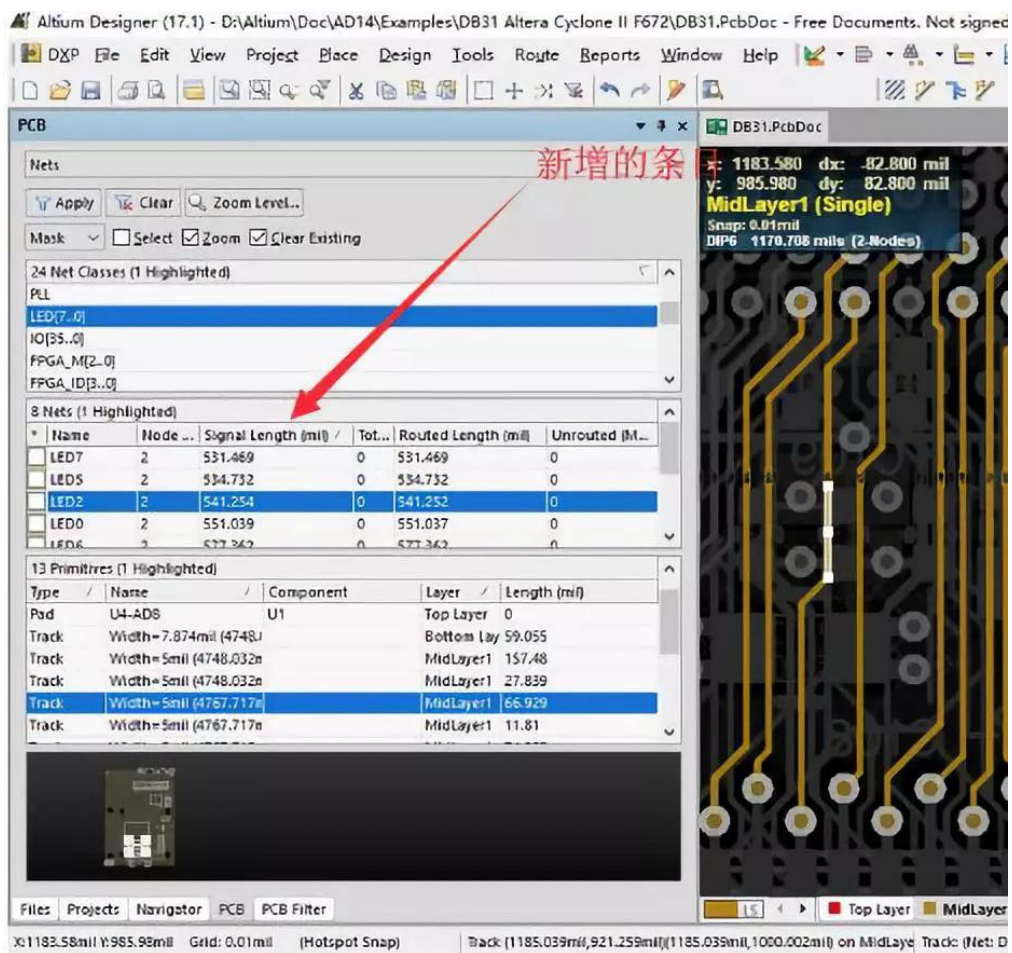


透明模式下看到多余的线头

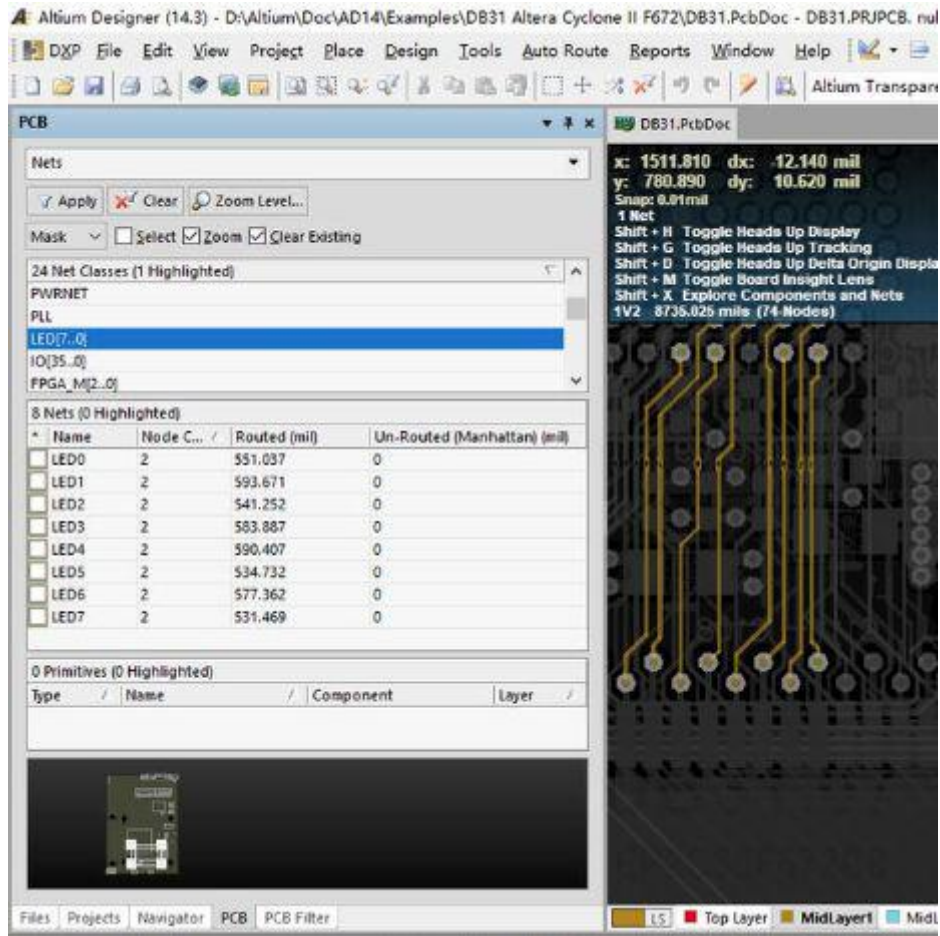
AD17 发布之后，再也不需要这样的操作。因为高版本的软件增加了一个 signal length 的概念。线段重叠、多余线段叠在焊盘、过孔上也不会被计

算在内，这才是真正的信号长度。

用 AD14 和 AD17 打开同一个 PCB 就可以发现差别，可以明显的看到 AD17 的表格里多了一列 signal length。设置等长规则时，软件就是以这个项目的值为标准进行绕线。

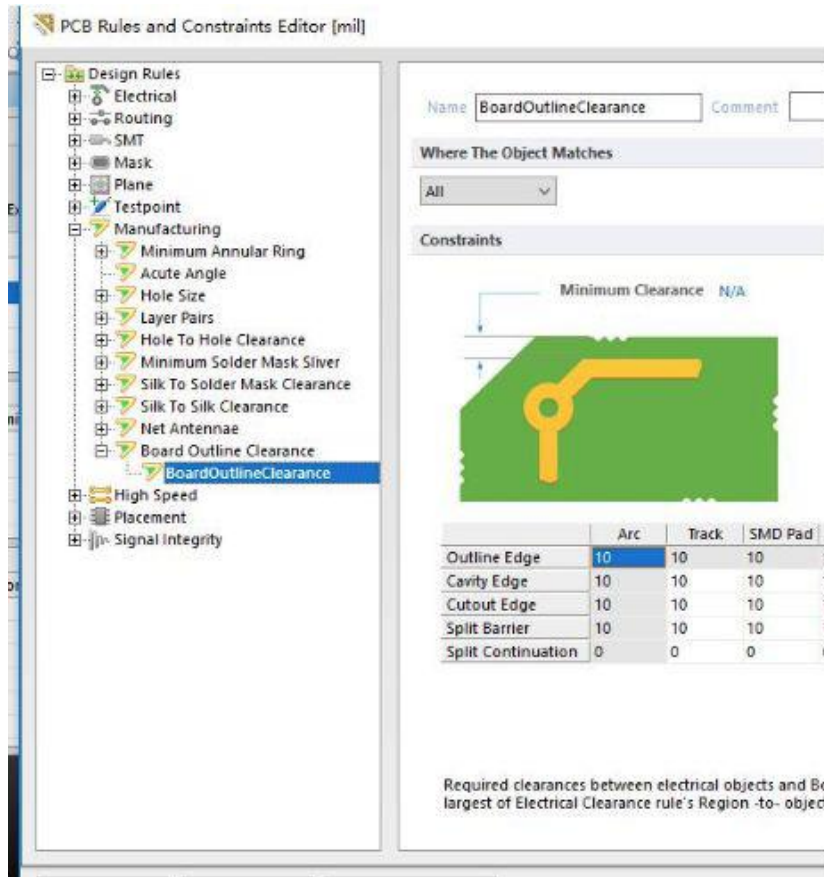


AD17 显示的信号长度



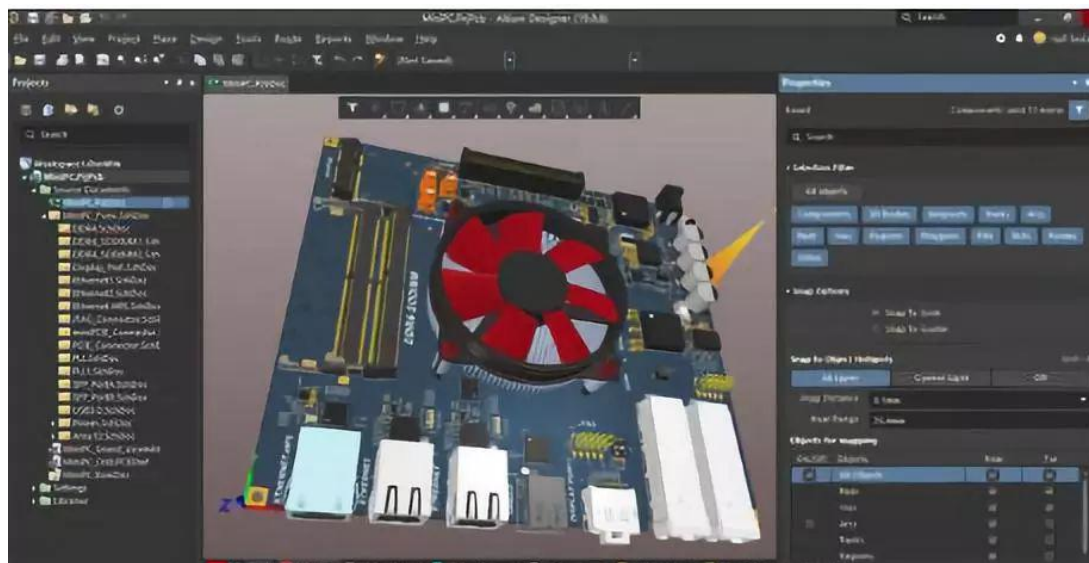
AD14 显示的信号线长度

另外，在高版本的软件里（AD17 或以上），还可以定义板边的间距规则。具体的来说，就是设置一个规则来限制走线、器件焊盘、过孔等物体到 PCB 边缘的间距。关于这部分内容，以前已经讲过了，可以参考这篇文章。Altium 电路板 outline（板框）规则。



板边规则

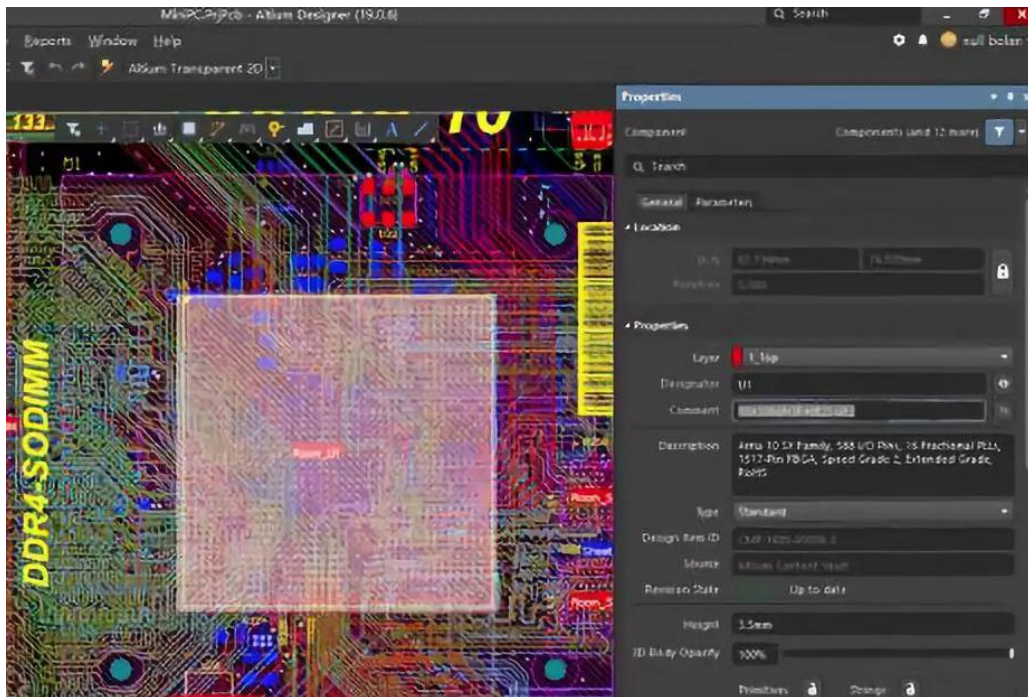
AD17 和之前的版本比较接近。AD18、AD19 则是发生了重大的改变。首先，AD18，AD19 是 64 位软件（之前都是 32 位软件），性能有所提升。主要表现在 3D、2D 切换、铺铜速度加快。



AD19

与之前版本相比变化较大的是，AD18 和 AD19 增加了 Properties 菜单。利用这个菜单可以很方便的进行很多操作，最有用的是过滤器功能。当需要单独选择一类物体时（过孔、器件、走线等），可以在过滤器里面只选中一种，这样就可以精确的选择需要的东西了。还有一些 PCB 的设置、单位、板子信息等，都可以在菜单里面找到。

这个菜单还可以用来修改各种物体的属性，比如我们选中 PCB 上面的一个 CPU，就可以在 Properties 菜单看到这个元器件的所有属性，而且还可以直接修改。

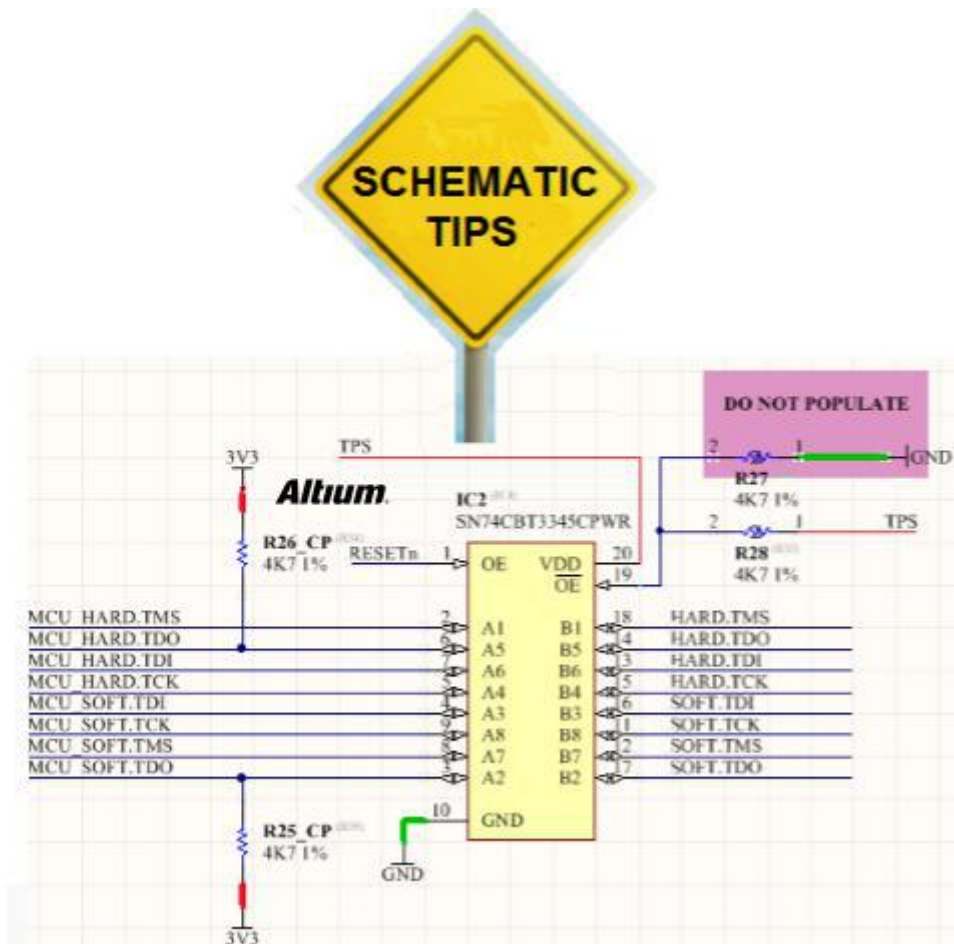


二、Altium Designer 软件功能介绍

1. 对象智能粘贴功能详解

俗话说，好的开始就是成功的一半。原理图设计输入作为整个电子产品设计的起始阶段，其设计效率和好的方法习惯，对后续各环节以致整个设计流程的影响，也是相当重要的。在整个项目的原理图设计阶段，需要设计

