

Pic32 开发实战干货版

目录

Pic32MZ 开发实战干货版	1
开发前	5
需要准备啥？这些东西都是干嘛的	5
到那里找这些资源，需要收费吗？	5
需要那些基础知识	5
IDE、编译器和 Harmony 库	6
IDE 的基本功能	6
问题：如何新建、编译、下载和调试一个 Hello world 的工程？	6
编译器的基本功能	31
Harmony 库介绍	31
如何将 Harmony 库的源码独立运行	32
IO 端口功能介绍	28
IO 端口有哪些特性	29
如何设置 IO 端口的高低	29
如何读入 IO 端口的状态	30
输入捕捉和输出比较	30
ADC 数模转换功能介绍	31
如何用轮询的方式读取 ADC	31
如何用中断的方式读取 ADC	31
ADC 与 DMA	31
UART 通信功能介绍	31
如何用简单的方式进行 UART 通信	31
如何用中断的方式进行 UART 通信	31
UART 与 DMA	31
IIC 通信功能介绍	31
如何用软件模拟的方式进行 IIC 通信	31
如何用中断的方式进行硬件 IIC 通信	31
CAN 通信功能介绍	32
如何建立一个简单的 CAN 通信链路	32

以太网通信功能介绍	32
建一个简单的以太网通信链路需要的资源	32
通过直连的方式和计算机进行通信	32
通过路由的方式和计算机进行通信	32
定时中断	33
如何写一个简单的定时中断的程序	33
如何切换不同的定时中断源	33
定时中断有那些常见的高级功能	33
看门狗计时器	33
如何写一个简单的看门狗程序	33
如何切换看门狗程序的计数源	33
Flash 读写功能介绍	33
如何读写内部 Flash	33
如何读写外部 Flash	33
如何利用自带的 IIS 搭建一个简单的声音输出系统	34
需要那些资源	34
如何搭建一个简单的 U 盘，SD 卡播放器实例	34
如何搭建一个简单的蓝牙播放器实例	34
如何搭建一个简单的 USB 耳机播放器实例	34
如何搭建一个简单的 USB 麦克风实例	34
如何利用 PIC32MZ 搭建一个蓝牙数据传输系统	34
需要那些资源	34
如何搭建一个简单的蓝牙数据传输系统实例	34
如何使用 PIC32MZ 的 Bootloader 引导程序	35
需要那些资源	35
如何搭建一个 Bootloader 引导程序的实例	35
如何进行图形界面的开发	35
进行图形界面的开发需要考虑那些问题	35
从哪里找到这些资源	35
常用的屏有那些，常用 Touch 控制器有那些	35

如何搭建一个图形界面的实例	35
如何换屏，如何换 Touch	35
如何利用 Pic32MZ 进行马达控制	35
如何搭建一个马达控制的实例	35
如何构建一个 RTOS 实例	36
如何搭建一个 embos、freertos、openrtos、threadx、uCOSII、uCOSIII 实例.....	36
以 freeRTOS 为例，如何创建、停止、挂起一个和多个任务？	36
以 freeRTOS 为例，如何进行进程间的通信	36
如何构建一个 USB 通信实例	36
如何搭建一个 USB 主通信实例.....	36
如何搭建一个 USB 从通信实例.....	36
如何搭建一个 USB 通过 Hub 通信的实例	36

软件开发前

需要准备啥？这些东西都是干嘛的

PIC32 单片机开发必备的三样东西是：Mplab X IDE，XC32，Harmony。下面我们分别讲讲这三样东西，至于开发板，容后介绍，目前做的所有工作只要有一台能上网的电脑就可以完成。

MplabX IDE 是一个集成开发环境的框架，它的作用是代码编辑，工程建立和修改，插件安装和集成调试。XC32 是 C 语言的编译器，它负责把源码编译成 HEX。Harmony 是一个代码集成开发库，它包含了很多例子和中间件可以供我们迅速开发一些现成的 PIC32 应用。

到那里找这些资源，需要收费吗？

这三样东西都是 Microchip 公司提供的，其中 IDE 和 Harmony 免费，XC32 编译器如果不使用优化功能也是免费的，如果要开启优化功能，则需要购买 license。

集成开发环境 IDE 的下载地址：<http://www.microchip.com/mplabx>

编译器的下载地址：<http://www.microchip.com/mplab/compilers>

软件库的下载地址：<http://www.microchip.com/harmony>

顺便说一句，microchip 的网站结构都是用主域名斜杠加功能的表示，比如，关于电源的解决方案就是：<http://www.microchip.com/power> 再比如，关于触摸的解决方案就是：<http://www.microchip.com/touch> 等等以此类推。此外，如果要获取对应芯片和芯片组相关的信息也可以用这种方法，例如：<http://www.microchip.com/pic32mz2048efg100> 或者 <http://www.microchip.com/pic32> 等等。下载好之后，将这几个软件安装到默认的目录中，根据提示进行选择与安装，基本上选择默认选项即可。

需要那些基础知识

阅读本书的读者应该具备 C 语言开发的知识，嵌入式开发的基础知识，如果读者缺乏这方面的知识，可以推荐读者读一读经典的谭浩强版本的 C 语言入门即可。

由于对于 PIC32 的开发往往是由大量的源代码构成的项目基础进行的非初创开发，程序结构复杂代码量大，为了便于代码管理和比较，建议读者学习 Git 或 Svn 版本管理工具。值得一提的是 MplabX 加入了 Git 和 Svn 的外壳集成，便于开发者利用 MplabX 直接进行方便的代码管理。

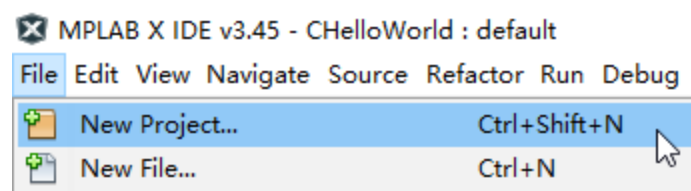
IDE、编译器和 Harmony 库

IDE 的基本功能

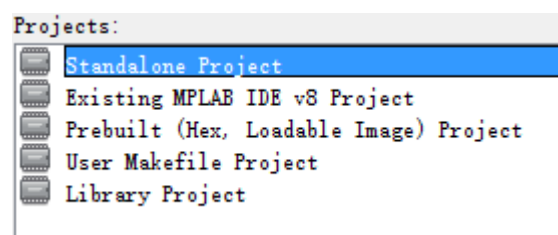
如何新建、编译、下载和调试一个 Hello world 的工程？

如何建立新工程？

安装好 IDE 之后，点击 File→New Project 之后会弹出一个对话框，在该对话框中，有如下工程类型可以建立。



Standalone Project: 这是建立一个独立的工程的选项。一般来讲，大部分的工程应用都是选择这个选项。



该菜单的其他选项意义如下：

Existing Mplab V8 Project: 这是通过现有的 Mplab V8 版本的工程建立一个新的能在 MplabX 下管理的一个工程的选项，也就是导入老款 Mplab 工具生成的.mcp 文件。

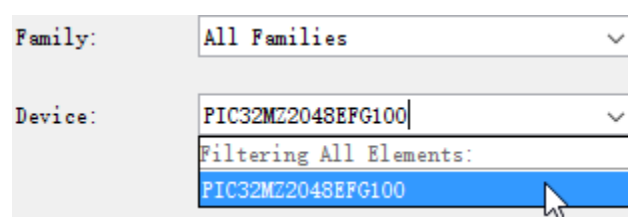
Prebuilt (Hex Loadable Image) Project : 建立烧录工程，通过这个选项，您可以建立一个独立的可以烧录 HEX 文件的工程，选择这个选项之后只能烧录不能编译。

User Makefile Project: 建立一个 Makefile 工程，不常用。

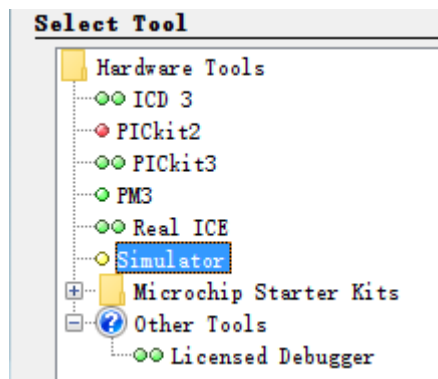
Library Project: 建立一个库工程的选项，可以将代码封包。

此外，您还可以将 IAR 工程转换成 MplabX 所识别的工程。

在这里，我们选择 Standalone Project 选项新建一个独立的工程然后进入 Select Device 页面如下图，在这个页面中选择或者输入需要的芯片型号：



选择之后点击下一步，进入开发工具的选择页面：

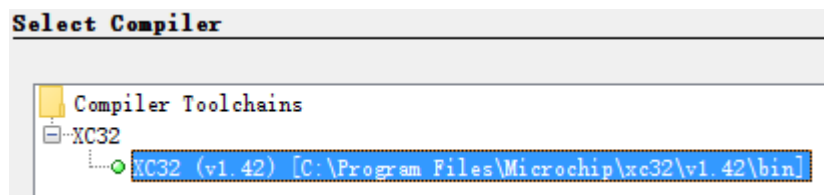


该菜单的其他选项意义如下：

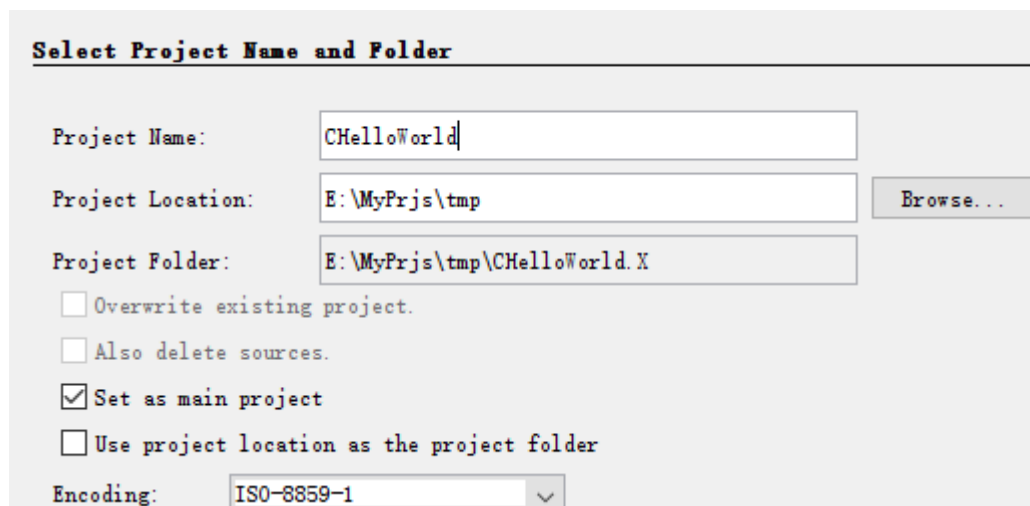
ICD3, Pickit2, Pickit3, PM3, Real ICE: 这些都是 Microchip 公司提供的开发工具的名称，通过这些开发工具可以链接电脑和目标板进行下载或者调试。目前客户使用比较多的工具是 PICKit3，它比较便宜，而且功能比较齐全，还有离机下载的功能。另外，ICD3 和 Real ICE 用的也比较多，它们比 Pickit3 昂贵一些。一般功能的开发建议使用 Pickit3 就可以了。

Simulator: 这是电脑对嵌入式板卡进行软件仿真，选择该选项则可以不用

我们选择 Simulator 然后下一步，选择对应的编译器 XC32，也就是从 <http://www.microchip.com/mplab/compilers> 下载的 XC32 编译器。