师胸有成竹，对项目或产品的功能需求以及相应的解决方案有高度而清晰的认识，才能对设计电路及功能模块随意搭配，任意组合。游刃有余，如鱼得水，在设计灵感的海洋，跳脱穿梭，如入无人之境！

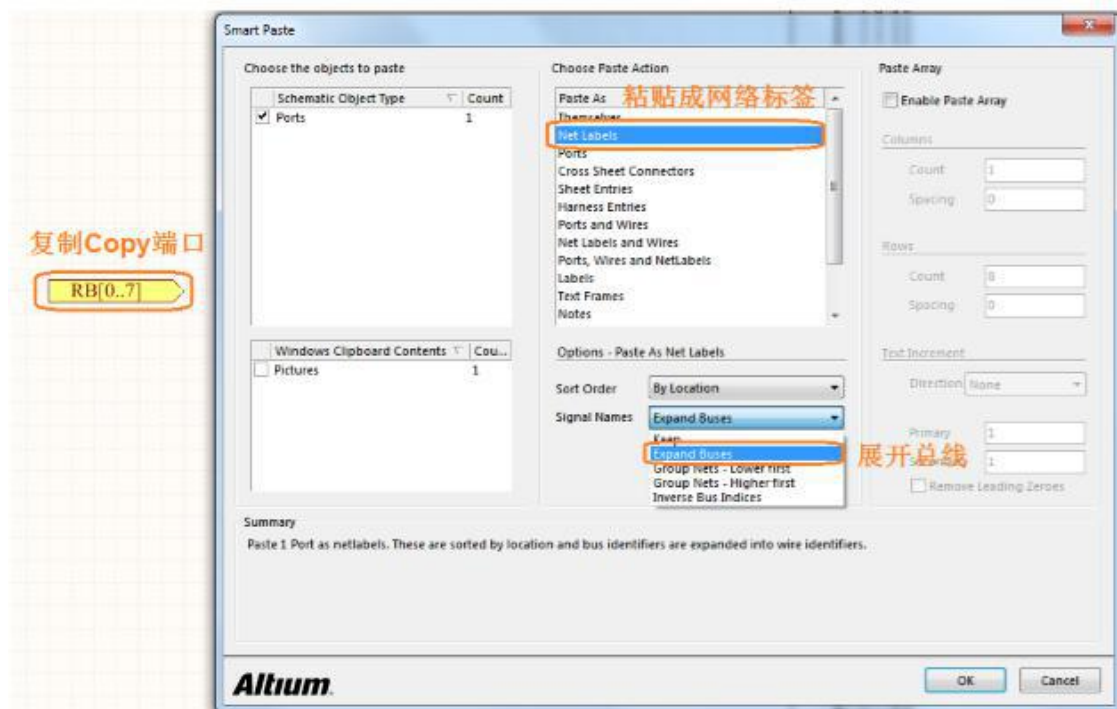


但是，在设计师正淋漓酣畅地将设计灵感赋予最终的电路模块的时候，需要方便顺手的设计工具快速高效地将手头作品呈现出心中想要的模样。而繁杂耗时的操作步骤很显然会阻碍设计师灵感的发挥。Altium Designer在提供方便易用的原理图输入工具之外，还有若干省时省事的小技巧。下面给大家随便介绍几个。

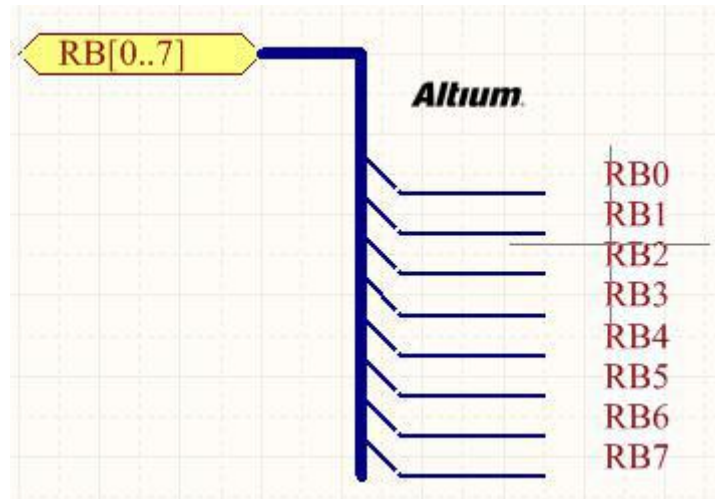
对象智能粘贴功能(Smart Paste)

将某个对象复制粘贴成另外的对象，还可以添加连线，组合信号顺序等等。比如我们来展示将总线端口复制粘贴成各分支总线网络标签。这只是个别案例，相信各位看官已经可以从粘贴对象列表里发掘更多有用的粘贴技巧。

1. 选择端口复制（快捷键 Ctrl+C）：如 RB[0..7]；
2. Edit > Smart Paste。将端口粘贴成网络标签，并且展开总线各分支；

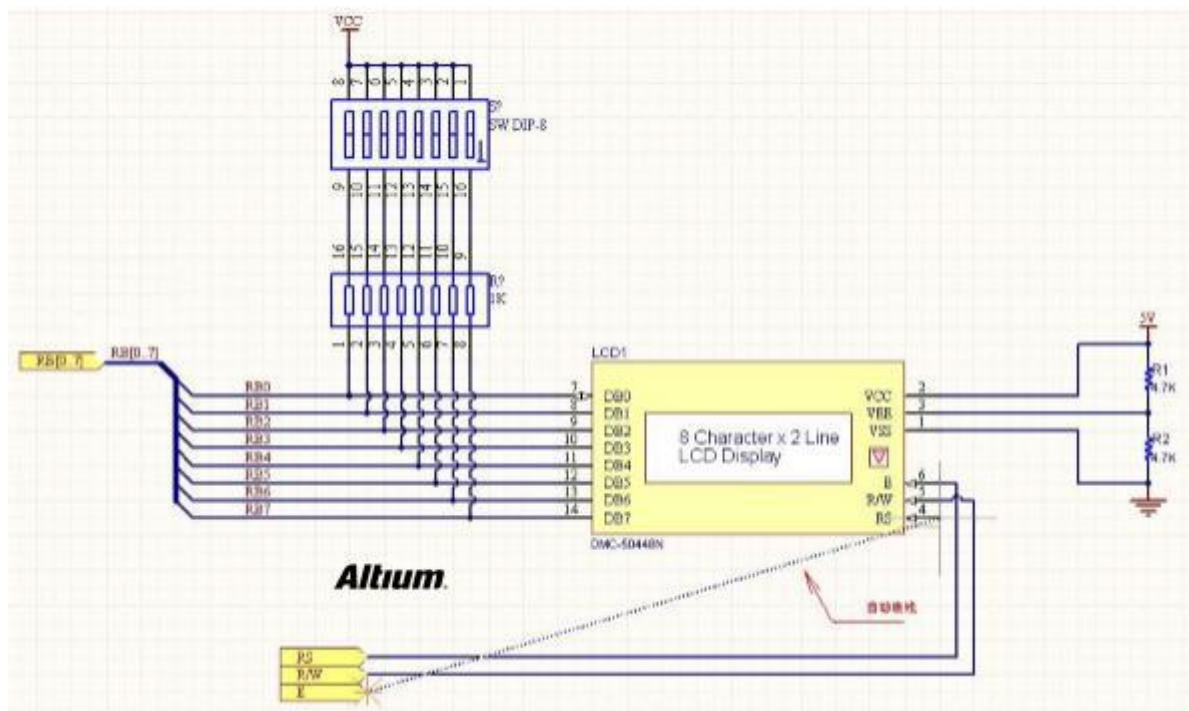


3. 放置自动生成的符号。

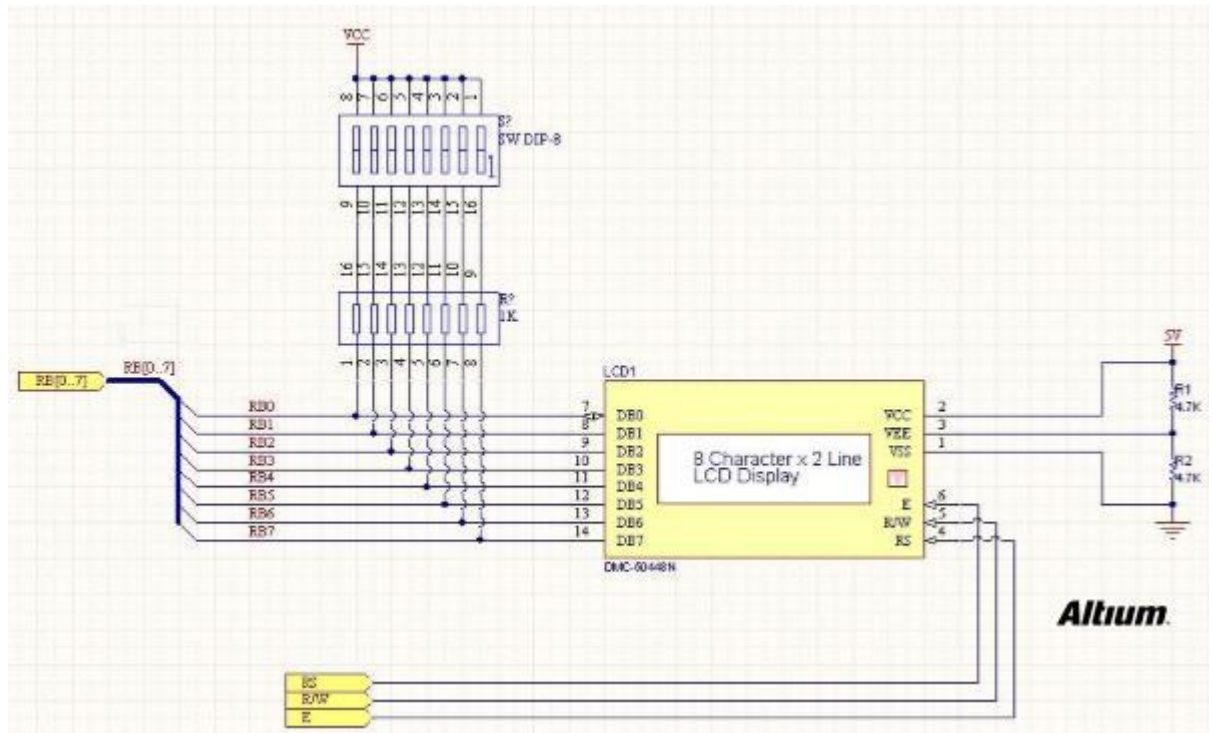


自动连线 (Place Wire (auto-connected))

1. Place > Wires ;
2. 快捷键 Shift + Space , 切换电气连线模式到虚线连接模式 ;

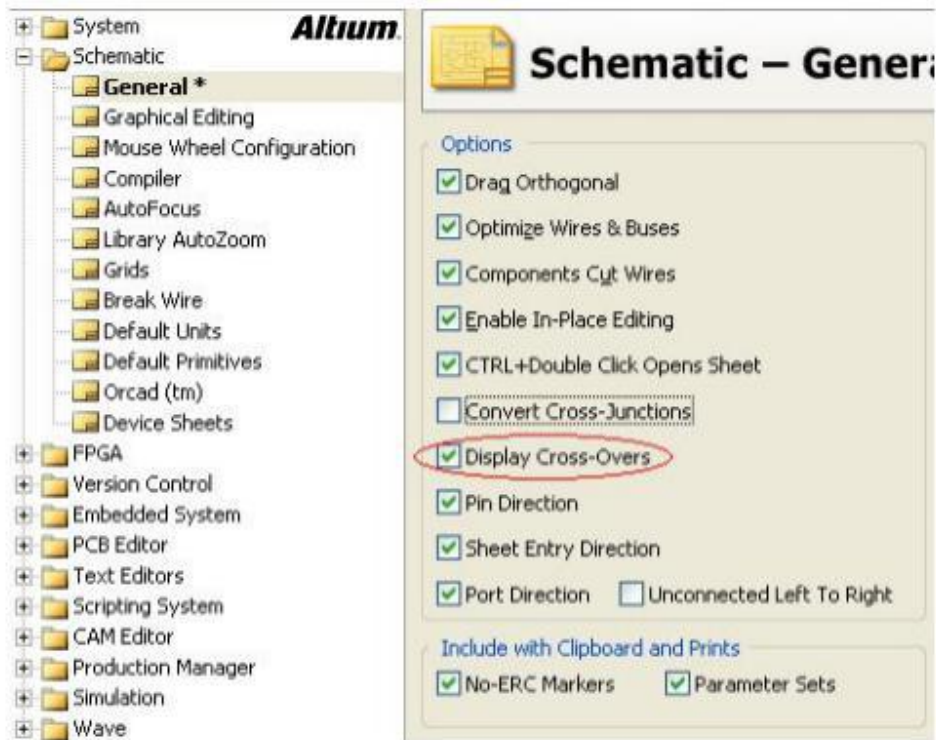


3. 分别用鼠标点击需要连接的网络两端。

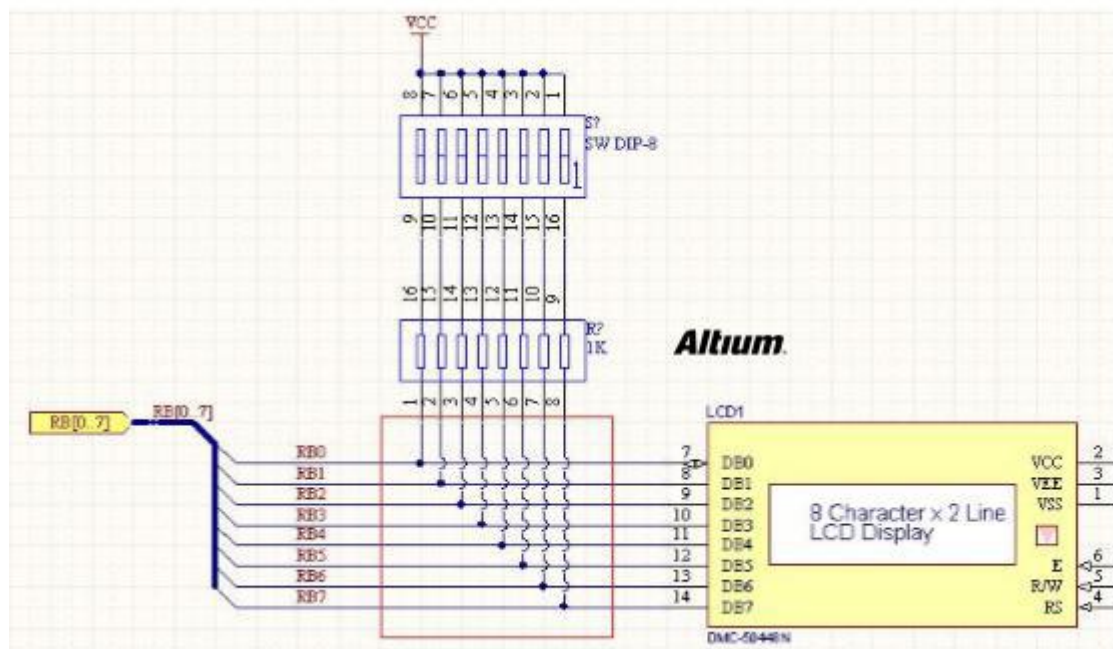


交叉跨线显示模式(Display Cross-Overs)