选择之后点击下一步进入如下界面：



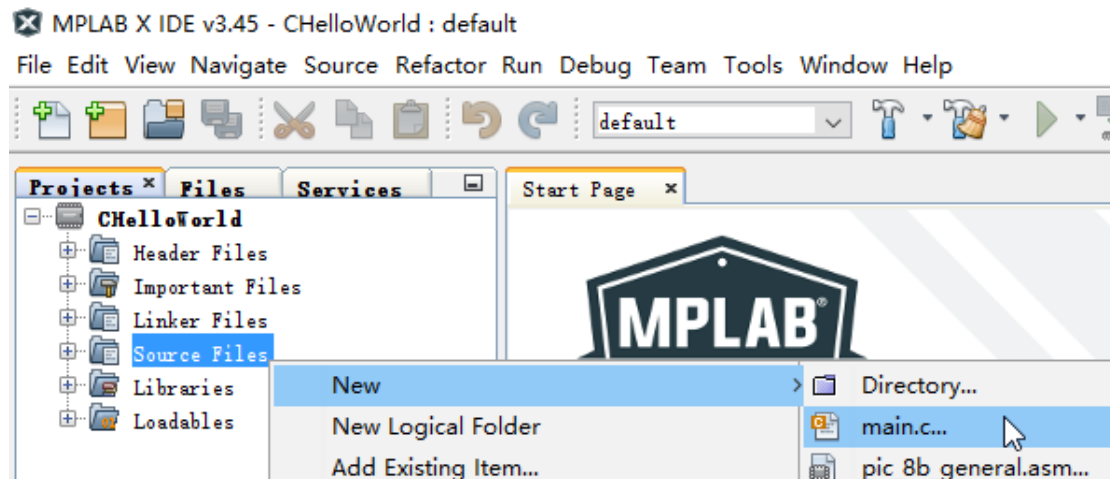
在这里，有三点要注意，首先，Project Name 最好不要选择太长和含有中文名；其次，要勾选 Set as main project 否则当 IDE 有同时开启多个工程时有可能编译的不是您新建的工程；最后，Encoding 要选择 GB2312，否则注释可能不能显示中文。选择好之后点击 Finished 之后工程创建就完毕了，其他选项一般不常用，默认即可。

小任务：从官网下载 IDE，XC32 和 Harmony 库，并且安装好。

问题：如何在新工程中添加文件和进行工程配置？

创建好工程框架之后，紧接着我们将对工程进行文件添加和配置。首先我们添加一个 main.c 文件，步骤如下：

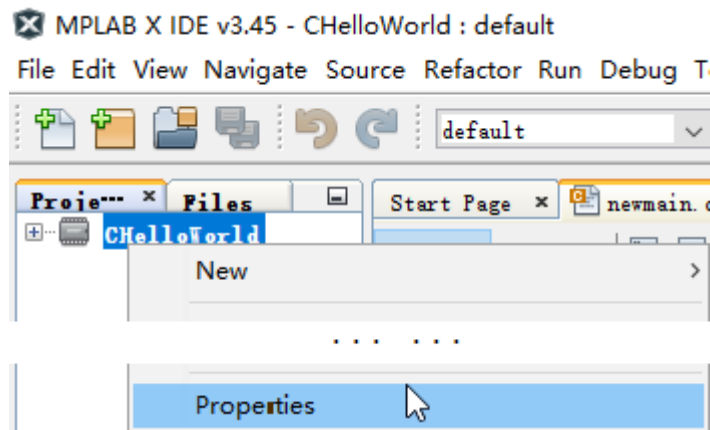
在创建好的工程项目右键点击 Source Files 虚拟文件夹，选择 New→main.c 如下图：



在弹出的界面上点击 **Finish** 生成一个新的 main.c 文件。在 main 函数里写一个 while(1); 的循环把程序挂住，写一个 volatile int i 的变量对这个变量可以做一些自加减的操作进行实验。代码如下图所述：下面我们将结合这段最简单的代码介绍如何编译、下载、调试代码：

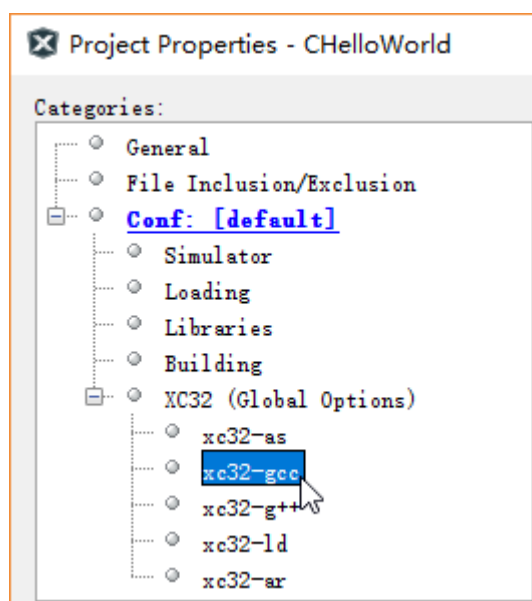
```
1  #include <xc.h>
2  void main(void)
3  { volatile int i=0;
4      while(1)
5      {
6          i++;
7      }
8      return;
9  }
```

在这之前，我们需要了解如何如何设置工程中的属性，其中包括如何设置包含路径、头宏定义、堆栈的值、优化等级等方面，其中工程属性的打开方法是在工程窗口的当前工程下点击右键，然后点击属性，如下图：

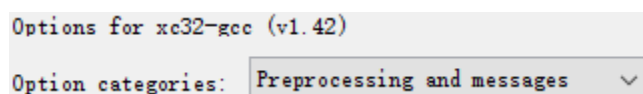


问题：如何设置预编译宏定义和包含路径？

点击之后，点击 xc32-gcc 选项，如下图：



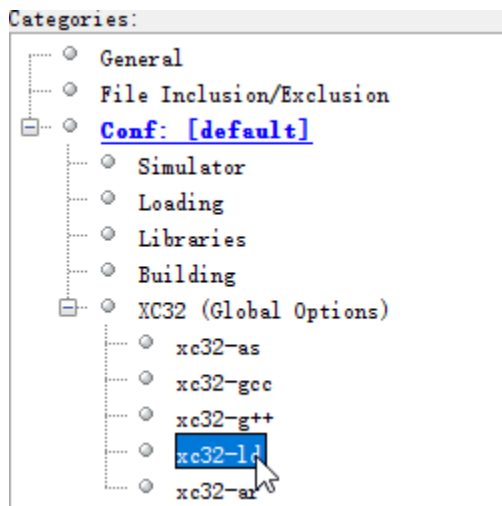
然后确定在 Option categories: 中选择 Preprocessing and messages，如下图：



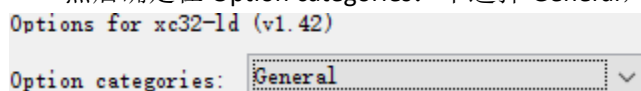
之后便可以在 Preprocessor macros 定义添加头文件的宏定义，在 Include directories 设置头文件包含路径。

问题：如何设置程序中的堆栈值？

在工程中属性配置页面点击 xc32-ld 选项，如下图



然后确定在 Option categories: 中选择 General，如下图：



之后便可以在 Heap size(bytes)中设置程序中堆的大小，在 Minimum stack size (bytes) 设置程序中栈的大小。

小任务：从官网下载 IDE，XC32 和 Harmony 库，并且安装好。

第三步：编译和向虚拟的“MCU”中下载工程

我们从两个方面对编译、下载、调试进行介绍，快捷键和工具条。首先我们介绍编译、下载的工具条：



(虚拟机界面，中间三个图标为灰色，不能点击)



(连接目标板界面，全部图标为亮色，可以点击)

第一个锤子图标就是编译链接工程，对应快捷键（Alt+R 然后 B）。第二个带有符号的锤子图标就是清除中间文件重新编译链接，对应快捷键（Alt+R 然后 C）。第三个右箭头图标是运行，对应快捷键（Alt+R 然后 C）。第三个下箭头图标是下载。第四个上箭头图标是读取。第五个环形箭头图标是复位。第六个小三角图标就是调试运行。


至此，程序的编译，下载，运行就介绍完毕了。

第四步：以虚拟机为例，如何调试工程？

点击  图标将 HelloWorld 工程下载到虚拟机，此时在工具栏上应该出现如下图标：



栏编程如下状态：

此时，按下  按钮暂停该程序，此时工具

第一个方块图标的功能是停止运行程序，对应快捷键（Shift+F5），第二个双竖杠的功能是暂停程序，该操作对应的快捷键是（Ctrl+Alt+Pause），第三个双箭头图标是 Reset 程序，第四个绿色白三角图标是继续运行程序，该操作对应的快捷键是（F5），第五个弯箭头的图标的功能是 StepOver，也就是跨过函数单步运行程序，该操作对应的快捷键是（F8），第六个向下的箭头功能是 StepInto，也就是进入函数单步调试程序，该操作对应的快捷键是（F7）第七个图标是 StepOut 也就是跳出函数进行单步调试，该操作对应的快捷键是（Ctrl+F7），第八个图标是将程序指针（PC）地址强制设置为光标指示之处，第九个图标是将光标强制移动到程序指针（PC）指向之处。

备注：程序指针的作用简言之就是控制程序当前程序的运行，它指向哪里，程序就运行到哪里，反之亦然，程序运行到哪里，它就指向哪里。

问题：如何查看和临时修改变量的值？

方法 1：将光标移动到对应的变量上，则该变量显示当前的值。

```
while(1)
{
    Address = 0x8007FFC8, i = 0x000993D9
    i++;
}
```

该方法只能查看变量。

方法 2：打开变量观察窗口，输入要查看的变量

主菜单→Window→Debugging→Variables 该操作对应的快捷键是（Alt+Shift+1）

方法 3：打开 watch 观察窗口，输入要查看的变量

主菜单→Window→Debugging→Watches 该操作对应的快捷键是（Alt+Shift+2）

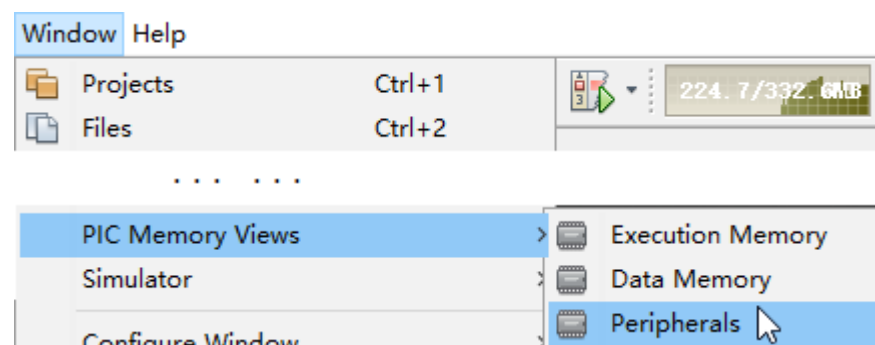
方法 2 和方法 3 可以查看与修改变量。

问题：如何查看和临时修改特殊功能寄存器的值？

要回答这个问题，首先要知道什么是寄存器。寄存器是中央处理器内的组成部分。寄存器是有限存贮容量的高速存贮部件，它们可用来暂存指令、数据和地址。在 MCU 中有些外设被映射成为了寄存器，比如 IO 端口，UART 通信波特率寄存器，IIC 缓存寄存器等等。这些不执行通用寄存器存储功能而执行某些特殊功能的寄存器被称之为特殊功能寄存器。特殊功能寄存器英文缩写为 SFR，掌握 SFR 对调试非常重要，例如在调试的过程中，我们对某 IO 端口的特殊功能寄存器进行了设置，则相应端口的真实的高低状态就会发生对应的改变。

打开特殊功能寄存器的窗口如下图，在主菜单上打开

Windows→PIC Memory Views→Peripherals



然后，在调试的时候，就可以查看和修改对应外设的特殊功能寄存器了，比如 UART 或者 IO 端口等等，例如 IO 端口 A（LATA）如下图：

Peripherals	Terminal	Trace - CHelloWorld	Output	CPU Registers	RegisterTrace
Address	Name	Hex	Decimal	Binary	Char
BF81_01C0	IPC8	0x00000000	0	00000000 00000000 00000000 00000000	'....'
BF81_01D0	IPC9	0x00000000	0	00000000 00000000 00000000 00000000	'....'
BF81_0030	IPTMR	0x00000000	0	00000000 00000000 00000000 00000000	'....'
BF86_0030	LATA	0x00000000	0	00000000 00000000 00000000 00000000	'....'
BF86_0130	LATB	0x00000000	0	00000000 00000000 00000000 00000000	'....'
BF86_0230	LATC	0x00000000	0	00000000 00000000 00000000 00000000	'....'