1. Tools > Schematic Preferences > Schematic > General ;
2. 选中 Display Cross-Overs 功能 ;



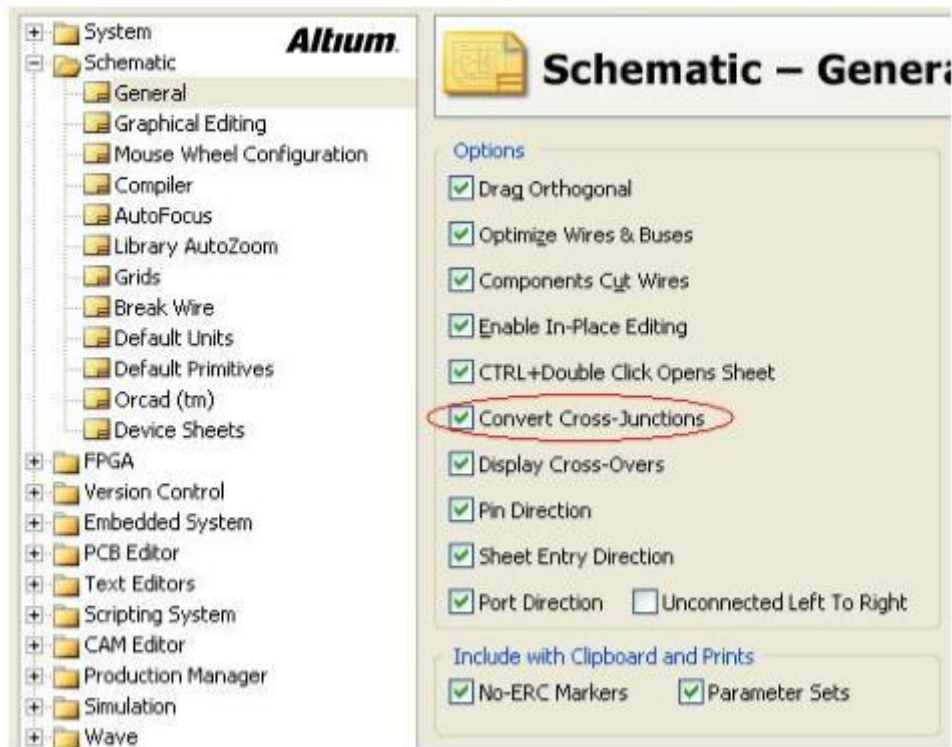
3. 原理图连线自动显示模式。



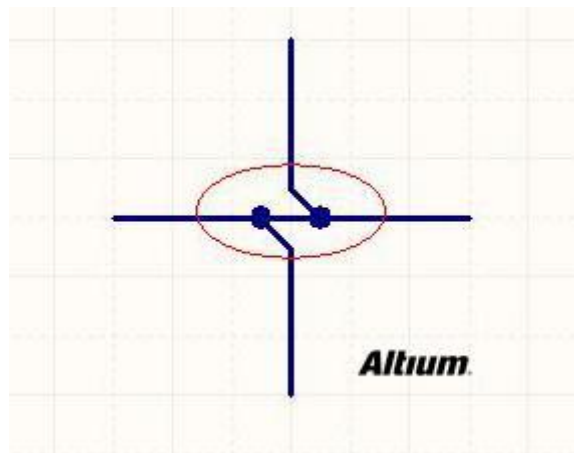
自动生成交叉节点 (Convert Cross-Junctions)

1. Tools > Schematic Preferences > Schematic > General ;

2. 选中 Convert Cross-Junctions 功能；



3. 原理图交叉连线节点将自动分开。



三、Altium Designer PCB 电路板设计步骤与技巧介绍

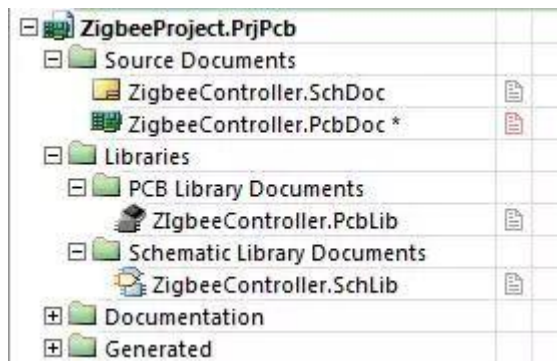
PCB 项目中一般包含四个文件：xxx.SchDoc(PCB 原理图)、xxx.PcbDoc(PCB 封装图)、xxx.SchLib(PCB 原理图库)、xxx.PcbLib(PCB 封装库)。

一般流程为：先绘制【PCB 原理图库 SchLib】、再绘制【PCB 封装库 PcbLib】，将原理图库组件与封装库关联起来。

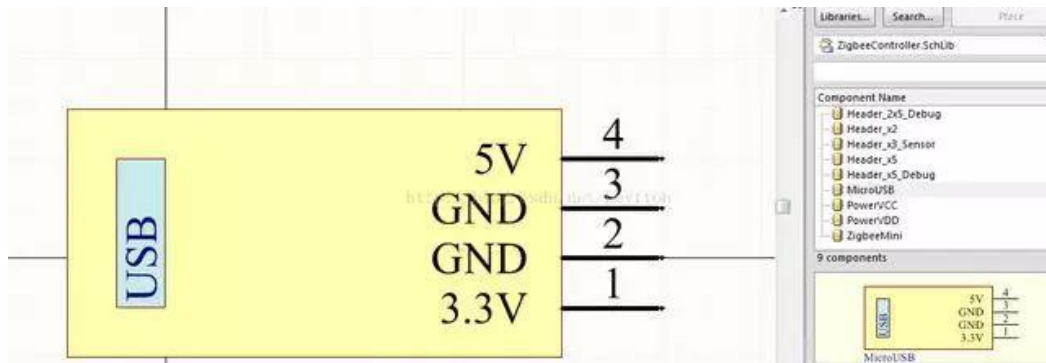
然后绘制【PCB 原理图 SchDoc】，将其更新到【PCB 封装图 PcbDoc】，Design->update PCB Document，最后布线扑铜即可。

效果预览

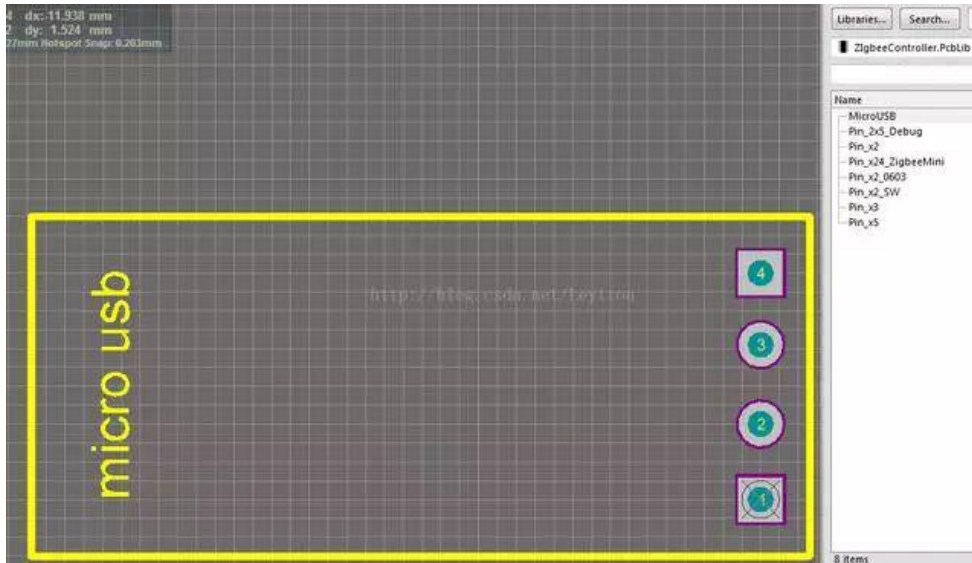
1. 项目结构图



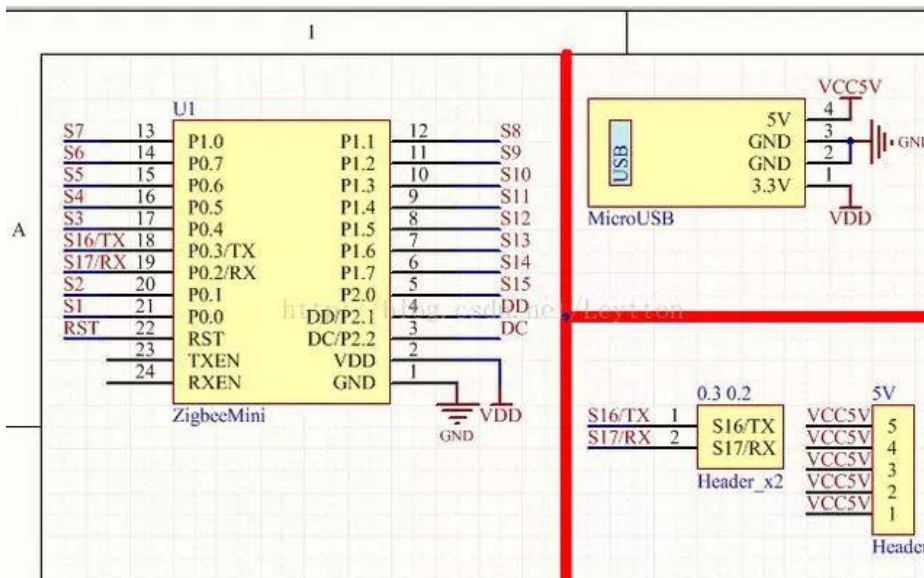
2. 原理图库



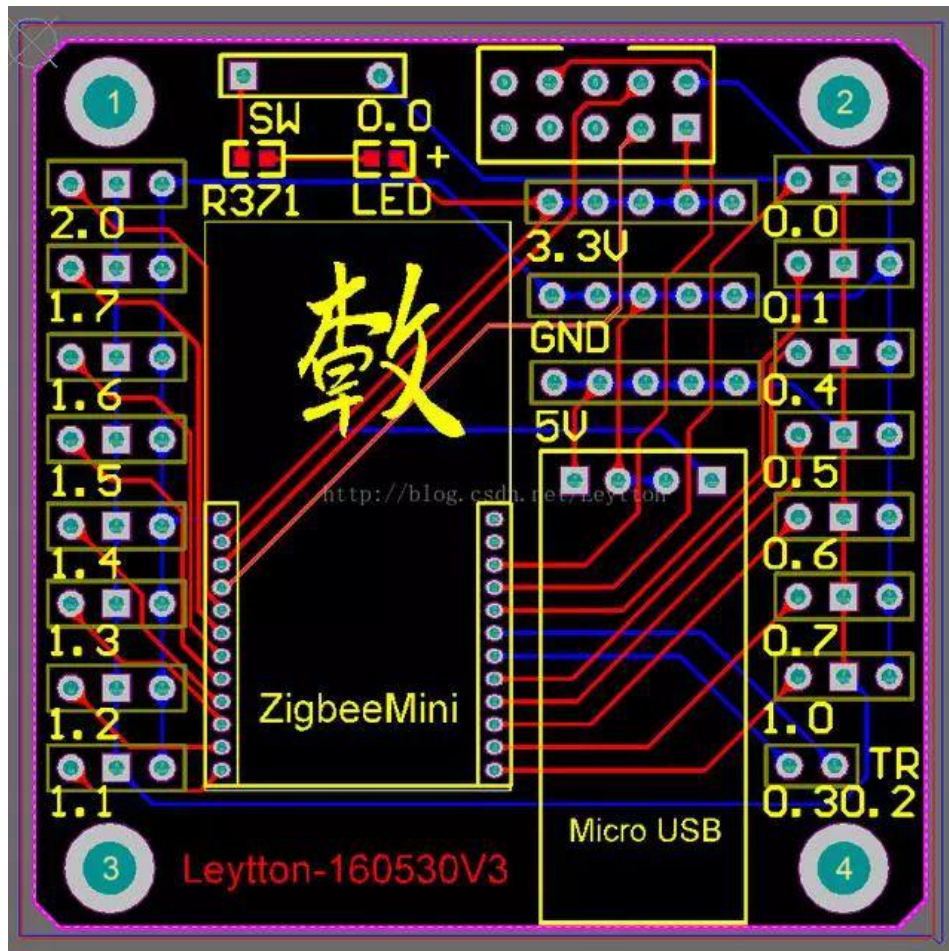
3. 封装库



4. 原理图



5. 封装图



一、集成库制作

制作集成库(原理图库、PCB 封装库、PCB3D 库)、添加元器件到原理图、导入 PCB、PCB 设计和布线、3D 预览

1、引脚等放置前按 TAB 设置属性、按空格旋转

2、Designator 为引脚标志，与 PCB 封装库引脚对应，Tools->模型管理把原理库与 PCB 封装库关联起来

3、显示上划线 \A\B

二、添加 3D 库

三、PCB 原理图绘制

Tools->Annotate Sch.. ->Update Changes List -> Accepted Changes 自动添加标注, 消除 altium designer duplicate part 错误

四、PCB 电路板绘制

自动布线:Auto Route->All..

检查规则:Tools->Design Rule Check..

小技巧

扑铜:

利于散热等

Place->Ploygen Pour

Top 层和 Buttom 层都铺, 注意铺铜的时候选择接地 GND(在布线的时候地线不用连接)

注:第二种 Hatched 扑铜方式,线宽要选择 10mil 不然无法通过规则检查

PCB 封装库修改后更新原理图库 Tools->update PCB..

原理图库修改后更新原理图 Tools->update Sch..

原理图修改后更新 PCB 封装设计 Design->update PCB Document

execute changes->validate changes

设置板子形状:

绘制边缘线条，shift 键选择

Design->Board Shape->Define Form selected objects

板子变成灰色->右键->Filter->Clare Filter

设置电气边界:

切换到 Keep-Out 层，Place->Line 厂家会根据这个形状生产。

电气边界用来限定布线和元件放置的范围，它是通过在禁止布线层 (Keep Out Layer) 绘制边界来实现的。禁止布线层是 PCB 工作空间中一个用来确定有效位置和布线区域的特殊工作层面，所有的信号层的目标对象和走线都被限制在电气边界之内。

Q 键实现英制和米制单位转换

正反两面放置元器件:

法 1、在拖动元器件的同时，按下 L，该元器件就自动到底层去了。

法 2、右击查看属性，修改 Layer 属性为 Bottom Layer 即可。

隐藏 GND 网络,扑铜时一般连接 GND，那么在绘制 PCB 连线时隐藏 GND 连线会比较清爽,快捷键 N-H-GND 点击 GND 焊盘就行了,注意一定要敷铜接地，不然整块板子全废了。

四、Altium-Designer-PCB 布局布线及规则设置

创建 PCB 文件

- 通过菜单或工程面板创建

-在工程面板中，右键单击工程

- Add New to Project - PCB

-新建的文件并不直接在硬盘中创建，需要保存

-这样创建的 PCB 文档，不包含图纸边框和标题栏

•通过文件面板（File Panel）中的向导创建

-在文件面板中的“New From Template”区域（可能需要收起上面的其它区域才看得到）中，单击“PCB Board Wizard...”

-在接下来的几个页面中，依次设置：

•单位，英制或公制

•图纸类型，自定义或预定义的纸张大小以及一些专用的板卡模板

•自定义图纸尺寸，如果在图纸类中选择了自定义，则需要自行设置图纸尺寸和一些相关信息

•信号层（顶、底和内层布线层）数量和内电层数量，一般为偶数

•过孔形式

•主要器件类型（贴片或穿孔）和器件是否分布在 PCB 的两面

•最小线宽、安全间距、最小过孔直径和孔径

•一般采用第一种方法创建 PCB 文件即可

编辑环境

•在 PCB 图纸中

-缩放、图纸拖动与原理图中一样操作

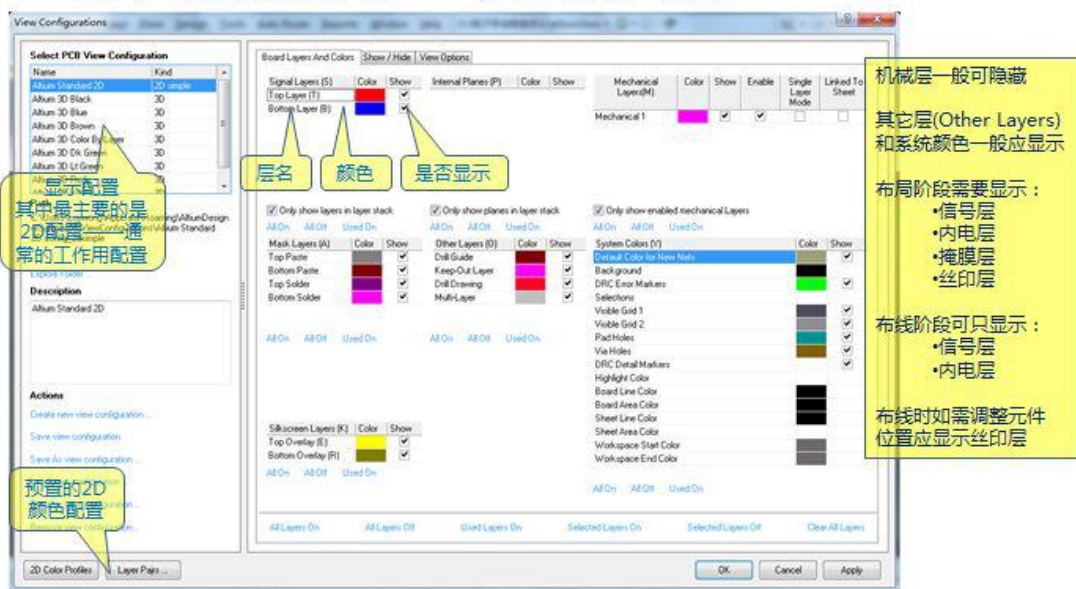
-选择操作与原理图中一样，但是因为 PCB 有多个工作层，如果在点选的位置上，有多个层的元素，AD 会弹出一个带有缩略图的列表框询问具体需要点选的元素

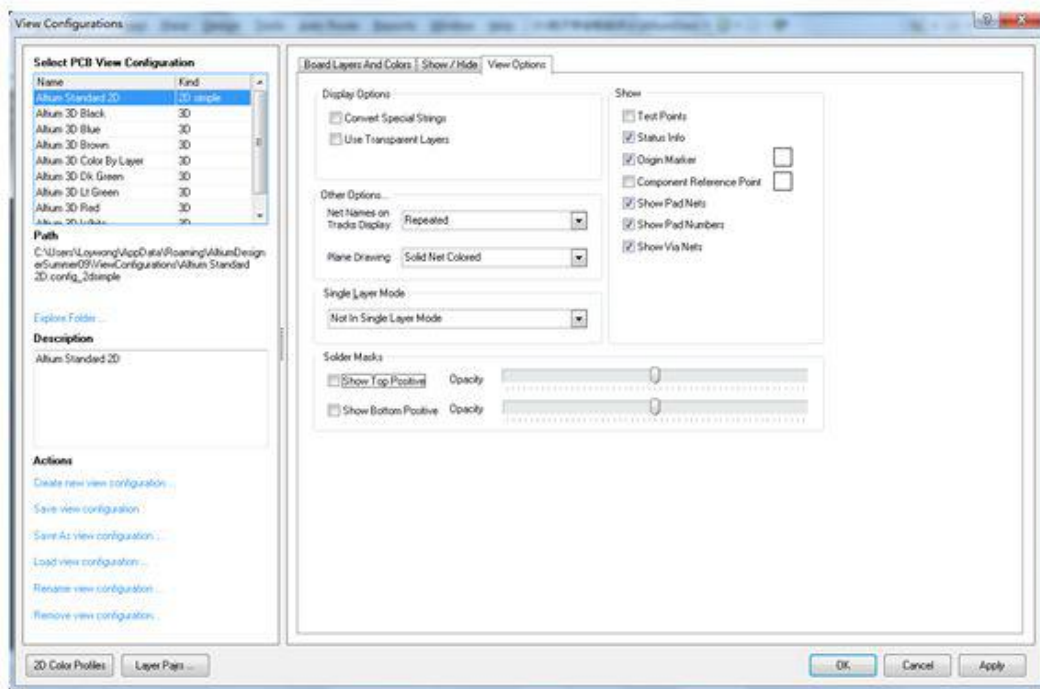
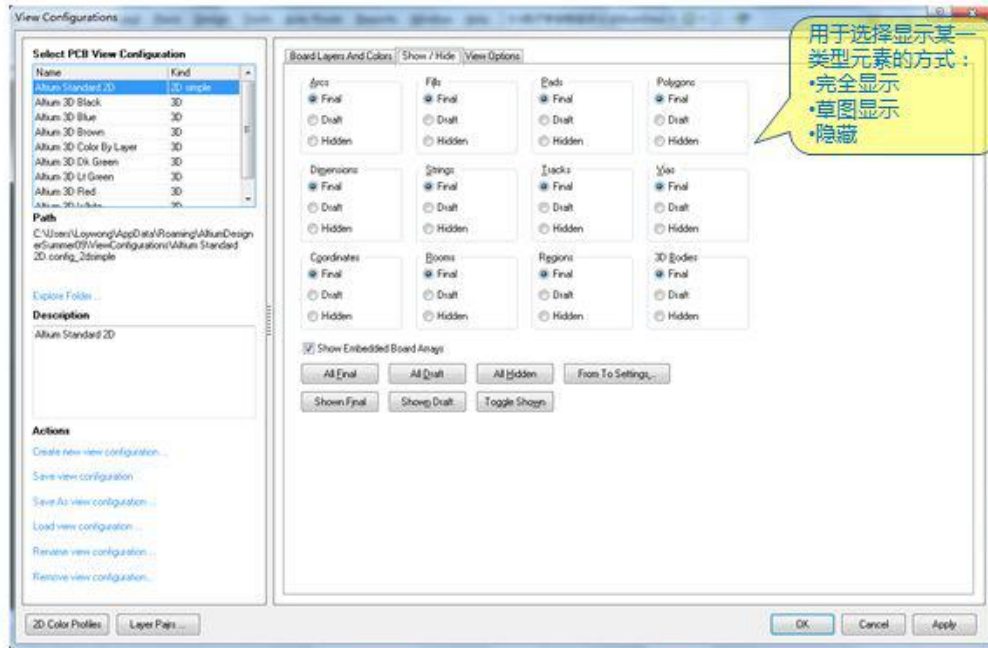
-选择相似对象（“Find Similar Objects”）和 PCB 检视器（“PCB Inspector”）的使用与在原理图中一样，也可以用于批量更改元件属性

-拖动元件时，“Space”和“Shift+Space”与原理图中一样，可用于元件的旋转，但是“X”和“Y”键不可再用于元件镜像翻转

选项

•图层显示选项（快捷键“L”），层和颜色标签

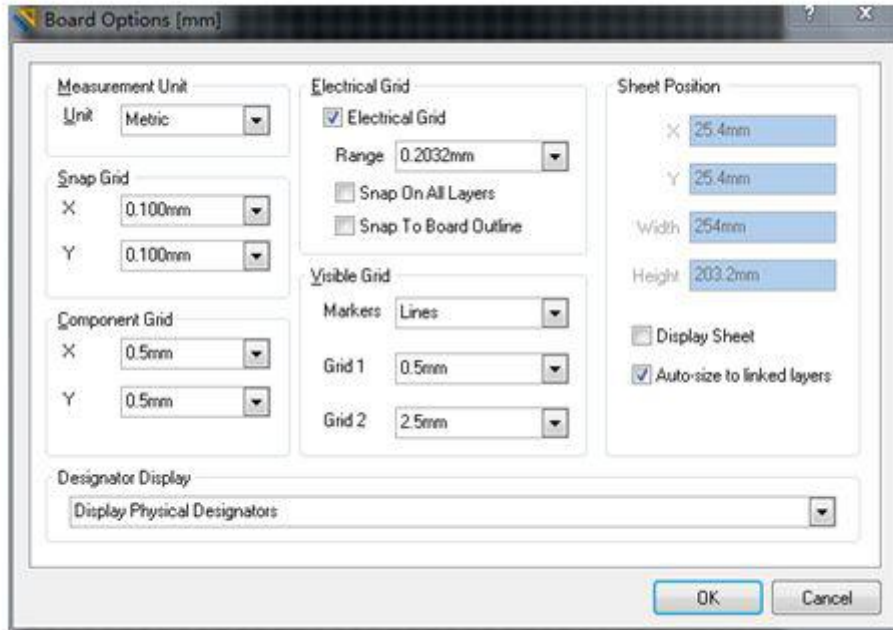




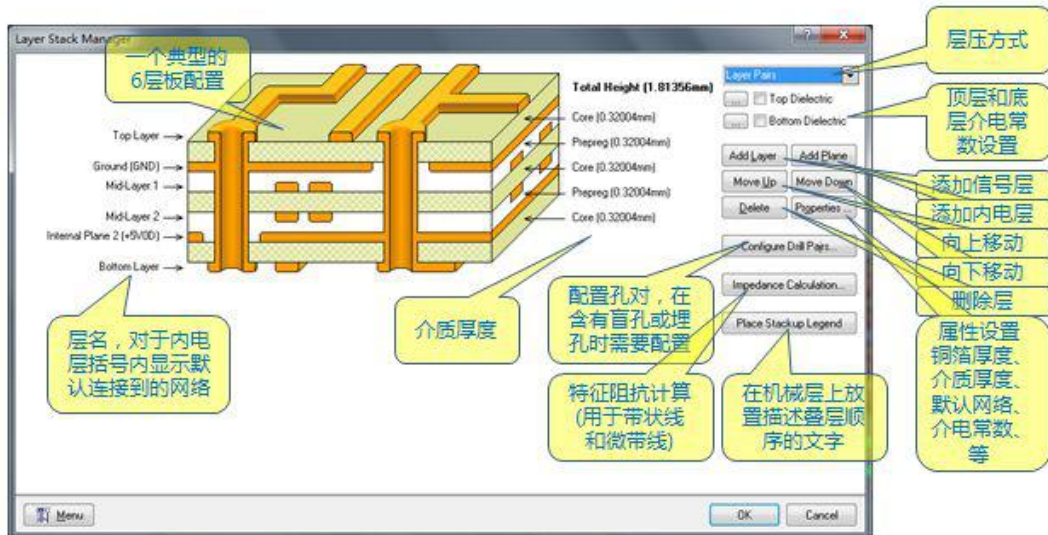
• 图纸选项

-与 PCB 库文件中的图纸选项类似，不再赘述

-在编辑 PCB 文件时，要善于使用“G”键，实时地改变栅格大小，便于定位



•板层（电气层）设定



在“DataAcq51”工程中，只需要Top和Bottom两个信号层即可

规则

•PCB 规则

-是 PCB 设计中至关重要的一个环节；保证 PCB 符合电气要求、机械加工（精度）要求；为自动布局、布线和部分手动布局、布线操作提供依据

-为规则检查提供依据，PCB 编辑期间，AD 会实时地进行一些规则检查，违规的地方会作标记（亮绿色），也可通过“T - D - R”进行全面的批量规则检查

•PCB 规则分类

•Electrical（电气规则）：安全间距、线网连接等 •Routing（布线）：线宽、过孔形状尺寸、布线拓扑、布线层、封装出线等

•SMT（表面贴装（贴片））：贴片元件焊盘的一些要求

•Mask（掩膜）：阻焊和焊膏的扩展

•Plane（内电层和铺铜）：内电层和铺铜与焊盘的连接方式 •Testpoint（测试点） •Manufacturing（加工）：孔、焊盘、丝印和阻焊的尺寸及相关关系

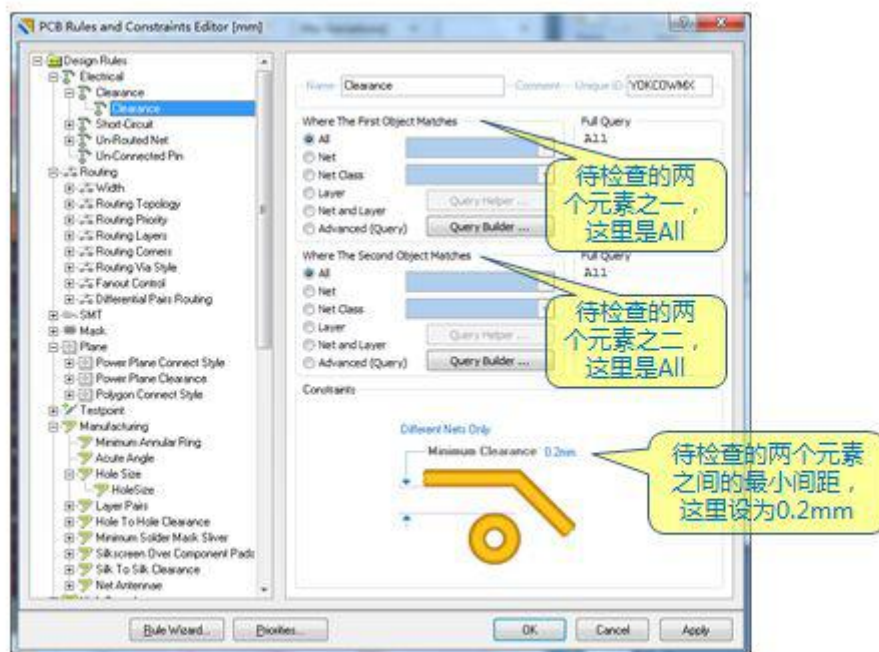
•HighSpeed（高速信号）：串扰、线长、配长、过孔数量等与高速信号相关的

•Placement（放置）：元件放置和元件间距等 •SignalIntegrity（信号完整性）：走线阻抗及高速信号的过冲、摆率等

-同一类规则下可以包含（新建）多个规则，并为每个规则设置不同的适用范围和优先级，以根据具体需求实现灵活多样的规则

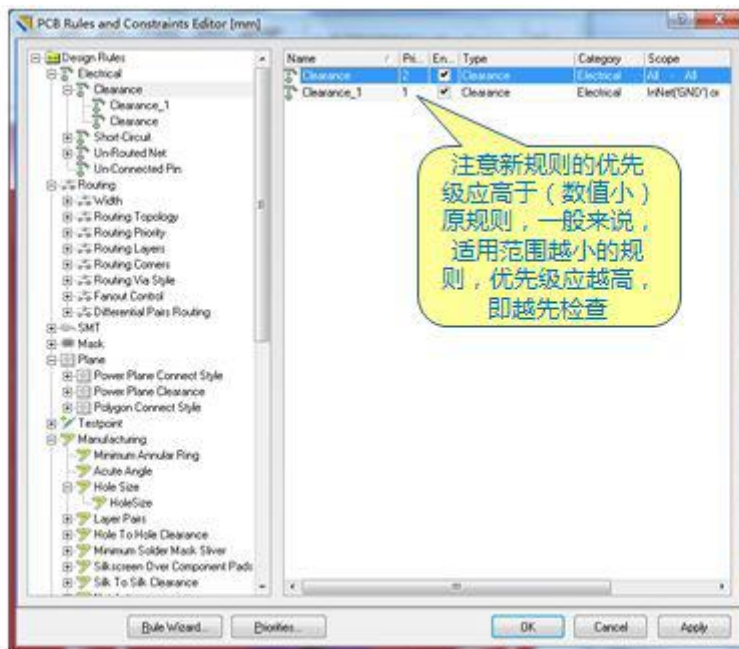
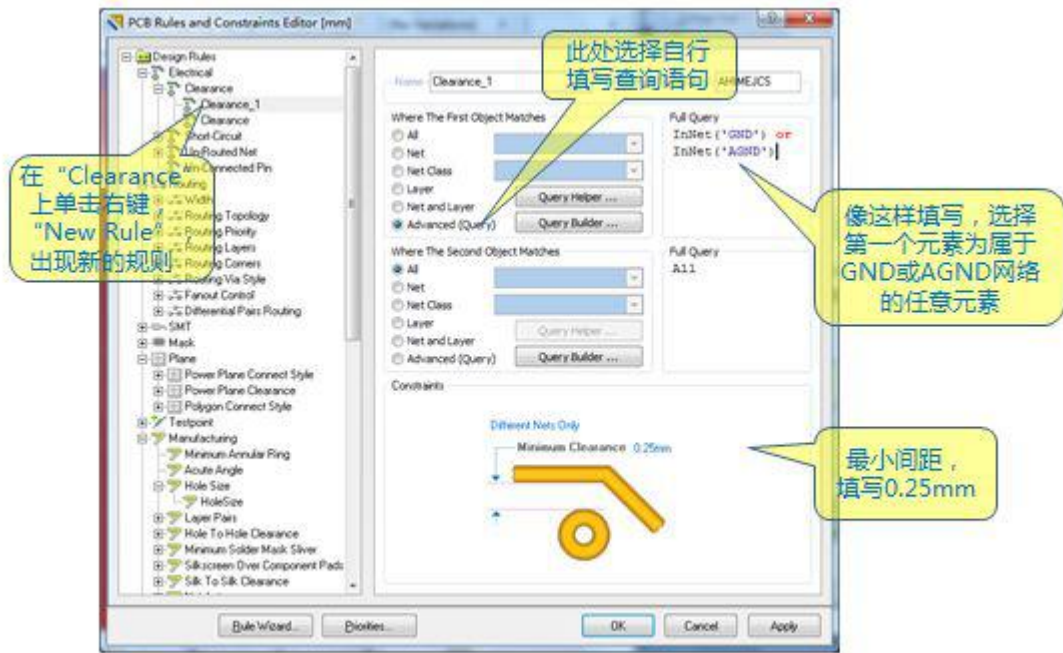
•常用规则设置（以“DataAcq51”工程为例）