Memory Peripherals Format Individual

点击 HEX 或者 Decimal 便可以修改 LATA 寄存器的值，在方向寄存器设置为输出，并且连接好硬件的情况下，端口会产生电平高低的变化了。其他寄存器设置类似。

小任务：新建一个工程，然后用虚拟机进行调试，将 LATA 这个寄存器修改一下。

讲完了软件的基本功能，我们把目光投向硬件，下一个章节将以 Altium Design14 为例讲解如何进行简单的硬件开发，我们的目标是做一块基本的能下载的实验板。

硬件开发前

需要准备啥？这些东西都是干嘛的

PIC32 单片机硬件开发必备的两样东西是：画图软件（本文以 Altium Designer 为例），原理图和 PCB 的库转换软件 UltraLibrarian。下面我们分别讲讲这两样东西，目前做的所有工作只要有一台能上网的电脑就可以完成，当电路板设计完成之后，需要进行 PCB 的制板。

Altium Designer 是一个集成原理图、PCB 的设计工具，它的作用是进行原理图，PCB 的绘制，库文件的编辑，工程建立和修改，插件安装和集成调试。

到那里找这些资源，需要收费吗？

UltraLibrarian 和 AltiumDesigner 都是需要收费的，其中 UltraLibrarian 有功能简单的免费版，基本也够用，下载地址如下：

PCB 原理图库转换器 UltraLibrarian 的下载地址：<http://www.microchip.com/CAD> 在网页点击如下链接：

Ultra Librarian Installer Program for CAD/CAE Schematic Symbols 9/26/2014 12:05 PM 85473 KB

下载 UltraLibrarian 然后安装之。

AltiumDesigner 的下载和安装本文不赘述了，有兴趣的读者可以参照其他的书籍。

如何使用 UltraLibrarian 生成 AD 的图库？

以 PIC32MZ2048EFE100 和 AD14 为例，我该如何生成 AD 可用的图库呢？首先，要下载


PIC32 的 BXL 文件，下载地址如下：

PIC32 CAD/CAE Schematic Symbols

10/15/2012 2:39:46 PM


11896 KB

下载之后将压缩包解压之，会发现很多的 BXL 文件。然后，点击

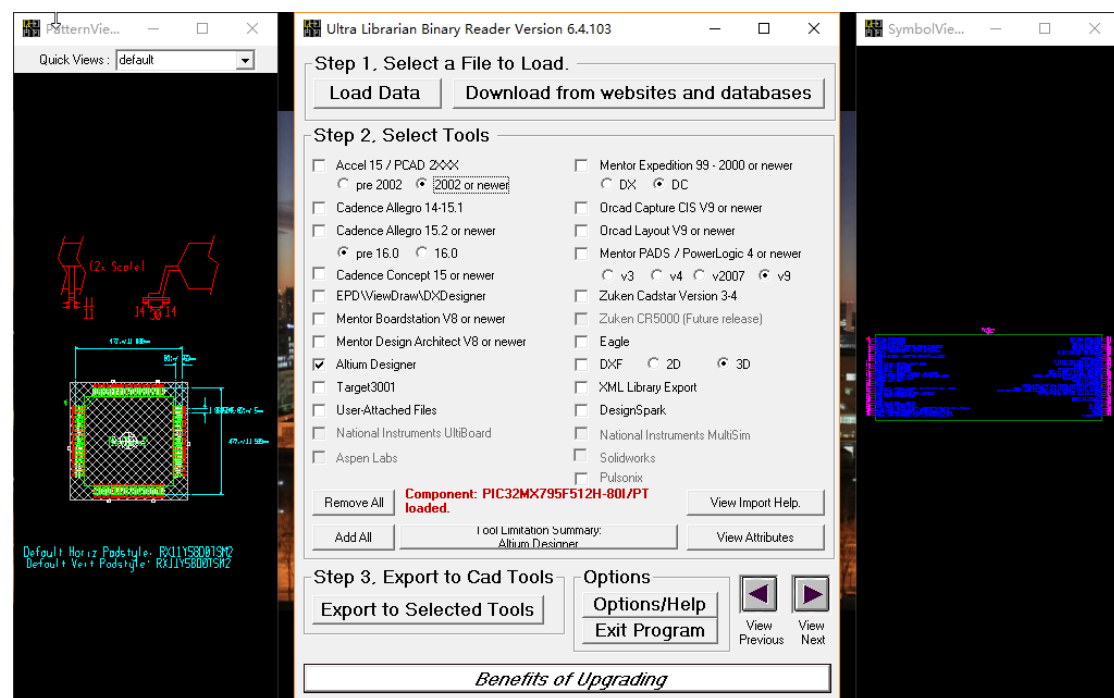
UltraLibrarian 安装目录下的  ULADICIS.exe 图标，启动程序之后点击



点击 LoadData 按钮载入对应的 bxl 文件：

 PIC32MZXXXEF(G&slash_H&slash_M)100-I&slash_PF.bxl。

然后你可以看到一个由三个独立对话框组成的界面：

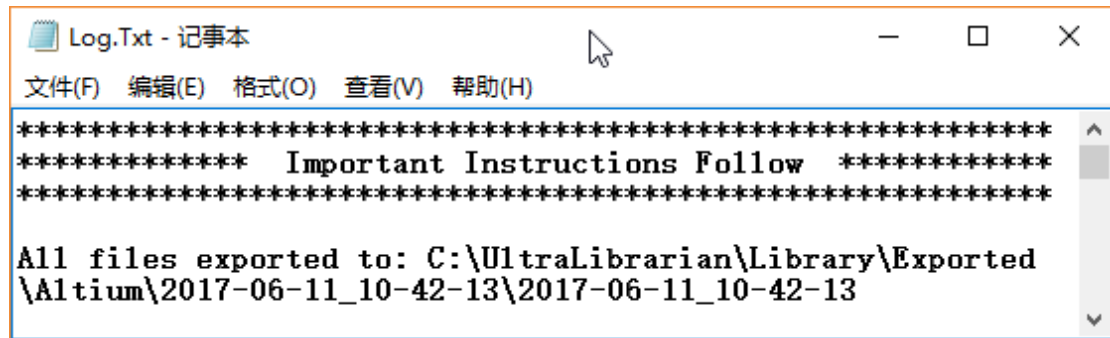


在这个界面中，左边的对话框是芯片的机械加工尺寸图，右边的对话框是芯片的逻辑引脚图，这两个界面都可以用按下鼠标邮件进行平移，用上下键来进行缩放。

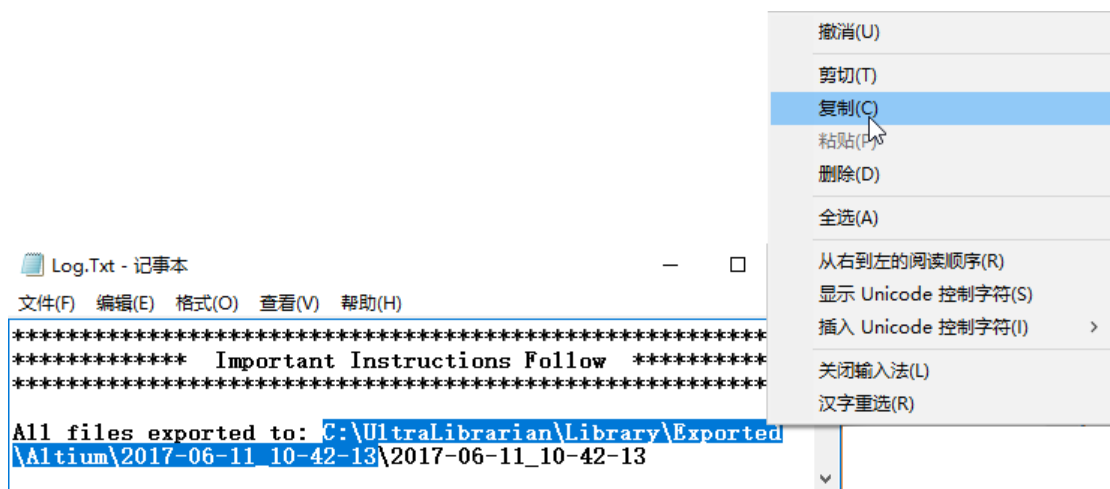
最主要的是中间的对话框，它是负责进行格式转换的，可以看到有很多的方形复选框，我们需要的是 Altium Designer 的原理图库进行转换，选择之。选择之后点击 Export to Selected Tools 按钮进行转换，如上图所示。导出原理图的按钮如下图所示：

Export to Selected Tools

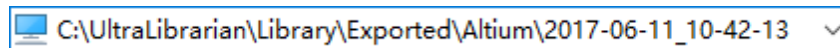
转换完毕之后会有一个 Log.Txt 的文本文件自动打开，在这个文本文件当中，有类似这样的一段路径：