-Electrical - Clearance



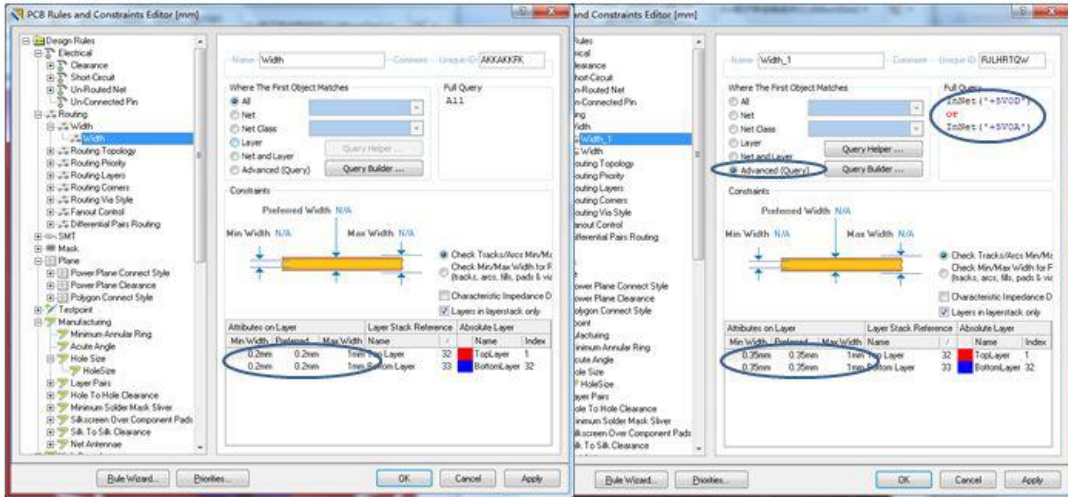
-Electrical - Clearance

•添加一条新规则，设置 GND 和 AGND 与任何其它网络的元素之间的最小间距为 0.25mm



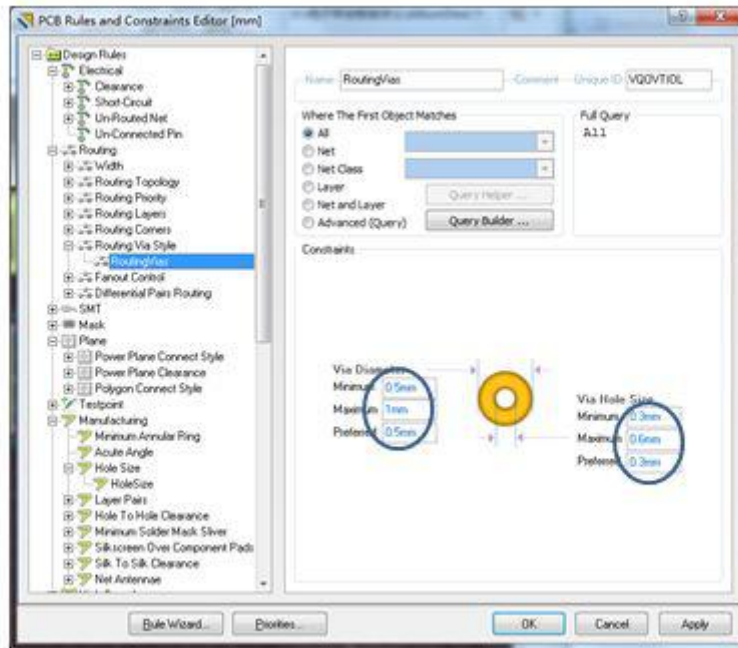
-Routing - Width

- 同样设置两个规则。添加新规则, 适用于 "+5V0D" 和 "+5V0A", 最小线宽定为 0.35mm

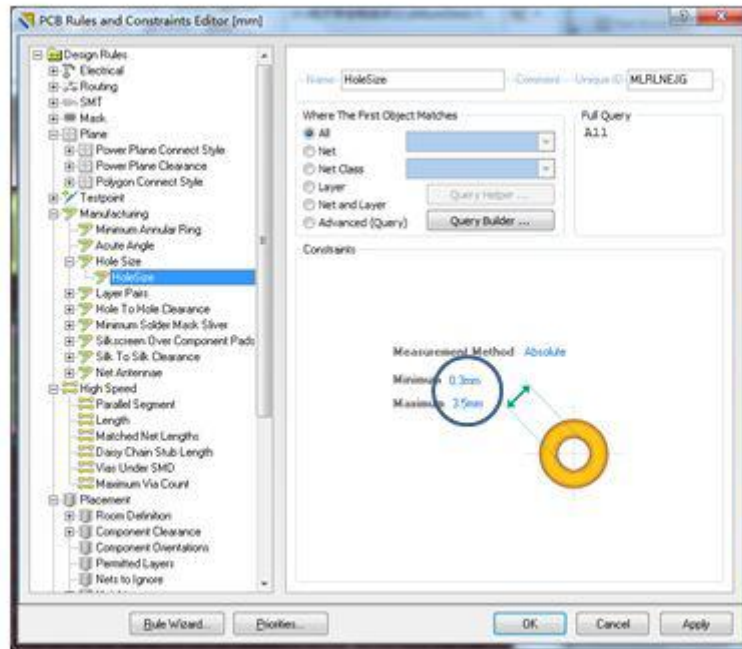


-Routing - Routing Via Style

- 设定过孔直径 0.5~1.0mm，孔径 0.3mm~0.6mm



-Manufacturing - HoleSize



- 常用规则设置（以“DataAcq51”工程为例）

- Minimum SolderMask Sliver，最小阻焊桥，可设为 0.01mm

- Silkscreen Over Component Pads，丝印与元件焊盘间距，可设为 0.01mm

- Silk To Silk Clearance，丝印间距，可设为 0.01mm

- 还有其它一些较常用的规则，将在后面的课中具体讲到
- 铺铜与焊盘、过孔的连接

- 内电层与焊盘、过孔的连接或间距

- 高速信号线的长度、配长
 - 微带线与带状线的阻抗

- 走线过孔的数量、焊盘下的过孔

绘制外形

- PCB 文件中的绘图工具与 PCB 库文件中的一样，不再赘述

- 定义 PCB 的外形

- 两种方法：

- 使用菜单快捷键 “D - S - R” (Redefine Board Shape) ，可绘制边框的外形 (绘制过程中可按 “Space” 更改出线方向，按 “Shift+Space” 更改线型) 绘制至最后一边时可单击右键结束，AD 会自动完成最后一边

- 在 Keepout 层使用绘图工具绘制任意形状的封闭线条，选中它们，然后使用菜单快捷键 “D - S - D” (Define From Selected Objects) ，AD 会根据选中的线条形状定义 PCB 外形

- 目前大多数 PCB 板加工厂仍然使用 Keepout 层线条定义 PCB 外形，因此在用上述第一种方式绘制外形后，还应在 Keepout 层绘制同样的线条，推荐采用第二种方式绘制外形

- 需要在 PCB 中切除部分，可使用菜单快捷键 “D - S - C”(Define Board Cutout)

- 绘制完成后，可重新定义原点位置，“E - O - S”，一般可定义在板子左下角、某个定位孔处或板子中央

- PCB 外形和原点位置可以在绘制 PCB 过程中随时更改

在“DataAcq51”工程中，将板子外形设置为 100mm×100mm 原点
定在板子左下角

与原理图同步

- 原理图与 PCB 同步工具

-可以将原理图中的更改导入到 PCB 中

- 在原理图编辑环境中，菜单“D - U”

- 在 PCB 编辑环境中，菜单“D - I”

- 对于新建的 PCB 文件，即是将整个原理图的内容导入 PCB

-或将 PCB 中的更改导入到原理图中

- 在 PCB 编辑环境中，菜单“D - U”

-同步过程中如有错误，应检查后再次同步，常见错误有：

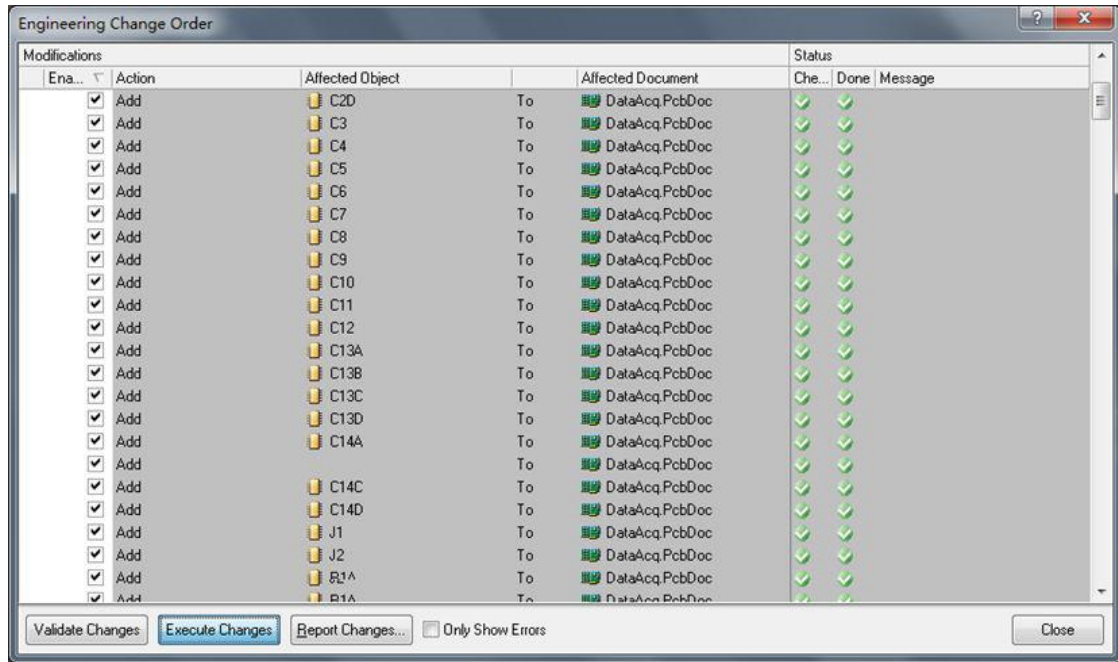
- 封装错误

- 封装库未引用

- 封装中焊盘与元件引脚不配对

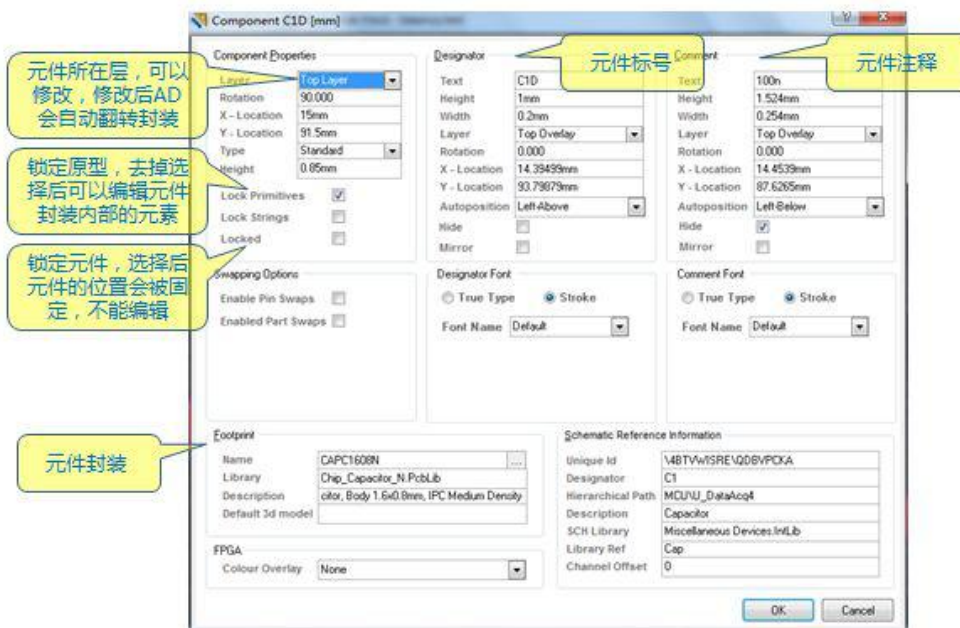
- 元件标号重复

- 等等



元件属性

- 双击元件可编辑元件属性



布局

- 布局

-同步完成后，元器件按照 Room 整齐摆放，在布线之前需要对元件布局

-布局和布线并不完全是两个分离的过程，往往在大概布局后就可以开始着手布线，在布线的过程中可以根据需要再调整部分元件的布局

-布局是 PCB 设计过程中的重要步骤，如果算上布线过程中对元件位置调整花去的时间，布局需要花去的时间和精力有时并不比布线过程花去的少

-良好的元件布局

- 可以使后续的布线工作更加简单，布线更简洁明了，可以说，布局在极大程度上决定了后面布线的成败和优劣

- 可以使整个 PCB 具有更好的信号完整性和高频稳定性

- 使得 PCB 面积更小

-布局时应结合原理图中功能单元的区分和信号的流向，结合原理图中的具体电路连接

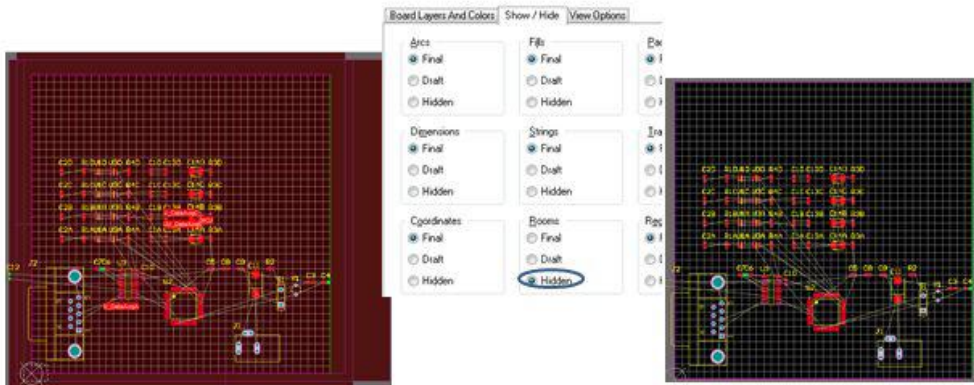
-自动布局的效果往往远远不及手工布局，不推荐使用

-规则中，一个 Room 中的元件是必须放置在 Room 中的，所以在手工布局前

- 可将所有 Room 拖动（会连带 room 中的元件）到 PCB 中

- 将 room 的大小拖放到比 PCB 稍稍大一些

- 可在图层显示选项中（“L”），将 room 设置为隐藏

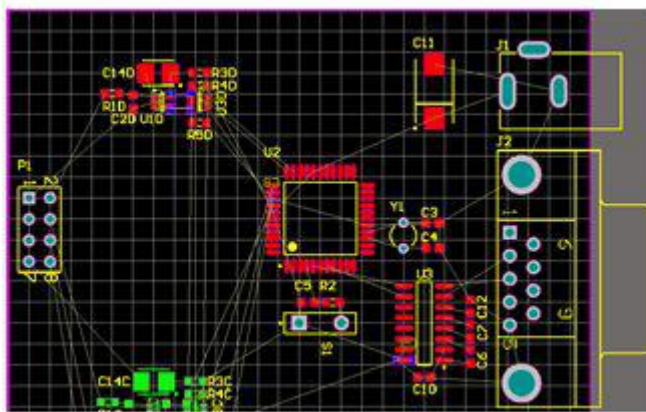


-如果觉得丝印层字符过大，可以使用“Find Similar Objects”（在对象上单击右键）工具和 PCB 检视器（PCB Inspector）批量修改

-根据电路结构，思考元件的大概布置，并粗略布局（以“DataAcq51”为例）

- 四个通道的信号调理和 ADC 电路可以上下并排放置在 PCB 左侧
- MCU 电源和串口可以放置在 PCB 右侧
- 100nF 的电源去耦电容需要与芯片电源引脚尽量靠近，可以放置在电源引脚正下方的 Bottom 层
- 信号主要通过顶层布线，电源主要通过底层布线，注意模拟部分和数字部分布置分开

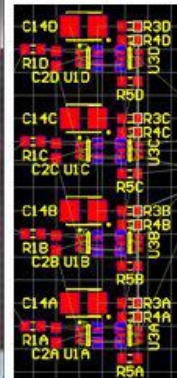
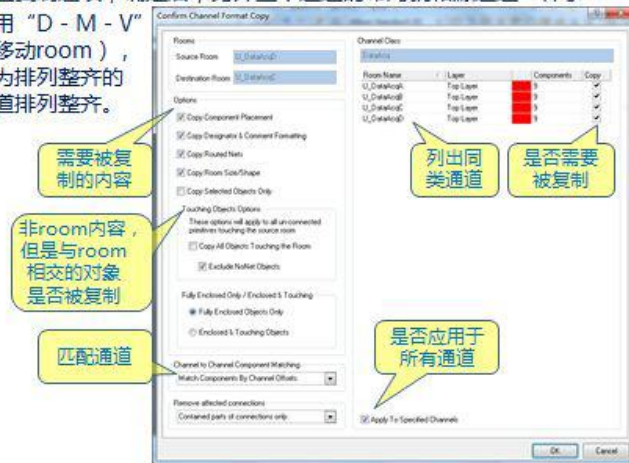
-多通道的设计中，可以先只对一个通道的元件进行布局，然后通过 Room 格式复制功能将布局复制到所有其它通道中



布局好的一个
信号采集通道
和单片机、
串口部分

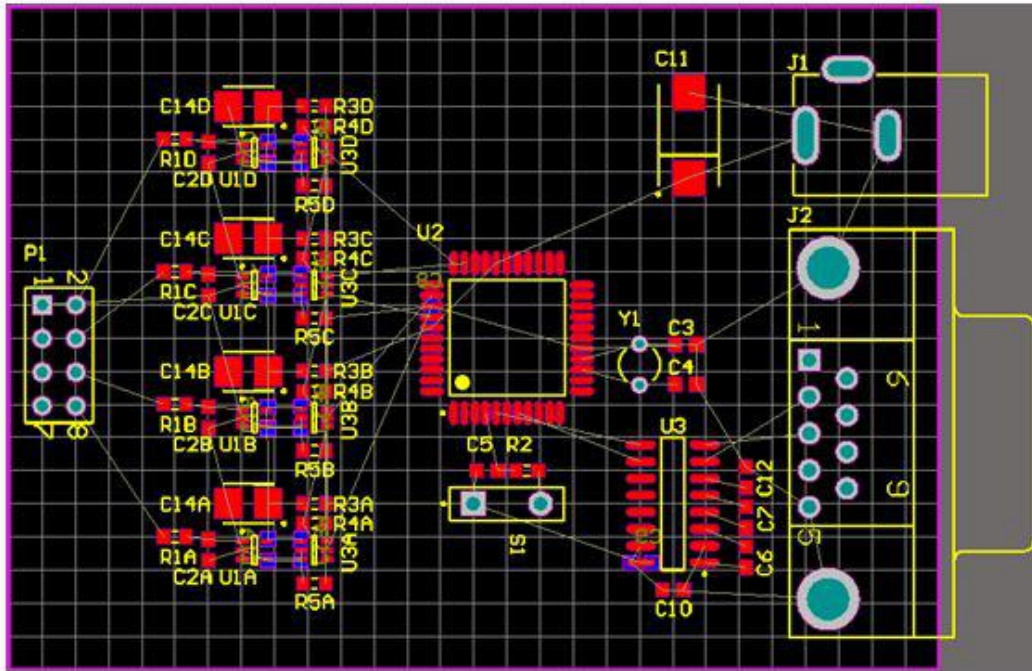
•Room 格式复制功能

- 在图层显示选项中，显示room
- 在准备复制的源room和目标room上单击右键 - Room Actions - Wrap Rectangle Room Around Component
- “D - M - C”，选择源room，然后选择目标room
- 设置复制选项，确定后，另外三个通道的布局就和源通道一样了
- 使用“D - M - V”（移动room），将为排列整齐的通道排列整齐。