这个路径显示了 Altium Designer 原理图库所需的资料所在的路径，要注意的是，我们拷贝该路径只需要拷贝到倒数第二层目录即可，如下图所示：



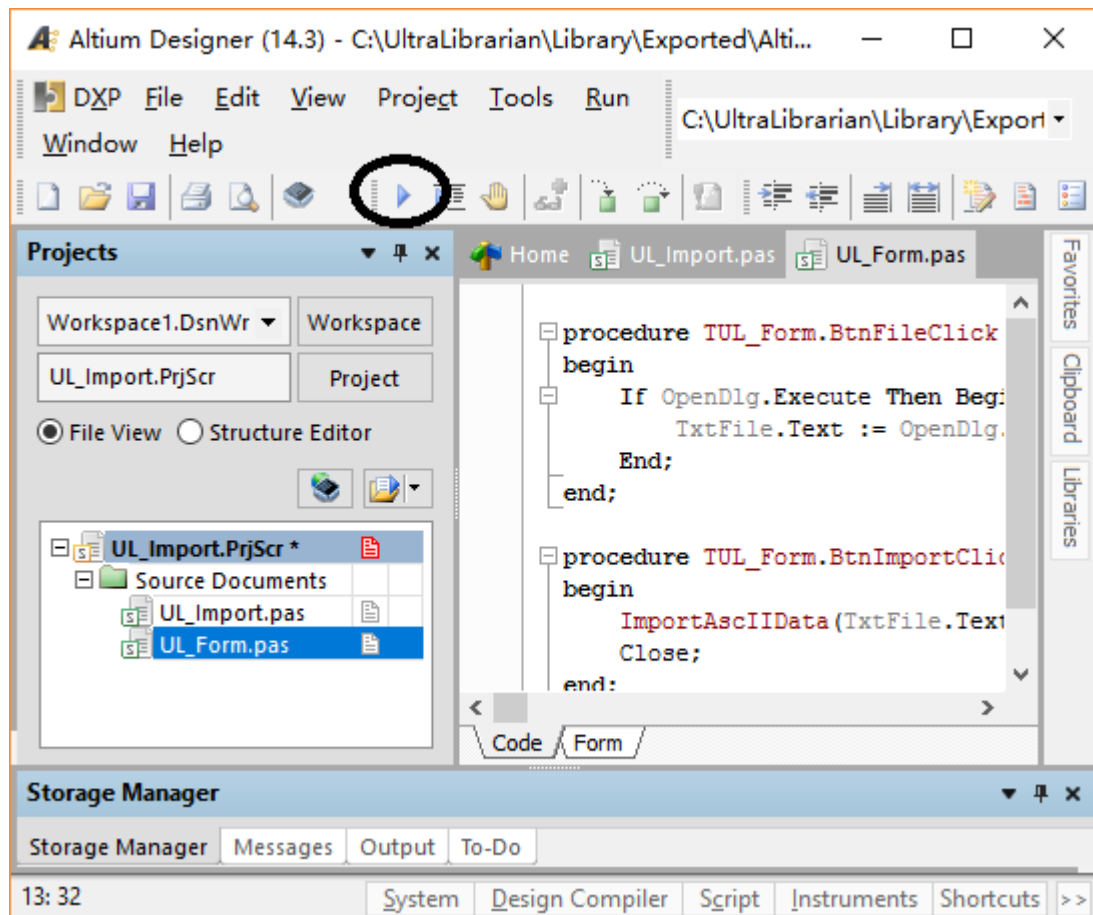
然后，将该路径粘贴到电脑的地址栏当中，回车，如下图所示：



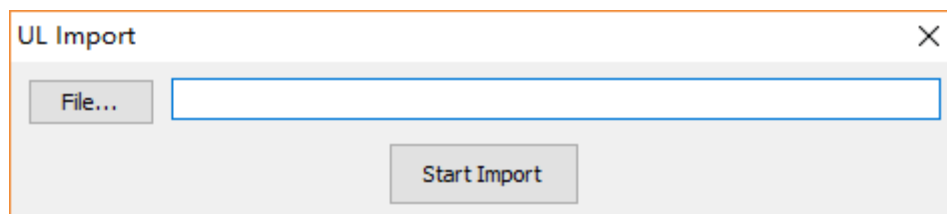
跳转到该路径之下，我们可以发现如下 4 个文件：

- UL_Form.dfm
- UL_Form.pas
- UL_Import.pas
- UL_Import.prjScr

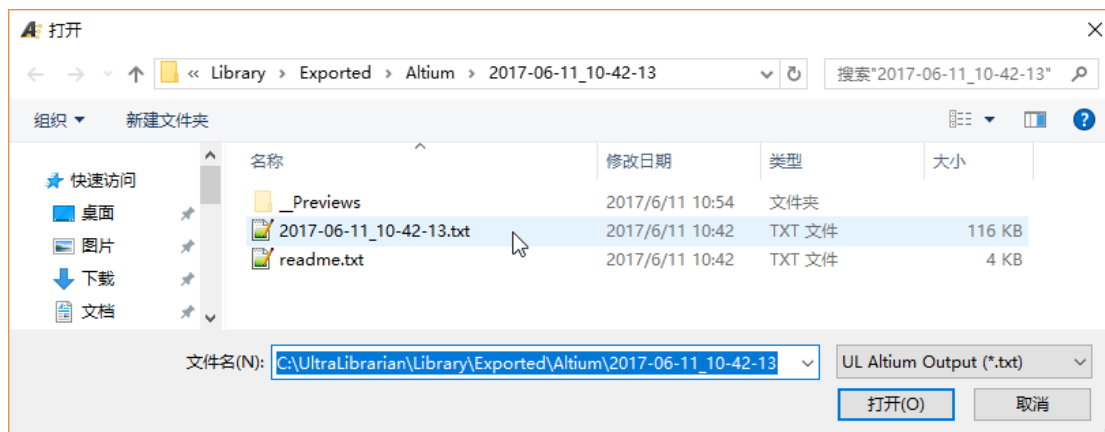
之后我们用 Altium Deisgner 打开 **UL_Import.prjScr** 文件，然后在工具栏中点击一个向右的三角形启动 UL_Form.pas 的脚本文件。如下图所示：



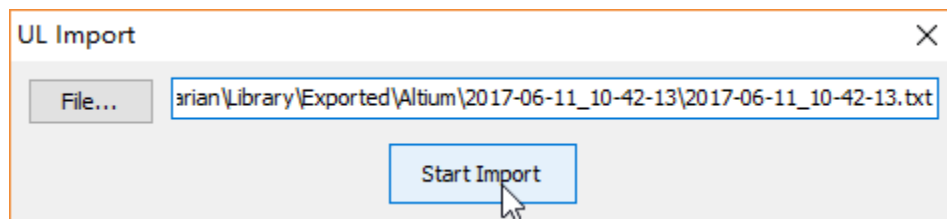
选择 UL_Form.pas 启动导入对话框如下图所示：



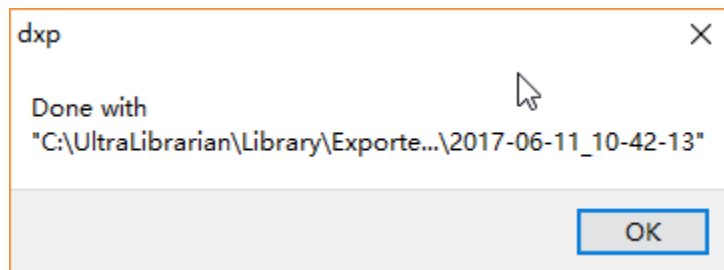
点击 File 按钮，并且贴入上述 Log.Txt 中选择的地址，点击一下打开按钮跳入对应的路径，然后选择那个带日期的 TXT 文件，它是 CAD 的中间转换文件，选择之后点击打开按钮，如下图所示：



然后回到 UL Import 界面，Start Import:



转换完毕，大功告成：



此时我们在 C:\UltraLibrarian\Library\Exported\Altium\2017-06-11_10-42-13 目录下发现多了一些文件，注意这些文件名和目录名会随着时间的变化而变化，请读者自行生成，不要照搬本书结果：

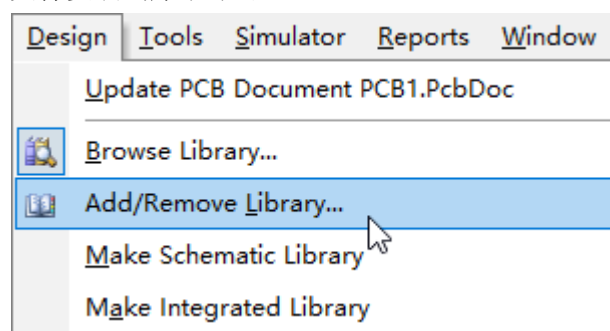
	2017-06-11_10-42-13.LibPkg	2017/6/11 11:22	Altium Integrate...	36 KB
	2017-06-11_10-42-13.PcbLib	2017/6/11 11:22	Protel PCB Library	166 KB
	2017-06-11_10-42-13.SchLib	2017/6/11 11:22	Altium Schemati...	14 KB

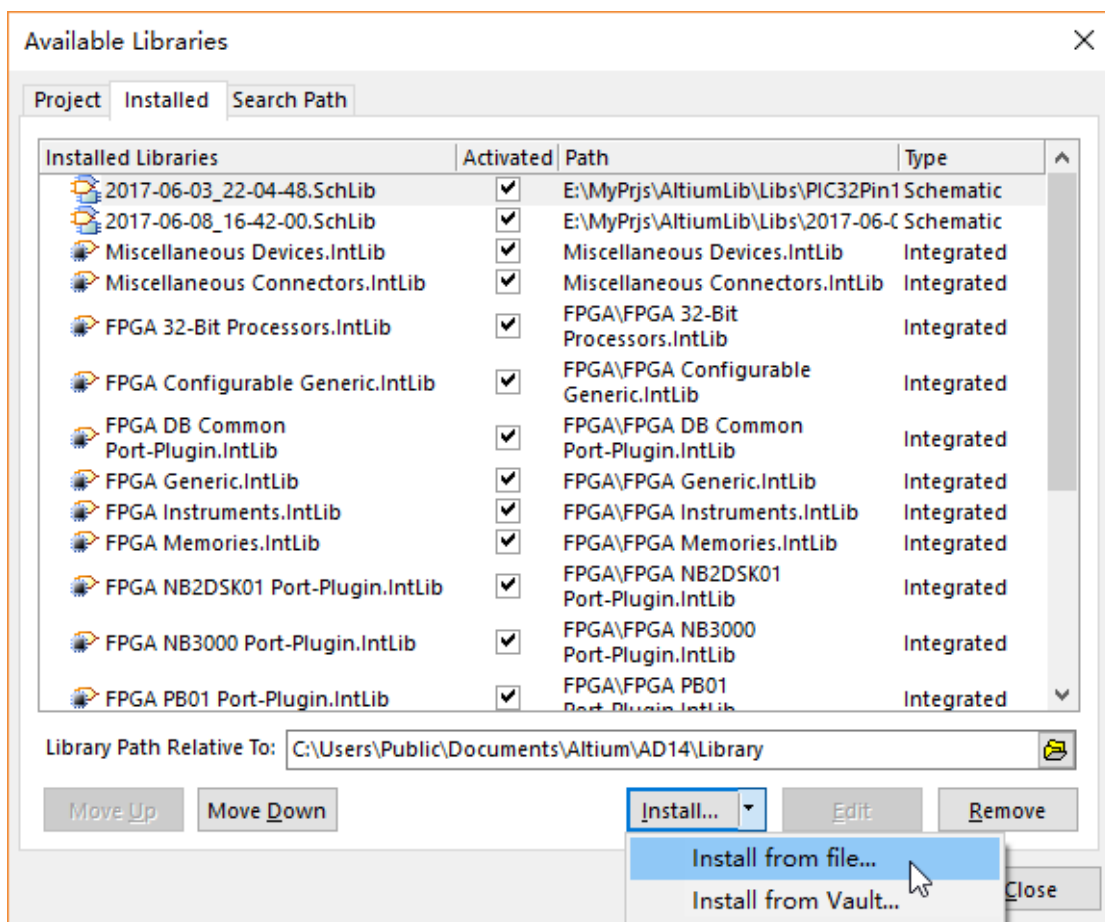
小任务：

1. 选择一个 PIC32MZ 的 MCU，然后安装 UltraLibrarian 转换出 SchLib 和 PcbLib 两个文件。
2. 安装和运行 AltiumDesigner

如何将生成的库文件添加到 AltiumDesign 当中？

把库文件添加到 AltiumDesign 当中比较简单，在 AltiumDesign 当中选择 Design→Add/Remove Library 即可，然后在下一个界面中点击 Install From File，然后选择对应的库文件安装到库中即可。

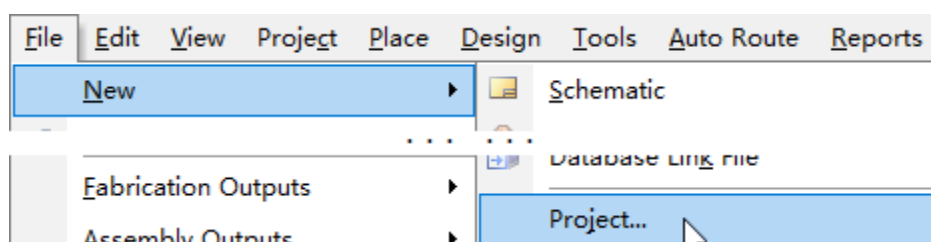




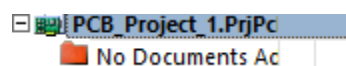
接下来我们将实际建立一个工程来说明如何使用我们生成的库。

我们新建一个简单的工程：

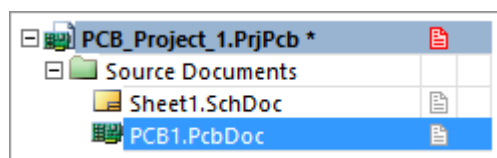
首先我们点击 File→New→Project



在 Project 的对话框中选择对应的基板模版，这里我们选择缺省值 PCB Project 即可。然后在左边的 Projects 窗口有如下图的对话框：

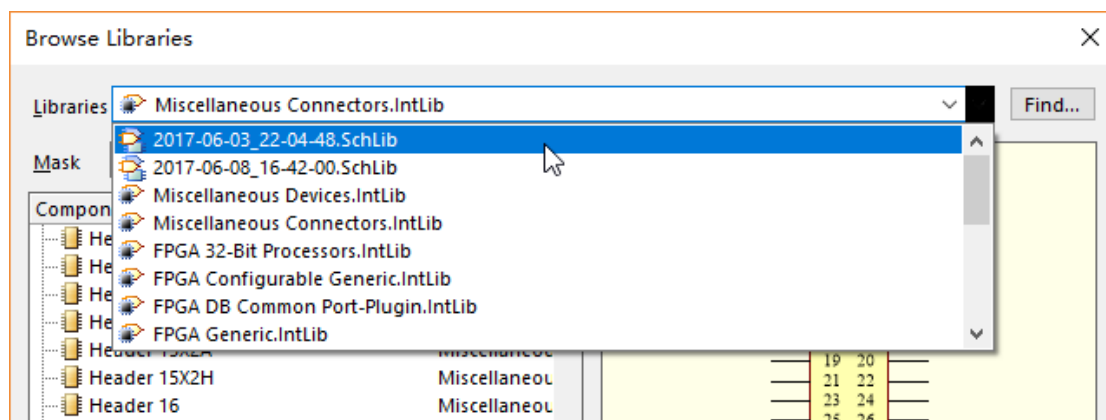


在 PCB_Project_1.Prjpcb 条目中点击右键，添加一个原理图文件和一个 PCB 文件，添加完毕之后如下图所示：

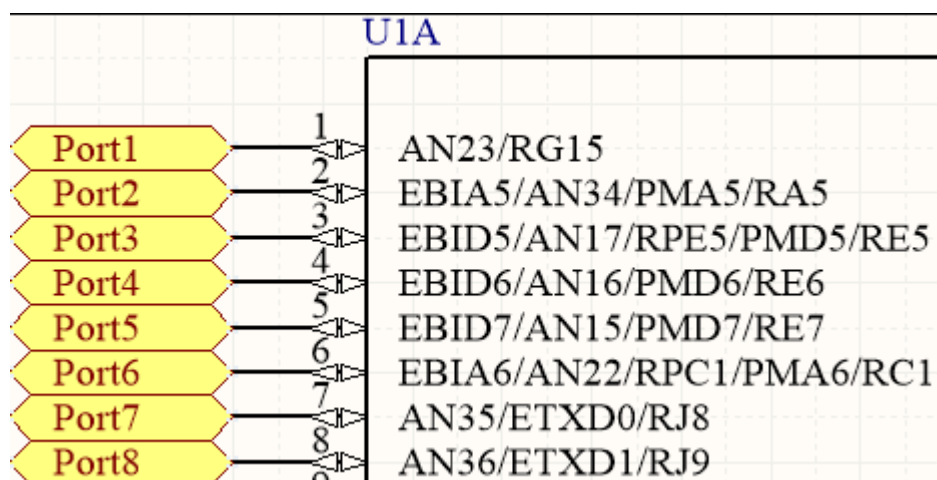


打开 Sheet1.SchDoc 之后，我们先布主芯片：

在 Sheet1 页面当中空白处，点击右键，然后点击 Place→Part...打开 Place Part 中点击 Choose 中打开对话框，然后在 Libraries 下拉框选择刚才生成的 PIC32 的库文件，如下图所示：

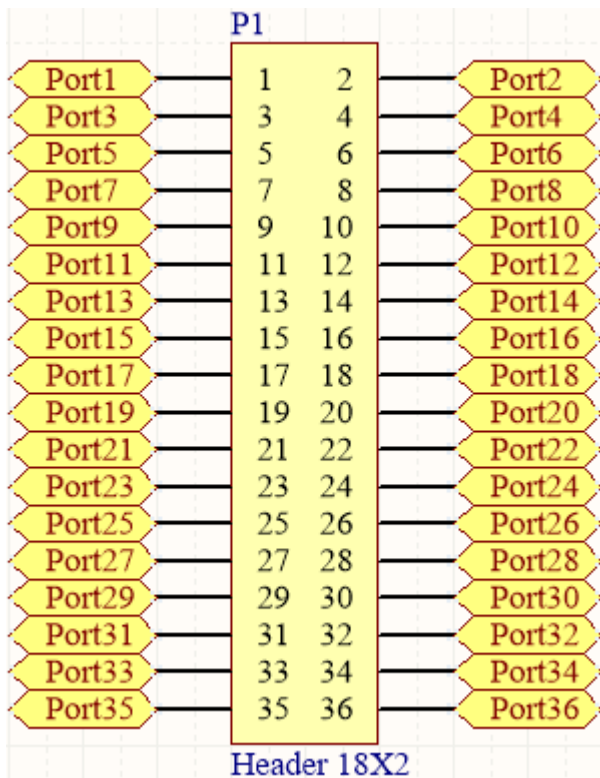


在上图中文件名以当时生成的为准，也可以是自定义的名称，不要拘泥于本书所述。把 PIC32 的元件布好之后在每一个引脚上添加一个 port 端口，名称和 Pin 对应，如下图所示，



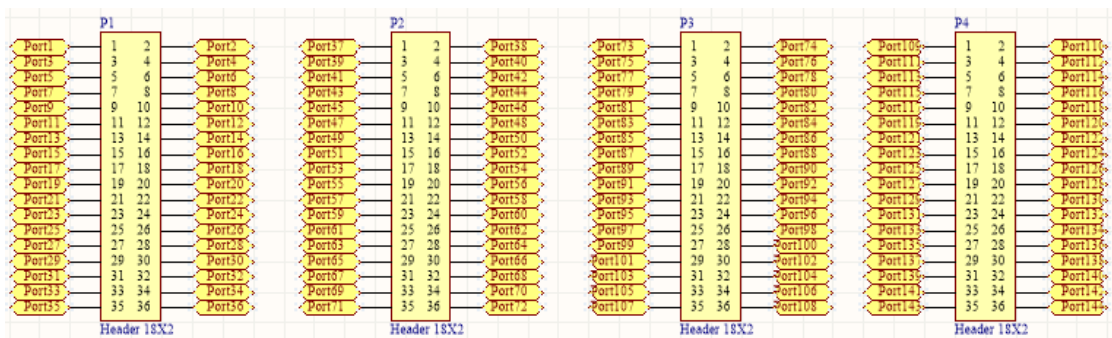
由于文件比较大，本图只给出了一部分原理图，余下的按照规律添加，有一个小技巧是可以使用 smart paste 功能快速添加 Port1~Port144

下面再添加一个原理图文件 Sheet2，然后在 Sheet2 页面当中空白处，点击右键，然后点击 Place→Part...打开 Place Part 中点击 Choose 中打开对话框，然后在 Libraries 下拉框选择 Miscellaneous Connectors.IntLib 选择 Header 18X2 选择项，然后点击 OK→OK，然后给该元件定名 P1，之后在 1~36 号端口上布好 36 个 Port，如下图所示：

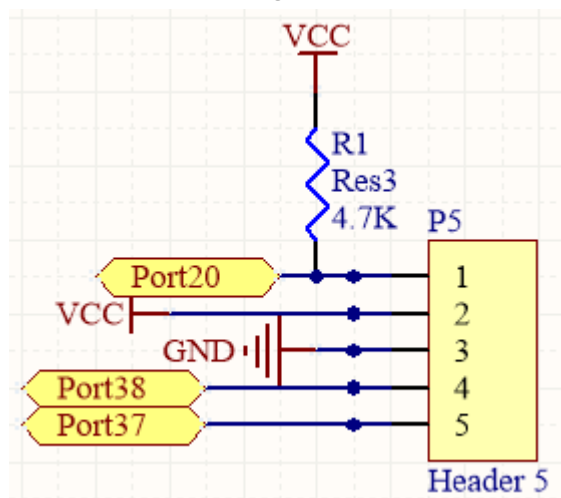


小技巧：在重复布 1~36 个端口的时候可以选择 smart paste 的功能以增加效率，下同。

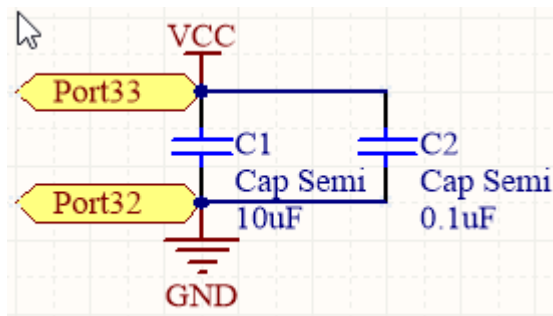
重复布 P1~P4，共计 4 个端口 144 个 Pin，如下图：



布好最简系统的 Debug 工具接口，如下图：

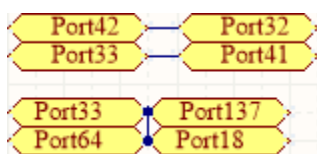


布好最简系统的电源和地，如下图：



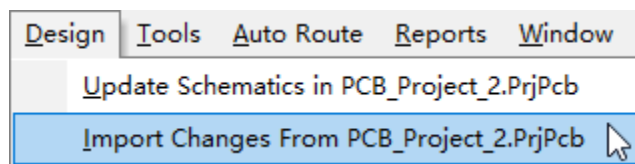
每一个电源和地都要如此处理，加一个 10uf 和 0.1uf 的电容，包括 AVSS 和 AVDD。