复制格式
并排列整
齐后的四
个通道

•调整接插件位置，并重新定义 PCB 外形



布线

• 布线工具

- 交互式布线 ("P - T") , 普通的交互式布线

- 交互式差分对布线 ("P - I") , 布置差分信号的线对

- 交互式多重布线 ("P - M") , 布置多个相邻的线 , 选中欲引出线的多个元素后使用该工具

• 交互式布线的相关设置 - 系统设定中的 PCB

- Interactive Routing



•布线

-在大多数最后大面积铺地的 PCB 中，布线时，可以不用布置地线

-同样地，多通道中，只需要先布通一个通道，然后使用 room 格式复制功能将布线复制到其它通道中

-在从焊盘引出导线时，出线方向应先顺着焊盘延伸方向

-实时规则检查会在 PCB 上打一些标记，按“T - M”可隐藏这些标记

-如布线需要穿越板层，可以放置过孔“P - V”

-布线过程中

•按数字键区“+”、“-”键可以更改工作层，并自动添加过孔

•按“Tab”键可设置线宽、过孔尺寸，布线方式等属性

•按“Space”键可更改出线方向

•按“Ctrl + Space”可更改线型，在下列 5 种方式中切换

-45 度直线拐角

-45 度弧线拐角

-90 度直线拐角

-90 度弧线拐角

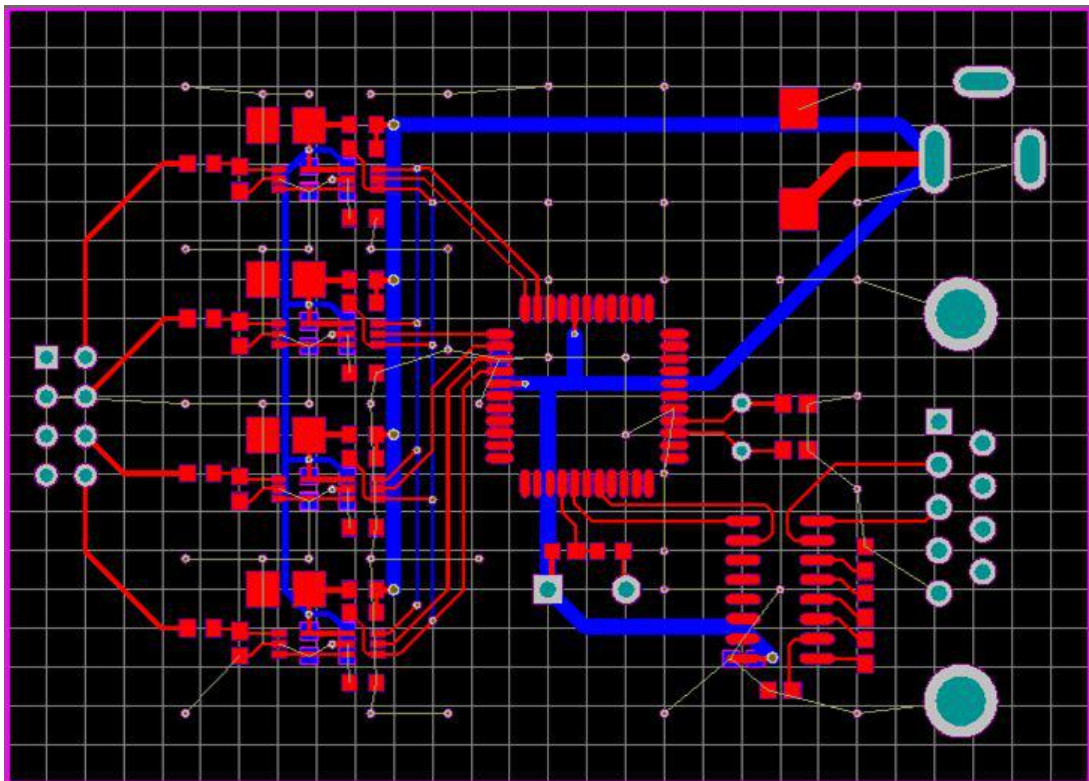
-任意角度直线

•按 “.” 键（主键盘区数字 1 键左侧），有更多功能的菜单

•布线完成的 PCB

-除地以外

-图中隐藏了丝印层



•元件封装修改

-有时候需要对元件封装内的一些元素进行修改，可去掉元件属性中的“Lock Primitives”后编辑封装内的元素

-例如“DateAcq.PcbDoc”中右上角的电源插座的焊盘，采用了较古老的多焊盘重叠的方法定义槽形孔，可将其修改

- 双击该封装，去掉属性中“Lock Primitives”处的勾选
- 删除封装中名为“J1-1A”、“J1-1B”、“J1-2A”、“J1-2B”、“J1-3A”、“J1-3B”的附属焊盘（隐藏在“J1-1”、“J1-2”、“J1-3”中）
- 依次双击余下的三个焊盘，更改其属性，变为槽形孔



五、方案设计中如何运用 Altium Designer 拼板？

随着整个电子产业的不断发展，电子行业的很多产品都已经有完善的上下游配套企业。从一个成熟产品的方案设计，外观设计，加工制造，装配测试，包装，批发商渠道等等，这样的一条产业链在特定的环境就这样自然地生成。因此，设计和制造之间的联系是极其紧密的，到了不可分割的地步。

电子产品从设计完成到加工制造其中最重要的一个环节就是 PCB 电路板的加工。而 PCB 加工出来的裸板绝大部分情况是要过贴片机贴片装配的。

那么问题来了，现在的电子产品都在向小型轻量化方向发展。当你的设计 PCB 板特别特别小，有的电子产品模块小到几厘米见方那么的小块时，PCB 加工制造倒还好说，但是到了 PCB 装配环节，那么小的面积放在贴片机上进行装配就带来了问题。没有办法上装配生产线！

这里就需要对小块 PCB 进行拼板，拼成符合装配上机要求的合适的面积，或者拼成阴阳板，更加便于贴片装配。一般情况下，制造板厂会提供拼板的服务，但在 Altium Designer 软件设计中进行拼板，除了更能清晰地展示设计师意图之外，还有诸多好处。好处如下：

可以按照自己想要的方向拼板

拼板文件与源板关联，源板的改动会自动更新到拼板

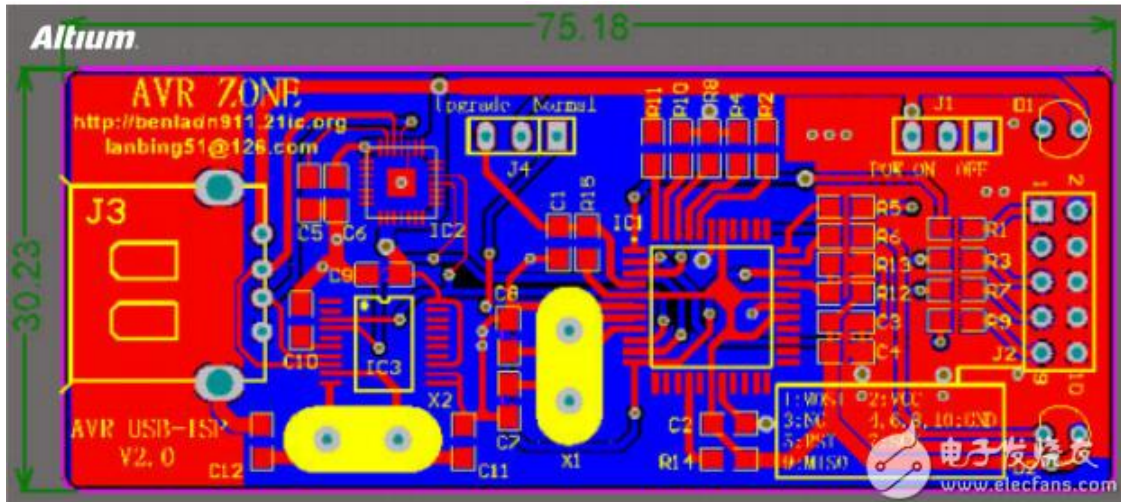
可以将几块不同的板拼在一起

可以拼阴阳板（正反面交替）

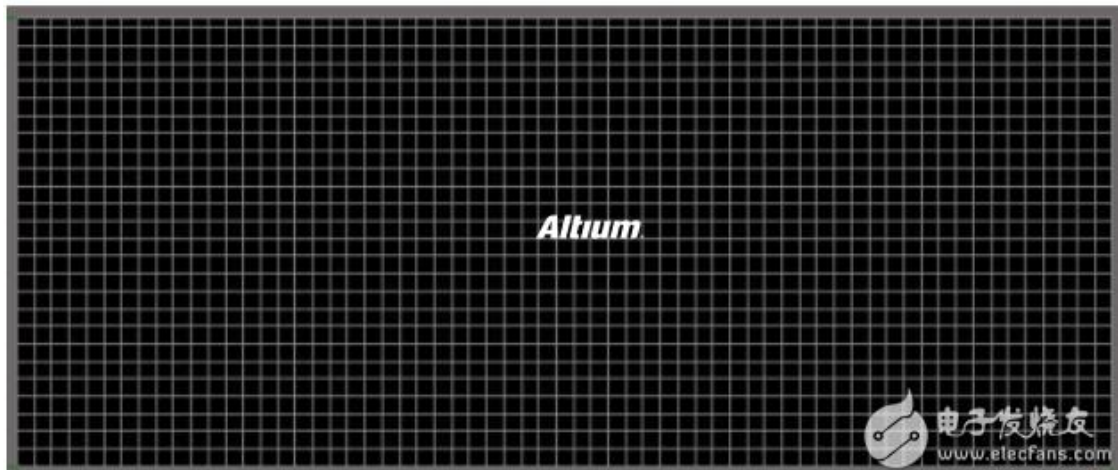
在 Altium Designer 如何拼板？

这里我们用一个例子来介绍在 Altium Designer 中拼板的过程和操作步骤。

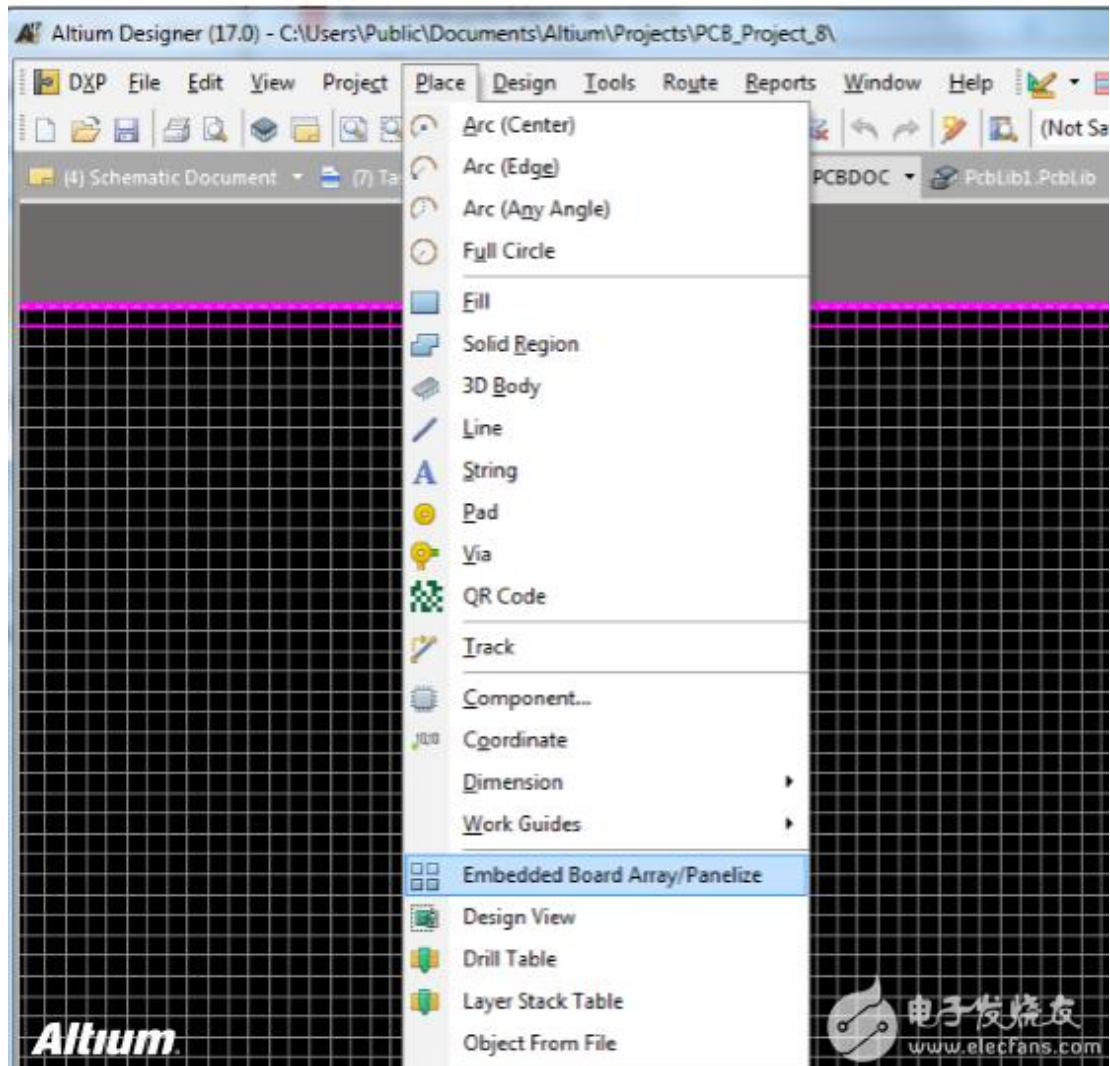
1, 首先确定板子的尺寸大小。这个可以用 Report > Board information 来看。如下图, 这块板作为范例的 PCB 板, 尺寸是 75.18 x 30.23mm。我们要在一张新的 PCB 图里拼出 2x2 的 PCB 阵列。



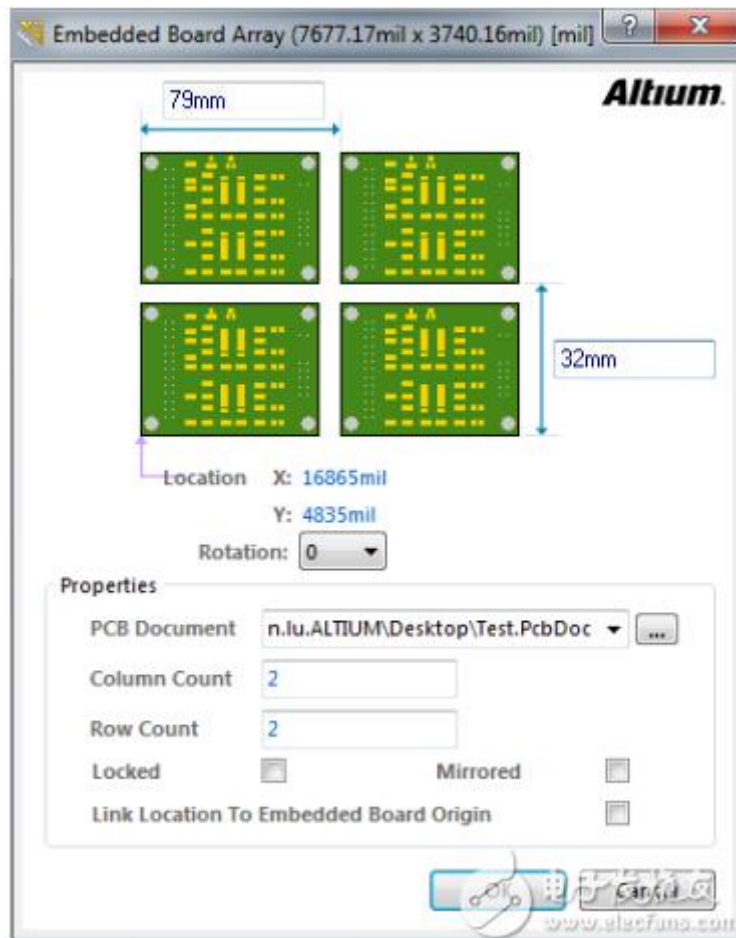
2, 使用 File > New > PCB 创建一个尺寸为 160 x 65mm 的矩形 PCB, 2 层信号层, 无电源层, 通孔。新创建的用于拼板的 PCB 如下, 保存。