将 VDD 和 AVDD，VSS 和 AVSS 连接起来，另外为了布线方便，笔者讲部分 VDD 连接在了一起，如下图：

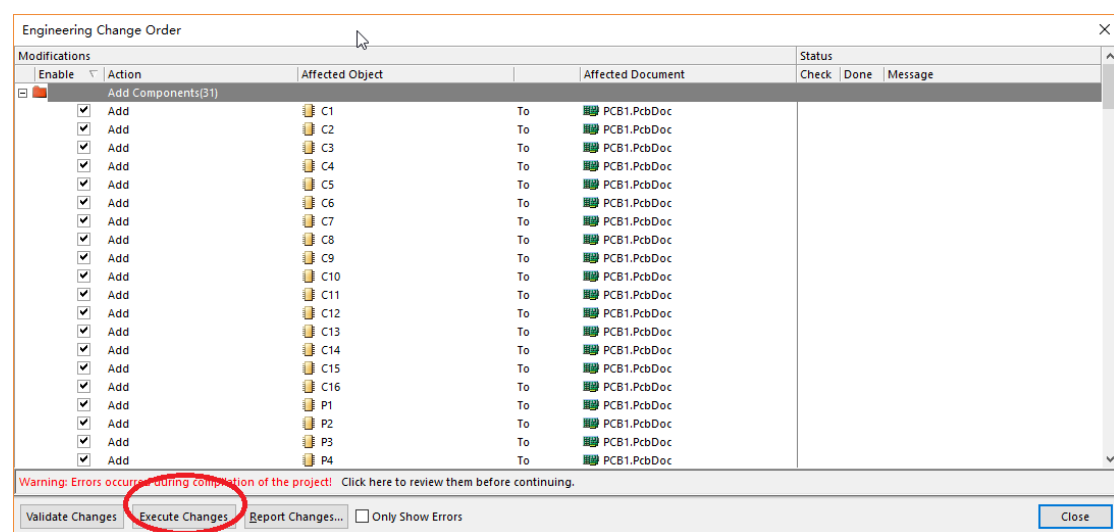


此外，笔者添加了电源引出线、LED、上拉电阻位方便使用，就不赘述了，有兴趣的读者可以自己尝试。

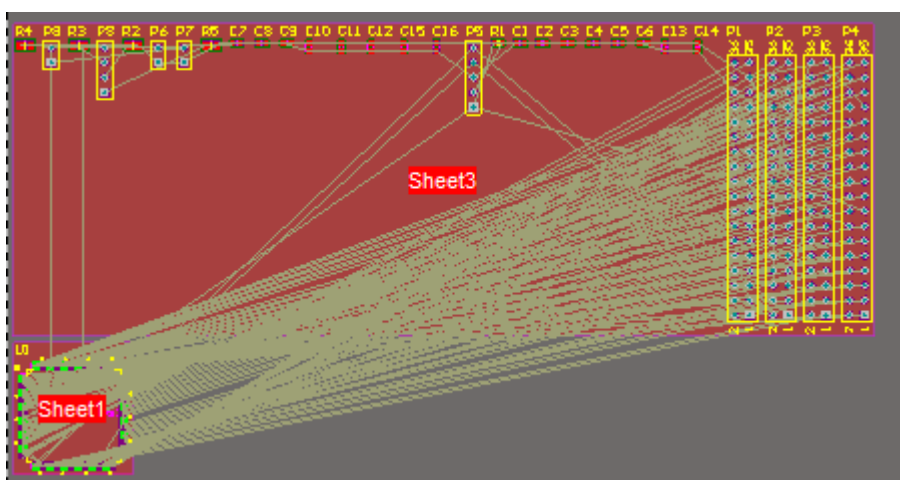
线路布好之后，保存之，然后打开工程中的 PCB 文件，准备导入网表，如下图所示：



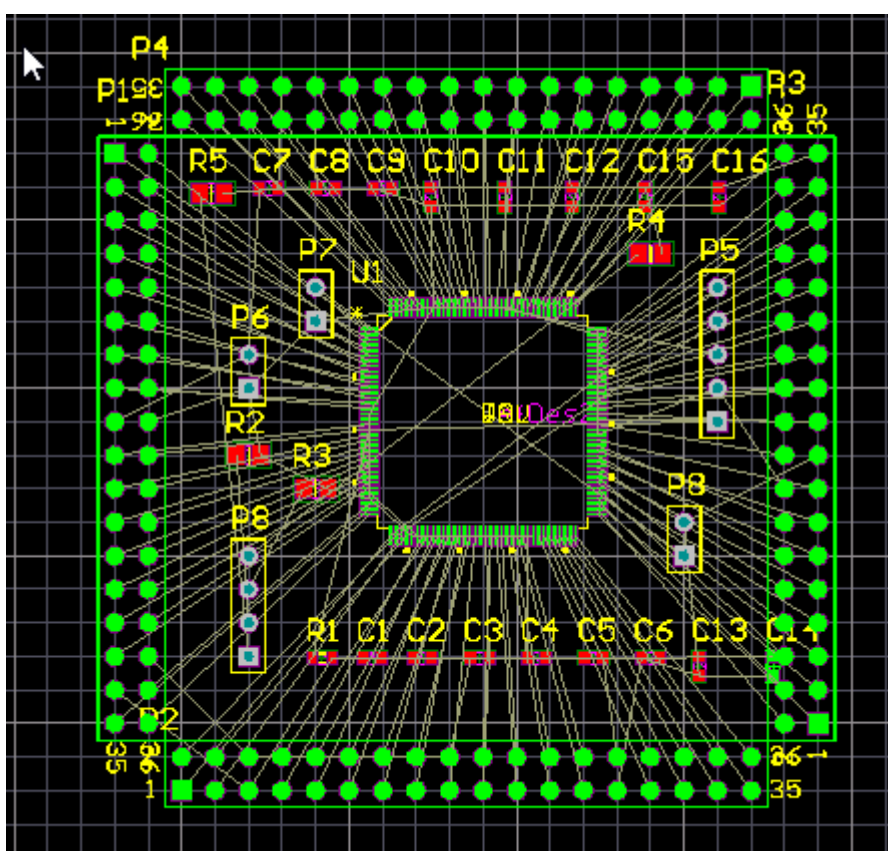
点击导入网表的按钮之后出现如下界面，找到 Execute Changes 按钮，点击一下，等网表导入完毕之后点击 Close：



如果一切顺利，会出现如下的图像：



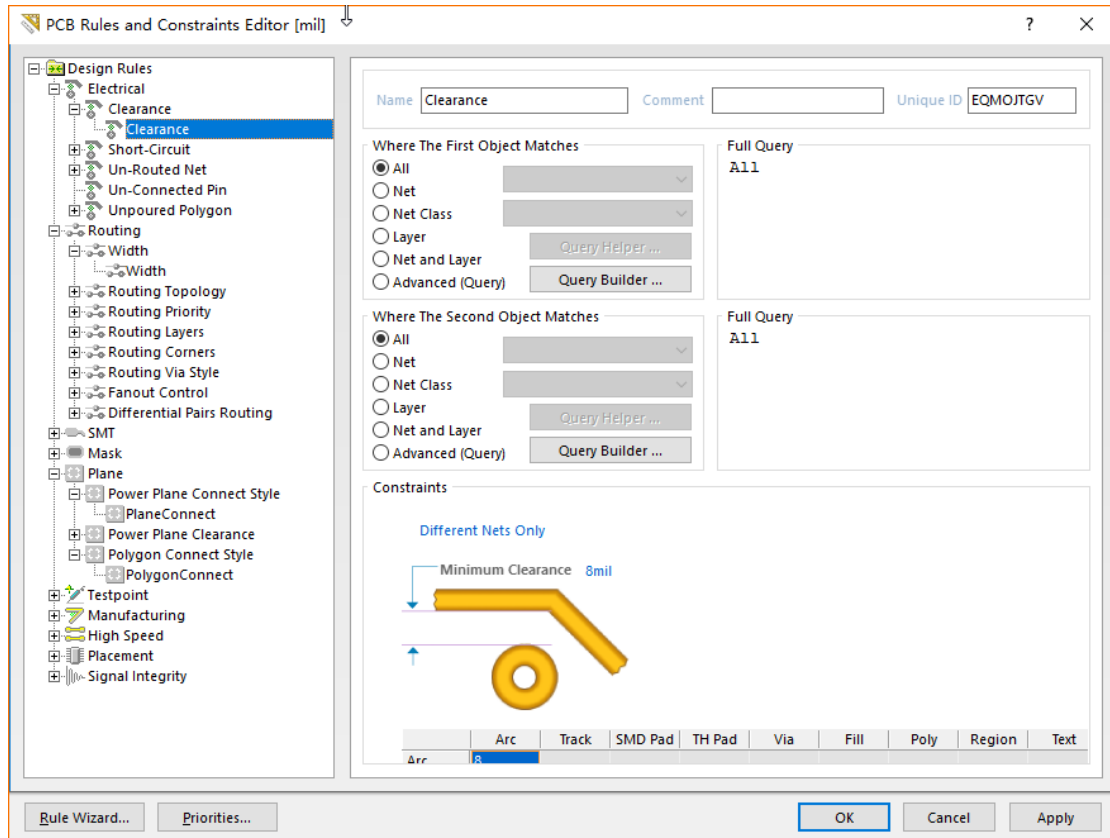
此时，将所有的单元按照顺序布好，并且删除掉 Sheet1 等的红框如下图所示：



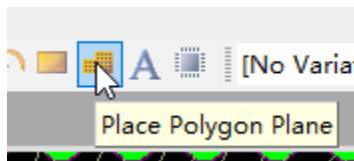
接下来就是针对各个元件进行精雕细刻的布线了，具体步骤本文不赘述了，只是扼要介绍一下常用的经验。

1. 对于有规律的成组链接，可以采用总线布线的方法加快布线进度。
2. 默认的安全距离过小，会引起小绿叉错误，设置安全间距，从而解决绿色小叉叉的问题，同时可以解决无法总线布线的问题，如下图所示：

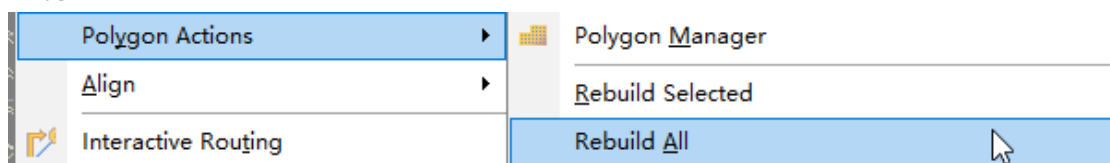
2. 默认的安全距离过小, 会引起小绿叉错误, 设置安全间距, 从而解决绿色小叉叉的问题, 同时可以解决无法总线布线的问题, 如下图所示:



3. 铺铜，首先选择 Place Polygon Plane 按钮，如下图：

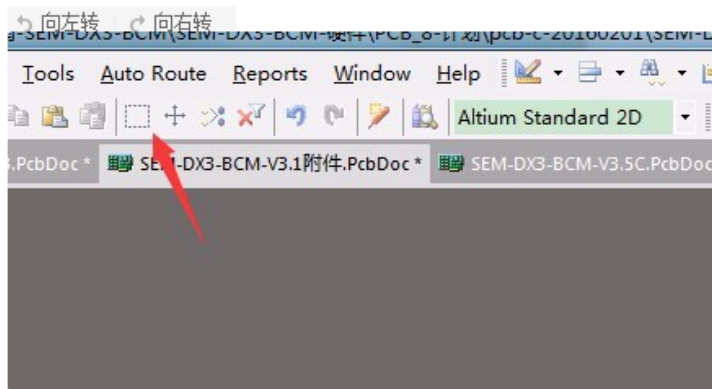


然后选择铺铜的区域，然后选择实体铺铜，网眼铺铜和边线铺铜，最后在阴影部分点击右键 Polygon Actions→Rebuild All 进行铺铜，如下图。



4. 如何旋转电路板：

1. 在PCB工具栏选择，选中工具，将板全部选中，如图：



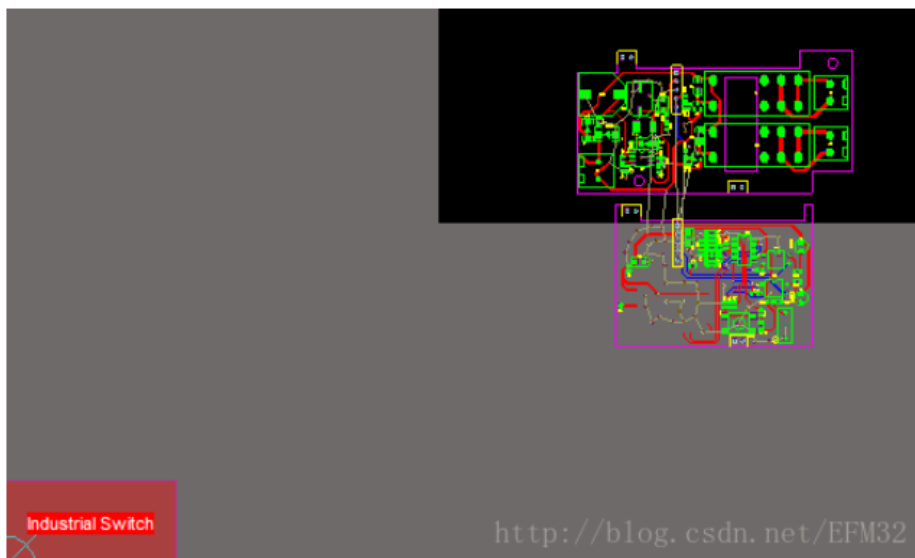
2. 再选择移动工具，选中板的任意位置，在移动时按键盘的空格键，就可以旋转90°了，如图：