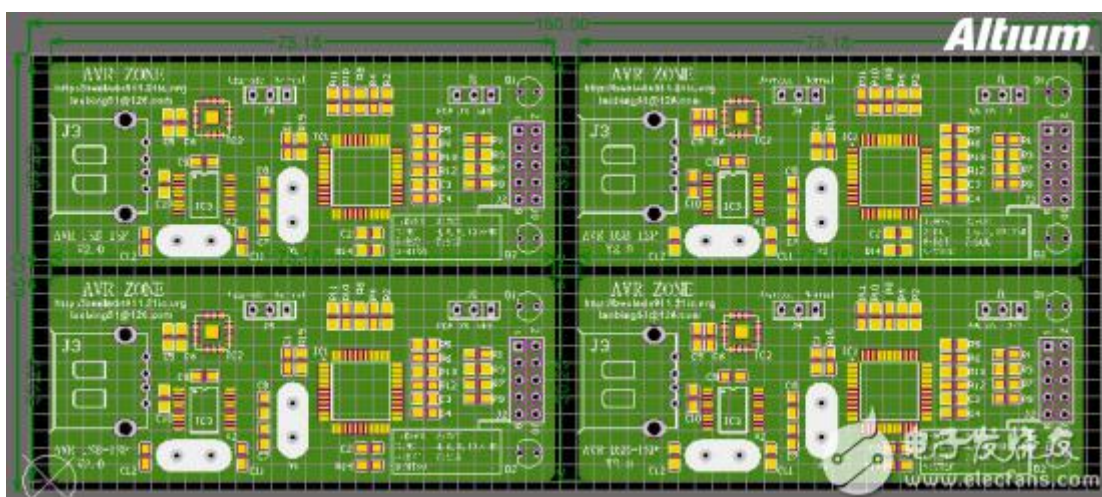
3, 在新建的 PCB 上, 点击 Place > Embedded Board Array/Panelize。如下图所示。



这个就是我们的拼版功能 ,点进去之后看到下图 ,在 Embedded Board Array 窗口中输出长和宽(这个参数一般比原板略大一点即可。视自己需求而定)。在 PCB Document 一栏选择你要拼的板 PCB 文件。然后在 Column Count 和 Row Count 的行列元素输入框中输入要拼板的横排和竖排的数量。这里各自选 2。



4, 设置好以上参数后, 点击 OK, 放置阵列板到 PCB 图上。调整好位置刚好居中即可。并且重新定位 PCB 的原点到阵列板的原点上。

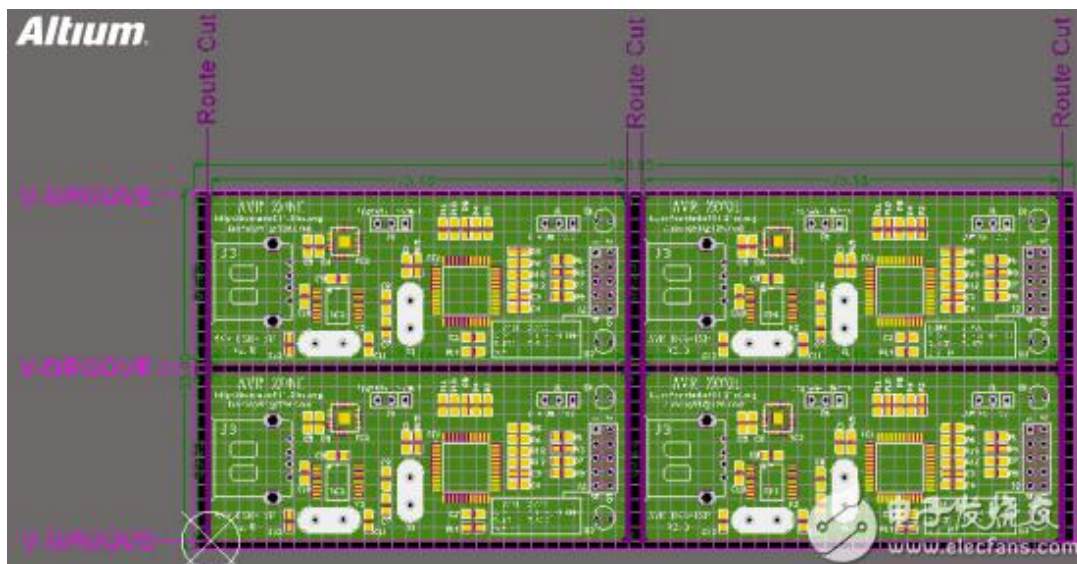


5, 下面进入层面颜色管理器, 把 Mechanical2 改名为 Route Cutter Tool Layer, 在这个层面上绘制的线定义为铣刀铣穿 PCB 的走线; 把 Mechanical5

改名为 FabNotes，在这个层上绘制的线定义为要在 PCB 上洗出 V 槽 (V-GROOVE) 的走线。

Mechanical Layers(M)	Color	ihov	inabl	Single Layer Mode	Linked To Sheet
Mechanical 1	Yellow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Route Cutter T	Red	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mechanical 3	Green	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mechanical 4	Blue	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FabNotes	Red	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

如下图为画好细节走线的阵列板。



注意，在 PCB 阵列板上画出我们需要的 Route Cutter Tool Layer 走线和 FabNotes 走线，这些走线时我们要让加工板厂 CAM 图纸处理的人员明白我们具体的需求和意图。但具体要走 Cut 还是 V 槽，以我们与板厂工程师的沟通和交流为准。此处只是示意图。

最后，就是将 PCB 拼好的阵列板转换成 Gerber 等加工图纸文件。给到 PCB 加工板厂，与板厂沟通具体工艺要求和细节。

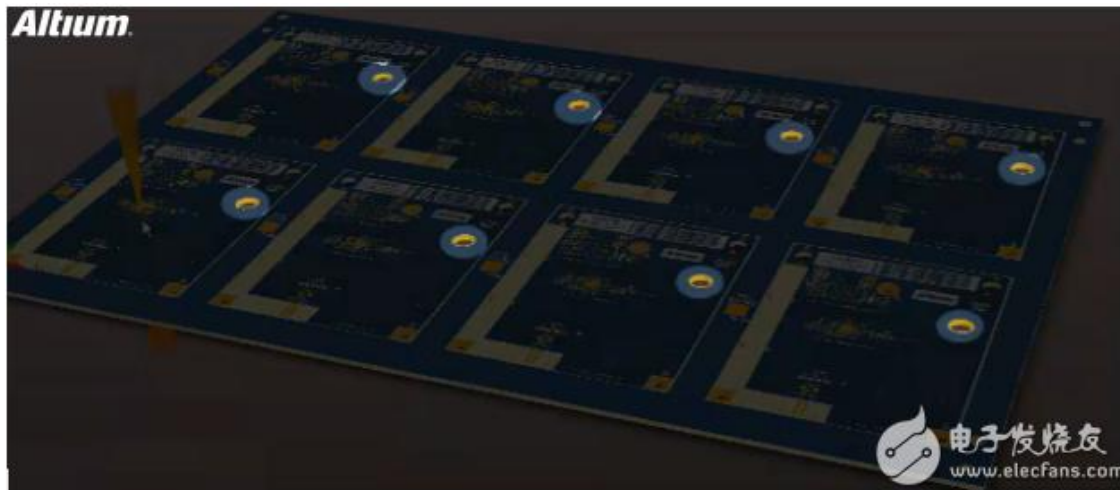
拼板与源板同步更新

如果在源 PCB 板上做任何改动,这些改动在 PCB 拼板文件中一键更新。

比如下图,在源 PCB 中放置一个焊盘。



然后在 PCB 拼板文件,点击 Refresh 刷新,这些每个板子上都会多出这样一个焊盘。随源板同步更新。如下图。



如何将不同的 PCB 拼在一起？

将不同的 PCB 拼在一起,只需要选择某块 PCB 文件,拼出阵列。然后再选择其他的 PCB 文件,再拼出阵列。如下图所示。



这里要注意的是，对于不同的 PCB 拼在一起的情况，那些需要拼在一起的 PCB 的层数设置和板层厚度设置都需要完全一致。才能拼在一起进行加工。

如要拼阴阳板的话，方法是先用此拼板功能放置一个板阵列，然后再放置另外一个板阵列的时候，将 mirror 选中即可。同样，阴阳板一定要每一层厚度都一样。才能拼在一起进行加工。

六、AD811 的应用实例

AD811 是美国模拟器件公司推出的一种宽带电流反馈视频运算放大器。它可应用于视频开关、视频线路驱动器、分配放大器、直流恢复电路，以及 ADC

和 DAC 的输入/输出缓冲器等方面的视频信号处理过程。亦能满足在宽波段、低失真和在高速时处理大信号的需要。其典型的技术特性如表 1 所示。

按使用温度可有 -40°C 到 +85°C 的 AD811A 类和 -55°C 到 +125°C 的 AD811S 类两种。从其封装方式来说，有 8 脚塑封 (N-8) 和陶瓷双列直插式封装 (Q-8)、20 脚 I.C.C 封装 (E-20A)、16 脚 SOIC (R-16) 封装、以及 20 脚 SOIC (R-20) 封装四类。这里仅列举 8 脚塑封 (N-8) 和陶瓷双列直插式封装 (Q-8) 的引脚示意图如图 1 所示。

在本文中后面列举的应用实例都是以这种封装形式为准。通常，把 AD811 用来作为非倒相联接如图 2 所示，而用来作为倒相联接的则如图 3 所示。其中 R_F 是用来调节带宽的反馈电阻， R_C 称为增益电阻，通过对 R_C 的变更来取得其调整增益。表 2 是一 3dB 带宽与闭环增益和电阻值的关系。

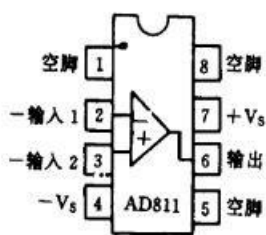


图 1 AD811 (N-8) 和 (Q-8) 引脚图

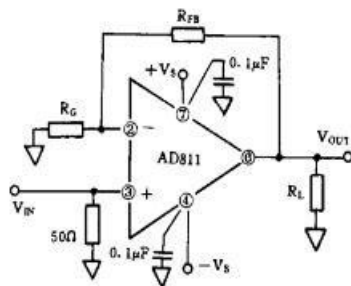


图 2 非倒相放大器的联接

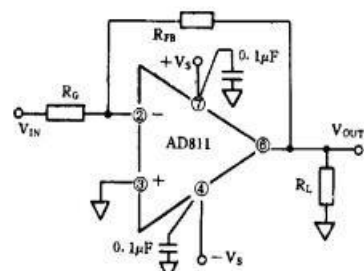


图 3 倒相放大器的联接

AD811 的应用实例

1、视屏线路驱动器

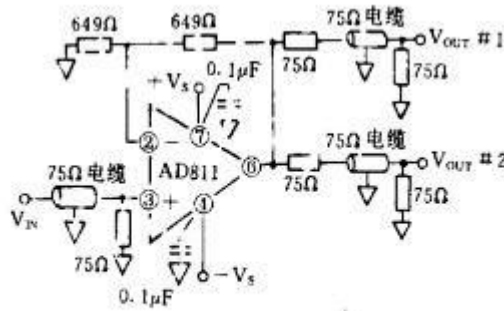


图 15 工作在+2 增益时的视频线路驱动器

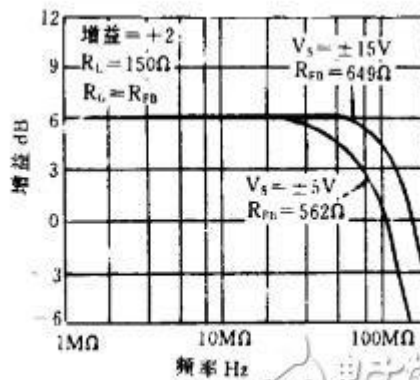


图 16 增益+2 时，频率与闭环增益的关系

当驱动多级反向端接的视频负载时，AD811 被设计成提供优质性能的闭环增益为 1 或大于 1 的线路。图 15 示出了工作在+2 的增益时的一个视频线路驱动器。当使用±15 伏电源时，将获得最小的微分增益和微分相位误差。而在±12V 电源时，则将略微增大这些误差并略微改善其增益的平坦性。

鉴于从功耗的角度来考虑，为获得最佳的视频性能，则推荐使用±12V 电源为宜。而在许多情况下，较低的电源也能获得很好的性能。在图 16 中示出了不同的电源电压时，频率与闭环增益的关系。图 17 和 18，各为电源电压与微分增益和微分相位的关系曲线图。

在驱动多根电缆时另一个重要的考虑乃是在电缆输出之间要有高频隔离。这是由于它是低输出阻抗，在用 5MHz 来驱动后接 75Ω 电缆时，在 AD811 的两个输出端之间的高频隔离要达到大于 40dB。

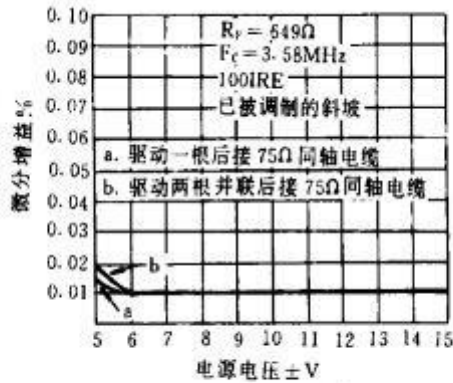


图 17 对图 15 所示视频线路驱动器的电源电压与差分增益误差的关系

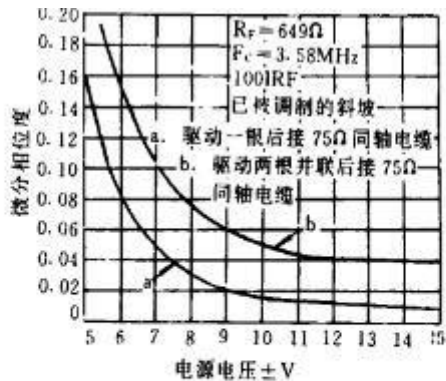
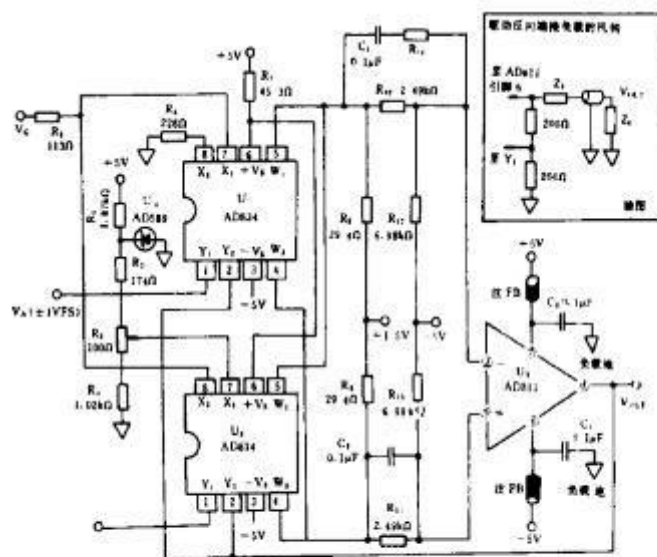


图 18 对图 15 所示视频线路驱动器的电源电压与差分相位误差的关系

2. 视频键控器电路

可由两个 AD834 乘法器，一个 AD811 和 1V 直流电源，一种特殊方式的两个输入 VCA 电路装配起来的电路称之为视频键控器。“键控”是一个术语，它使用了关于由一个第三信号控制下的进行对两个或更多的视频源混合，或使信

号建立在这种消散和交迭特殊效应。在图 19 中所示的这个电路是一组具有视频输入 V_a 和 V_i 和一个控制输入 V_c 的两输入键控器。在负载端有 V_{our} 的传递功能由 $V_{our} = GV_a + (1-G) V_i$ 给定。其中： G 是一个无量纲的变量（确切地说，恰好是“A”信号通路的增益），其范围是：当 $V_c=0$ 时，为 0，而到当 $V_c=+1V$ 时，为 1。由此，如 G 从 0 到 1 变化，则 V_{our} 连续地在 V_n 和 V_B 之间变化。



注：FB 铁氧体垫圈（与 C_1, C_2 构成滤波，
用作抗电磁干扰）

图 19 一个实际的视频键控器

此电路工作原理是简单的。首先来看，经 U_i 信号路经处理视频输入 V_a 。当 $V_c=0$ 时，其增益显然是零，以及当 $V_c=+1V$ 时，我们已选定的定标要确保它是不变的；这是要注意的传递功能的第一项 GV_n 。另一方面，当 X_i 是被偏置在一个精确的 $+1V$ 时， V_c 输入到 U_i 被认为是输入 X_i 的倒数。由此，当 $V_c=0$ 时，在其不变的满刻度值处，响应到视频输入 V_g 已经就绪，虽然，当 $V_c=+1V$ 时，差动输入 $X_i - X_i'$ 为零。这就形成第二项 $(1-G) V_g$ 。在乘法器的输出处，