5. 如何解决 Room Definition Between Component on TopLayer and Rule On TopLayer 错误？

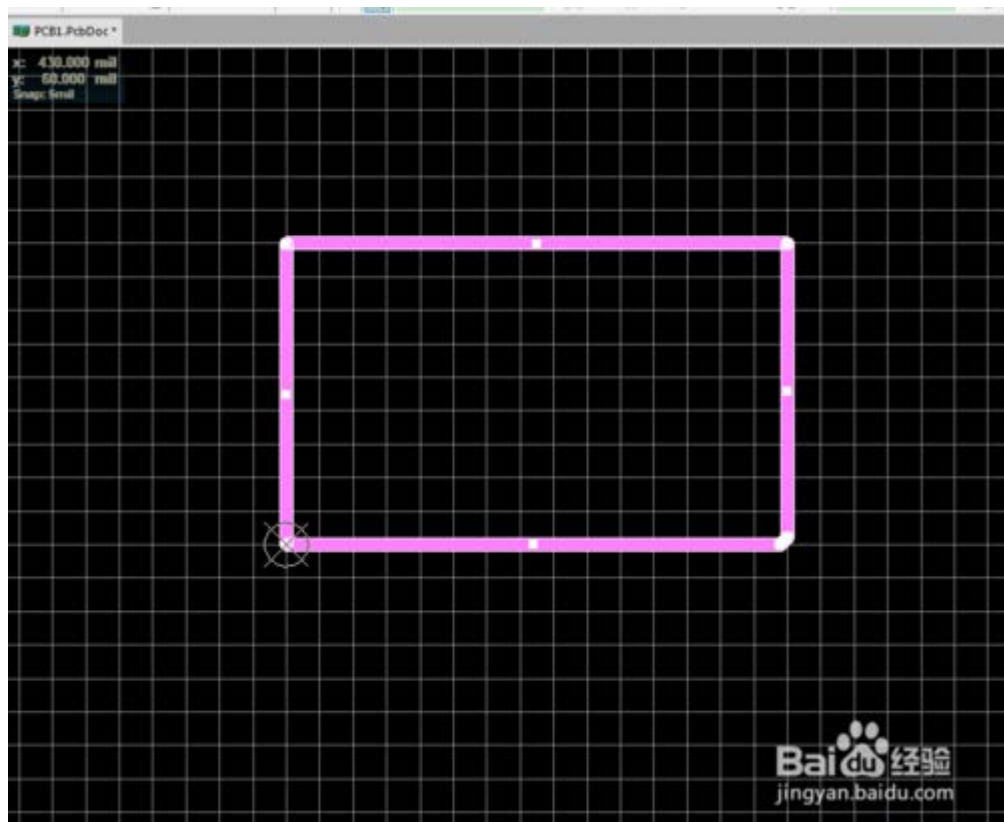
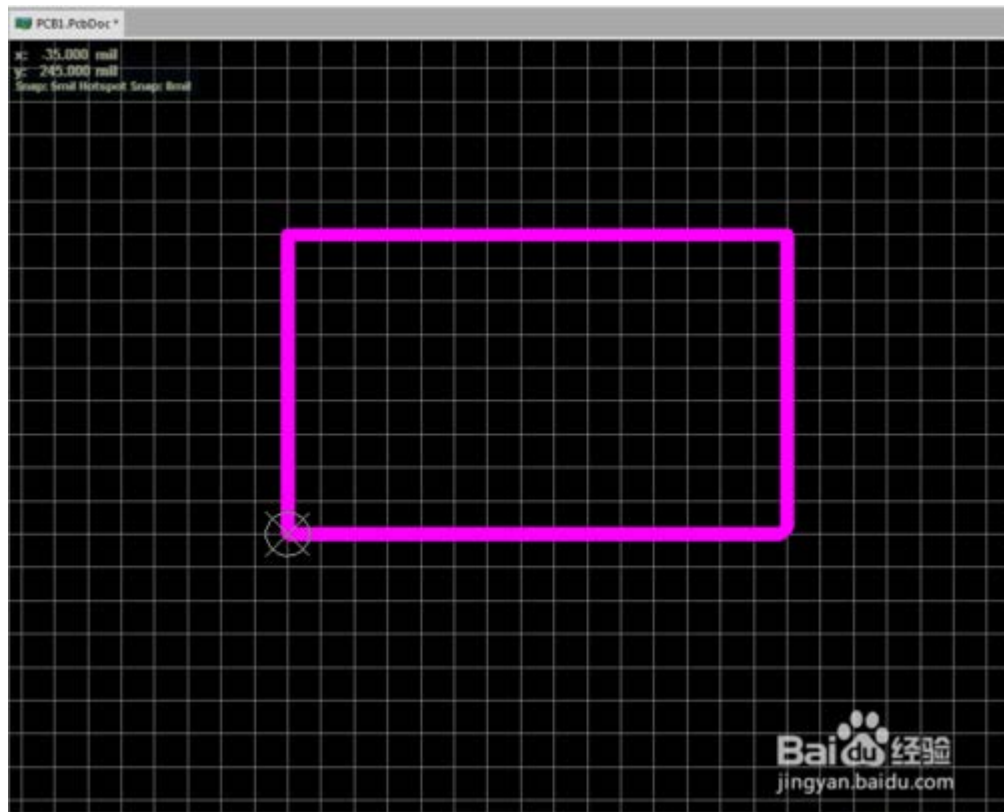
1. 从原理图导出到PCB之后，提示“Room Definition Between Component on TopLayer and Rule on TopLayer”。

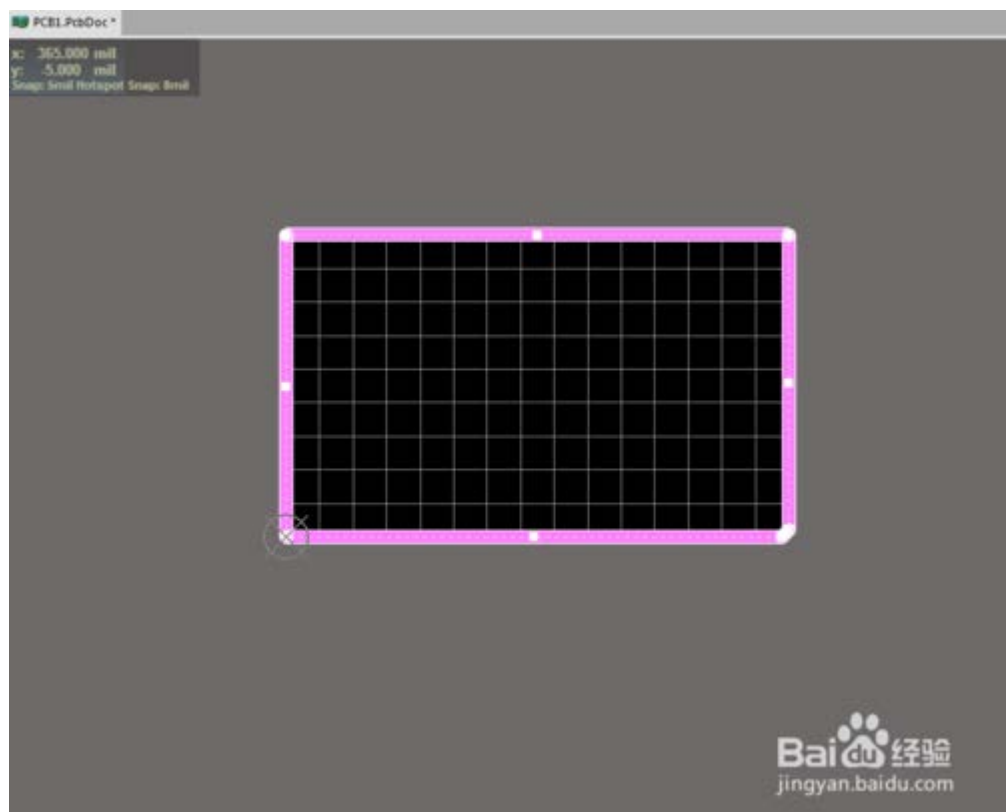
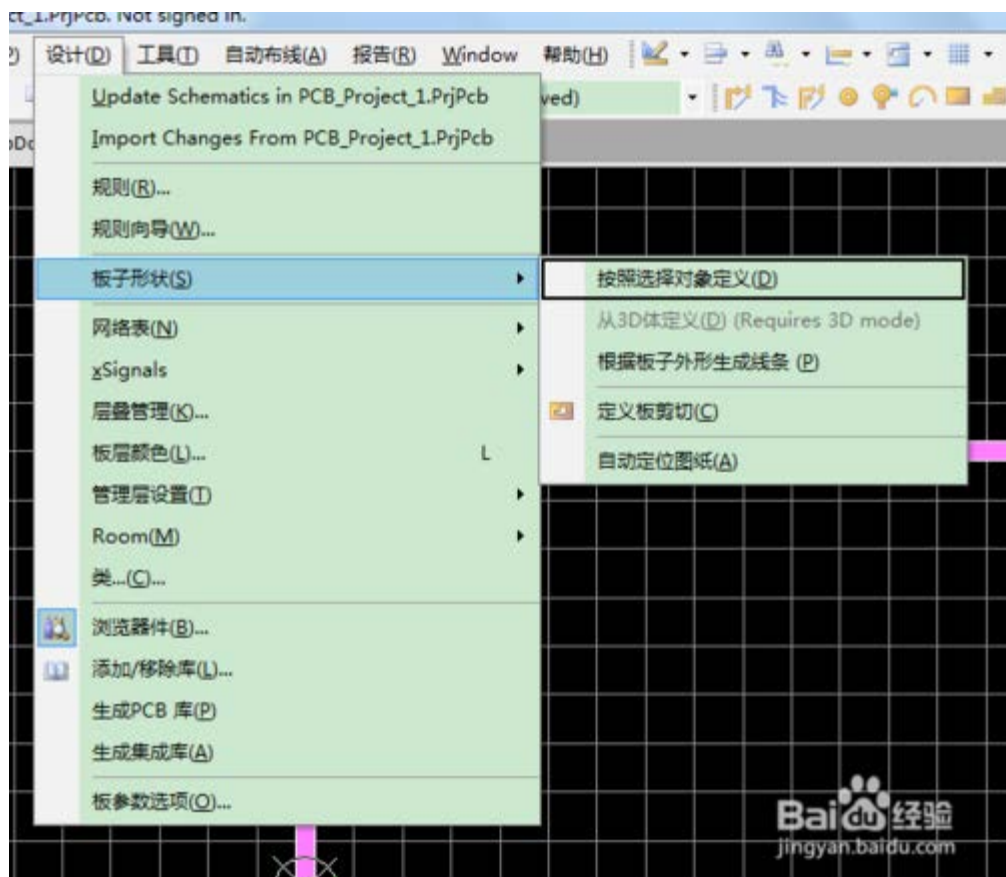
解决方法1：删除Room，即如下图左下角的红色部分。按Del键直接删。