所需要的偏流是由 R_g 和 R ，来提供的。包括 $R:O/RZ$ 和 RU/RS 的一个直流电平移位网络是用来保证 AD811 的输入结点都被安置在其共模范围的一个电压之内。

在高频时， C_i 和 C_Z 都各自旁路 R_O 和 R_q 。 R 是包括较低高频环节增益，以及在 AD834 中因为电压到电流转换所必需的，通过 YZ 输入，其结果是发生 2502 反馈电阻的一个有效值；这仅是在 AD811 的响应中，为取得最佳平坦性而需要的约为一半的值（要注意，此电阻是不受 G 影响的；当 $G=1$ 时，所有的反馈是通过 U_j ，而当 $G=0$ 时，则反馈是全部通过 U_Z 的）。 R_4 减少从乘法器进入到 AD811 的倒向输入总电流而引起的部分输出电流总量，它是用 R_g 共同分担的。

这个电阻能用来调整带宽和作为阻尼因子，以最佳满足应用装置。对于“1-G”项而言，用一个 AD589 基准电流 $1.225V+25mV$ 给含有 R ，到 R_4 电阻的分压器，以产生所需要的 1V 直流。电位器 R 将是可调的，使在 X ：输入处提供恰好 1V。在此情形下，对于 AD834 和 AD811 两者来说，已示出了应用士 15V 双电源的方案。亦即是当它从一个反向端接 752 线路驱动时，在这种情况下总增益在负载处的安排是一致的。这意味着“双 VCA”必须在一个最大为 2 增益时运行，而当在图 19 所示的 VCA 线路中，则宁可在 4 增益下工作。

可是，由减低反馈电阻，这是不能取得的。因为，在一个临界值略小于 5002 以下，AD811 的峰值脉冲是难以忍受的。这是因为在一个电流反馈放大器的开环交流响应中主要的标杆是受此反馈电阻控制的。它将有可能在 X_4 增益处工作，

然后在输出处衰减该信号。换言之，对 AD811 要在输入处选用 6dB 的衰减信号，这是 R_g 到 R 的功能。

3.同步脉冲分离装置电路

通用的视频信号处理过程的另一个要求是同步脉冲分离的功能。在一个同步脉冲分离装置中，要从 NTSC 全电视视频信号中去掉水平和垂直的定时信息，且要转换到下一步处理过程的逻辑电平。图 20 所示的就是一个同步脉冲分离装置的线路，它使用了一只 AD811 和两只分立式晶体管。在 J 处，它由一个 NTSC 全电视视频信号所驱动，且在输出处发送一个与 TTL 相兼容的正向同步脉冲。在此线路中， U_j 级要完成三个功能。第一， $L-C$ ：器件作为一个低通滤波器，是滤除 3.58MHz 色度成份。

U_i （一只 AD811）是兼有带增益的隔离器和缓冲器，用以驱动跟随级的，它是一个低输入阻抗同步脉冲顶部箝位电路。在 J 处的输入信号是用 R_i 和 R ： $-R$ ，组成分压，且被交流耦合到 U_i 输入的。 U_i 是由 R ， $-R$ 比率来构成一个带宽 $\times 6$ 的放大器的。从 J_i 到 C_i ，获得一个 3 倍的总亮度信号增益。用此所示的增益值和对 U_i 来说的 $\pm 5V$ （或更大些）直流电源，这级能处理正常的视频信号，而在输出处没有限幅。 U_i 的输出驱动一个在特别结构中的 PNP 分路箝位用的 Q ）。在稳态的直流期间， Q ！

由于 R ，的偏流而处在饱和状态，其发射极是接地的。因为交流信号是通过 C_i 驱动 Q 的，是一个具有同步脉冲顶部为负极限的全电视视频信号， Q_i 动态地作为一个直流电路恢复器。负向的视频波形同步脉冲顶部驱动 Q 进入到硬饱

和状态，以及当波形校正的部分将它带出饱和状态，它就起到了一个线性射极跟随器的作用。这个动作在 Q₁ 发射极处产生一个直流恢复的全电视视频信号，具有与地相关的同步脉冲顶部。Q₁ 的输出是通过肖特基二极管 D₁ 耦合到 NPN 开关管 Q₂ 的基极上。

此二极管的正向压降和 Q₂ 的 V_{be} 组合在 Q₁ 的基极处产生一开关夹断，就信号电平来说，在约为同步脉冲的顶部 50% 的幅度处，产生 Q₁ 的彻底的开/关。由 Q₂ 输出包括净化、无噪声的同步定时信息，在同部脉冲顶部期间趋于正向。由于 +5V 电源加到所示的 Q₂ 上，此信号是与 TTL 相兼容的。从此电路包括注意到 U_i 级的好的去耦取得了最实际的提示。在同步脉冲顶部期间，大的瞬时电流能发生电源与地的与/或噪声。对于逻辑电源的接地来说，带有大容量的 U_i 局部旁路电容将会有助于控制这个噪声，并将作为一种紧凑设计和接地平面的用法。

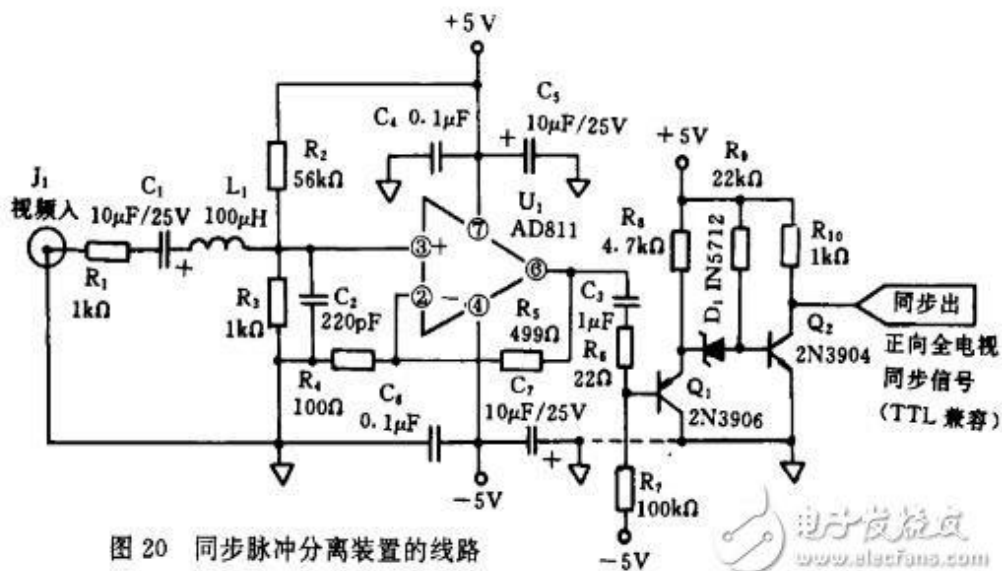


图 20 同步脉冲分离装置的线路

4.pic 单片机的 ad 转换

PIC 单片机如何表示电压

PIC 用十位二进制位的数来表示电压 ,也就是数值 0~1023 来表示电压。那比如现在这个数值是 400 那这代表多少的电压？这就要根据参考电压来确定了。

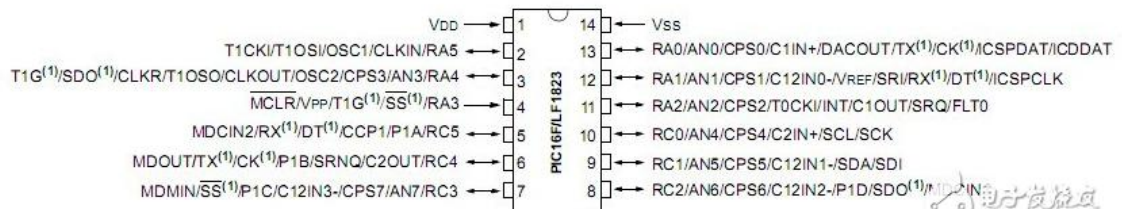
比如我们设置正参考电压为 3.3V ,当输入的电压为 0 时 ,数值就为 0。当输入的电压为 3.3V 时 ,数值就是 1023. 那如果输入的电压是 1.2V 代表多少电压。

首先 ,先算出一个数值代表多少的电压 3.3V 除以 1023 约等于 0.003V 。

然后 , 1.2V 除以 0.003V 等于 400. 这就得出了 400 代表的是 1.2V。

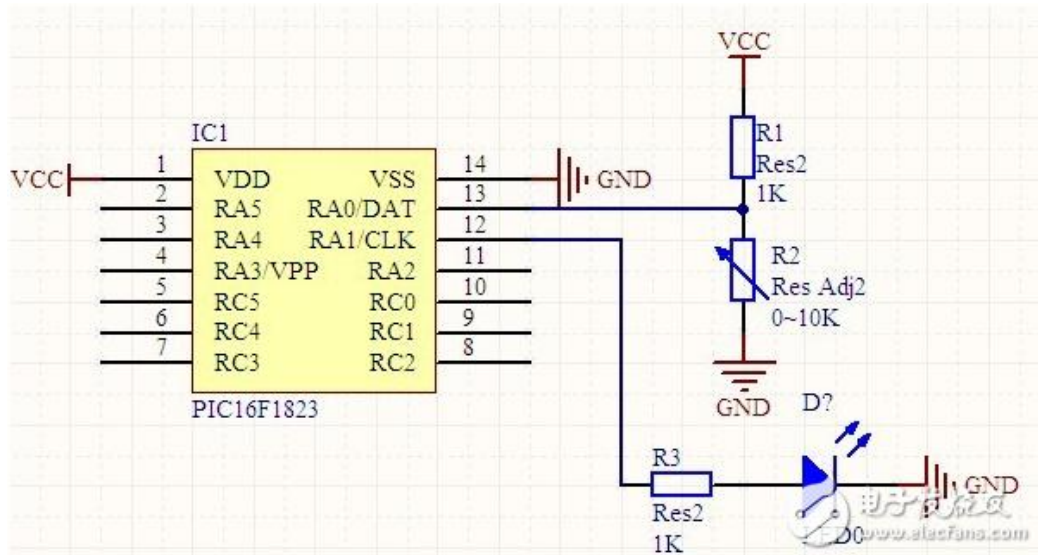
见下图我们可以看 AN0~AN7.这些都是可以配置成模拟输入的端口。只有这些引脚才能做为 AD 转换的端口。

DIP、SOIC 和 TSSOP



实例讲解

例如：下面的原理图，从 RA0/AN0 脚输入个模拟量如果电压大于 1.2v 则 LED 亮否则 LED 灭。



AD 的设置步骤

1, 设置端口

将 RA0 口设置为输入 $TRISA = 0x01$;

将 RA0 口设置为模拟 $ANSELA = 0x01$;

2, 配置 ADC 模块

选择 ADC 的转换时钟。

如何选择转换时钟呢 要根据现在的时钟频率进行选择。可以根据数据手册中的表格进行选择。

我们设置单片机的时钟频率为 32MHz，选择 ADC 周期关键不要选择阴影部分，在 32MHz 这一列 我们随意选择了 ADC 时钟周期 1us，对应的时钟源为 Fosc/32.，AD 控制寄存器 1 ADCON1 的 ADCS 《2:0》 =010
注：ADCS 《2:0》代表的意思就是 ADCS 的 0 到 2 位。

表 16-1: ADC 时钟周期 (TAD) 与器件工作频率关系表

ADC 时钟周期 (TAD)		器件频率 (Fosc)					
ADC 时钟源	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5ns ⁽²⁾	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs
Fosc/4	100	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	4.0 μs
Fosc/8	001	0.5 μs ⁽²⁾	400 ns ⁽²⁾	0.5 μs ⁽²⁾	1.0 μs	2.0 μs	8.0 μs ⁽³⁾
Fosc/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs ⁽³⁾
Fosc/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs ⁽³⁾	32.0 μs ⁽³⁾
Fosc/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs ⁽³⁾	16.0 μs ⁽³⁾	64.0 μs ⁽³⁾
Frc	x11	1.0-6.0 μs ^(1,4)	1.0-6.0 μs ^(1,4)	1.0-6.0 μs ^(1,4)	1.0-6.0 μs ^(1,4)	1.0-6.0 μs ^(1,4)	1.0-6.0 μs ^(1,4)

图注：阴影单元表示超出了建议范围。

配置参考电压

我们这里选择右对齐，所以 AD 控制寄存器 1 ADCON1 的 ADFM=1

寄存器 16-2: ADCON1: A/D 控制寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADREFS<4:0>	bit 0
bit 7							bit 0

上面将有关 ADCON1 寄存器的配置说完了。下面来讲解 ADCON0

选择 ADC 输入通道

AD 转换模块只有一个，而 AD 输入通道有 8 个 AN0~AN7.所以不可能同时进行 AD 转换，那个需要用我们就分配给那个，根据硬件我们将 AD 转换模块分配给 AN0.

所以 ADCON0 的 CHS 《4:0》 =0000;

开启 ADC 模块

ADC 模块开启，ADCON0 的 ADON=1，只是单纯的启用 ADC 模块。并不开始 AD 转换。如果不用 ADC 模块时候建议关闭。可以省点电哦!!!

开始 AD 转换

ADCON0 的 GO/DONE=1 开启 AD 转换。

4 等待 AD 转换结束

5 读取结果

一般情况下我们并不取一次的 AD 转换的值。而是取多次之后算平均值。这样来确保转换的准确性。配置 ADC 模块，有许多地方并没有讲解为什么这么配置，因为许多配置其实是比较随意的。并不是那么的绝对的。一定非要选择哪一个。当然实际的配置还是要根据你项目需求。

```
//开发环境 MPLAB X IDE ，单片机 PIC16LF1823.
```

```
#include 《pic.h》
```

```
__CONFIG  
( FOSC_INTOSC&WDTE_OFF&PWRTE_ON&MCLRE_OFF&CP_ON&C  
PD_OFF&BOREN_ON  
  
&CLKOUTEN_OFF&IESO_ON&FCMEN_ON );//这个要放到上一行去  
  
__CONFIG ( PLLEN_OFF&LVP_OFF ) ;  
  
#define ADC_NUM 8 //转换的次数  
  
#define LED LATA1  
  
void init_GPIO ( void )  
  
{  
  
TRISA = 0x01;//端口设置为输入  
  
ANSELA = 0x01;//设置为模拟输入  
  
PORTA = 0x00;  
  
LATA = 0x00;  
  
}  
  
void init_fosc ( void )  
  
{  
  
OSCCON = 0xF0;//32MHZ  
  
}
```

```
void init_AD ( void )  
  
{  
  
    ADCON1= 0xA0;//右对齐，AD 时钟为 Fosc/32，参考电压为电源电  
压，  
  
    ADCON0= 0x00;//选择通道 AN0  
  
    ADCON0bits.ADON = 1;//开启模块  
  
}  
  
unsigned int ADC_BAT_ONE ( void ) //转换一次  
  
{  
  
    unsigned int value;  
  
    value=0;  
  
    ADCON0bits.CHS =0;//选择通道 AN0  
  
    ADCON0bits.ADGO=1;//开始转换  
  
    while ( ADCON0bits.GO==1 ) ;//等待转换结束  
  
    value= ( unsigned int ) ADRESH;//强制类型转换，因为 ADRESH 是  
字符型的只能表示 8 位二进制。所以必须转换成可以容纳 10 位二进制的整  
型。  
  
    value= value 《 《8;// 将高两位左移 8 位
```

```
value += ADRESL;//低八位加入 ADRESL 的值。
```

```
return value;
```

```
}
```

```
unsigned int ADC_BAT_contiue ( void )
```

```
{
```

```
unsigned int ADV_MCU [ ADC_NUM ] , ADV_CNT , ADV_ALL;
```

```
ADV_ALL=0;
```

```
for( ADV_CNT=0;ADV_CNT <<ADC_NUM;ADV_CNT++ )//进行多次
```

AD 转换

```
{
```

```
ADV_MCU [ ADV_CNT ] =ADC_BAT_ONE ( ) ;
```

```
}
```

```
for( ADV_CNT=0;ADV_CNT <<ADC_NUM;ADV_CNT++ )//计算多次
```

AD 转换的平均值

```
{
```

```
ADV_ALL += ADV_MCU [ ADV_CNT ] ;
```

```
}
```

```
ADV_ALL= ADV_ALL/ADC_NUM;
```

```
return ADV_ALL;//得到结果返回

}

/*

*

*/

int main ( int argc , char** argv ) {

init_fosc ( ) ;//设置时钟

init_GPIO ( ) ;//设置 I/O 口

init_AD ( ) ;//设置 AD

while ( 1 )

{

if ( ADC_BAT_contiue ( ) >> 400 ) //判断输入电压是否大于 1.2V

{

LED=1;//灯亮

}

else

{
```

```
LED=0;//灯灭
```

```
}
```

```
}
```

```
}
```