6. 如何切边？

绘制边框，根据自己需要设计的板型来绘制。比如**放置—>走线**，绘制一个矩形边框。选中一条线，按 tab 键，选中整个边框。**设计—>板子形状—>按照选择对象定义**。

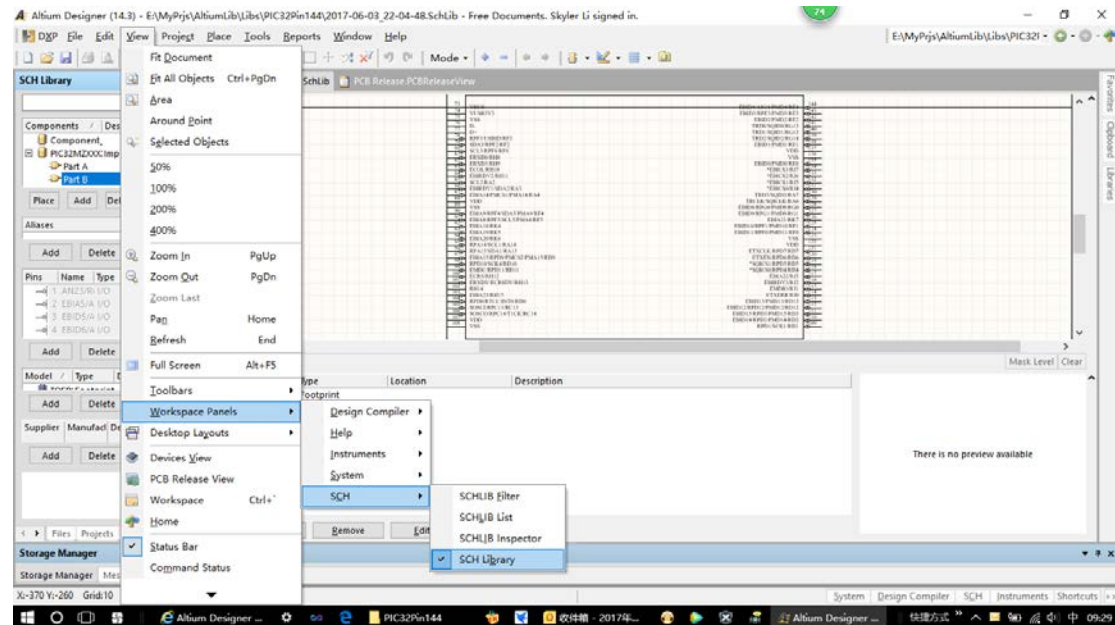




这样，边框就绘制完成了，至于其他形状的边框，绘制方法基本相同。可以按键盘功能键上的 3 键，查看 3D 效果。

7. 如何编辑原理图库？

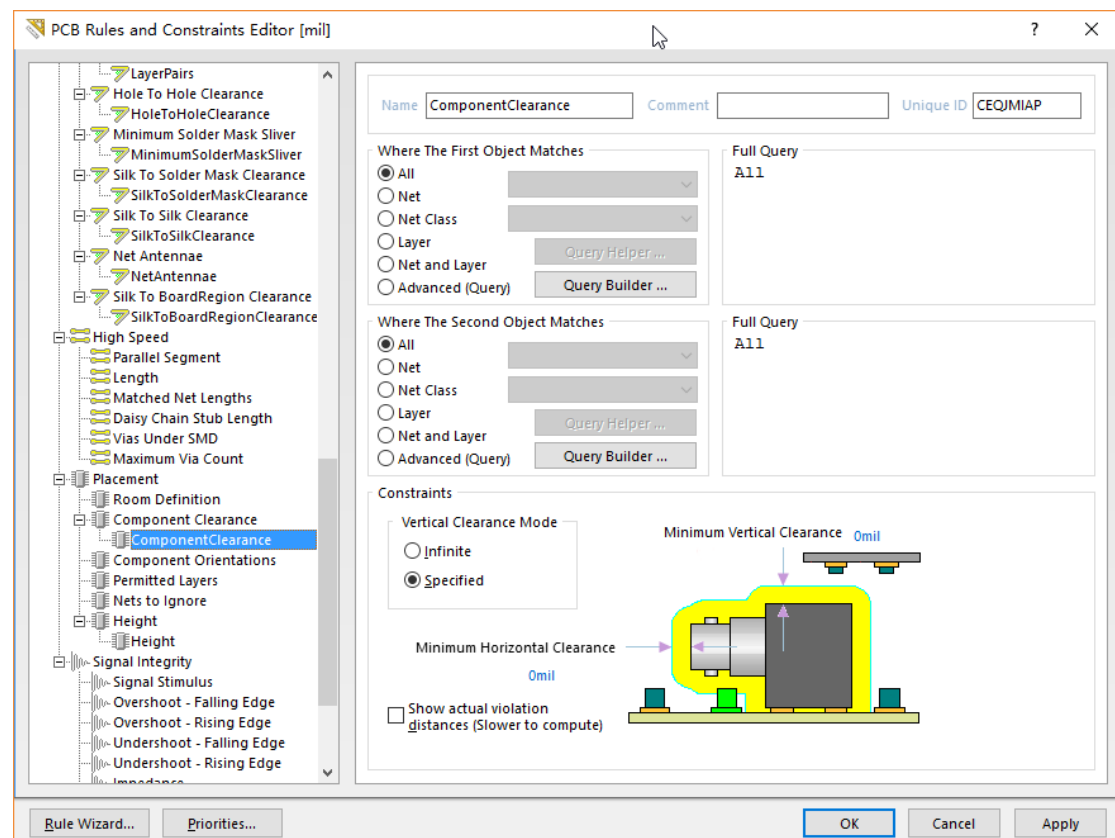
点击 View→Workspace Panels→SCH→SCH Library，然后可以选择 Part A 和 Part B 进行编辑，在实际应用中，笔者觉得原图的原理图太胖，把它变瘦了一些。



另外，在拷贝其他现成电路图的时候一定要留意是否有重名。

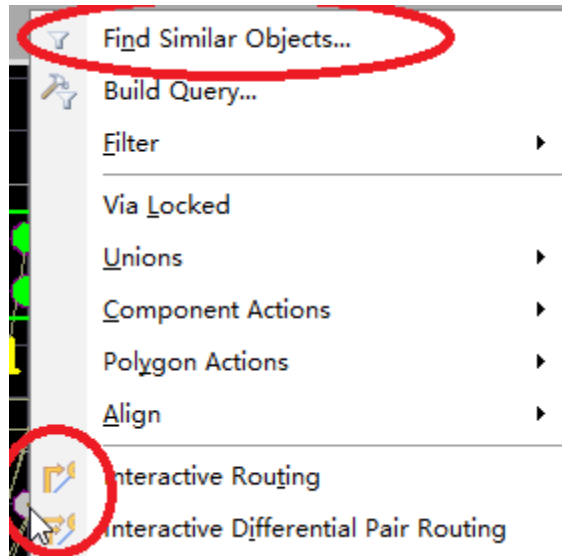
8. 如何处理元件离得太近错误：

在规则对话框中点击 Placement→ComponentClearance→ComponentClearance



9. 如何批量改过孔孔径？

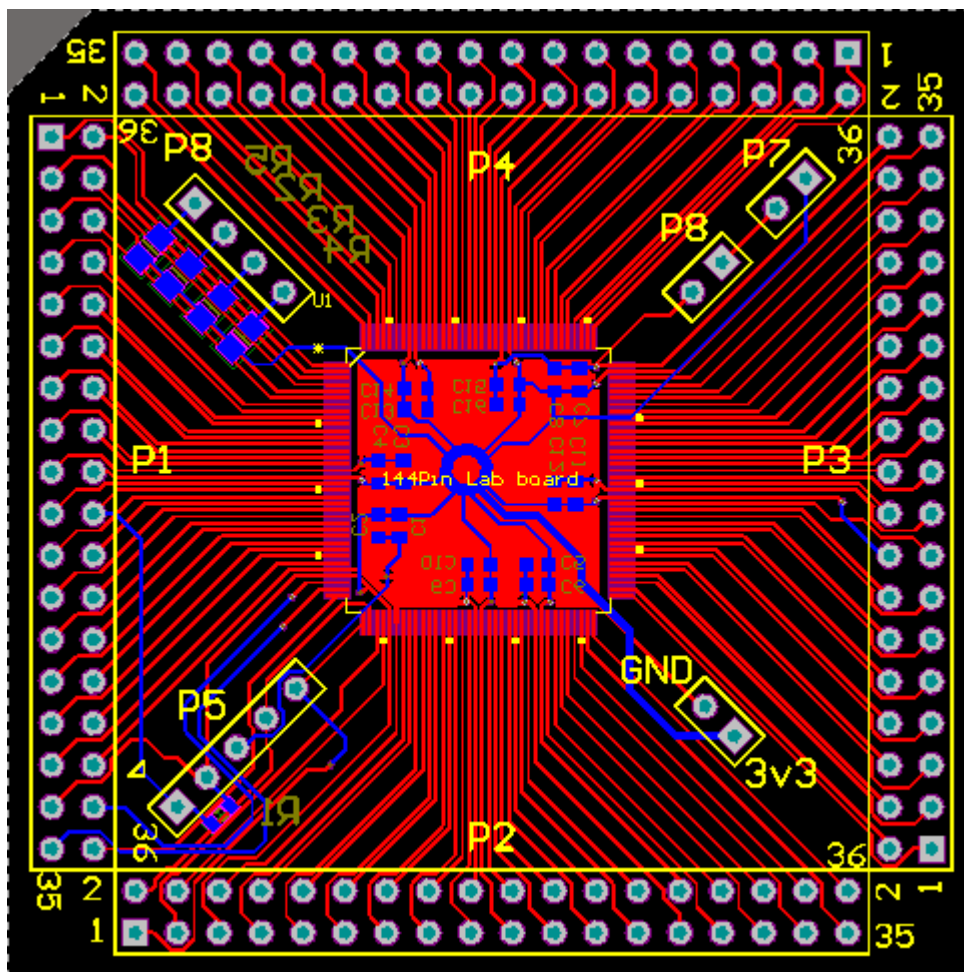
笔者在电路板的加工过程中被告知加工的孔径无法达到它们的工艺要求，要修改过孔的孔径，于是，笔者不得不修改所有过孔的孔径，方法如下：找一个过孔，点击右键然后选择 Find Similar Objects...，在弹出的对话框上点击 OK，如下图所示：



之后，在 PCB Inspector 的窗口中可以批处理的修改过孔的孔径，其他批处理元件的操作也是类同的。

总结：笔者的任务是制作一个 PCB 的实验板，以上流程只是管中窥豹，起到一个抛砖引玉的过程，真正的 PCB 设计与开发要复杂的多，也专业的的多，有兴趣的读者可以自行开发和设计需要的电路板。

笔者的电路板的最终形态如下：



下面的任务就是找电路板厂投板制作和焊接了，本文不赘述。

小任务：

1. 用 100Pin 的 PIC32 芯片制作一个类似的 DEMO 板子，要求引出所有的 PIN，能连接 ICSP，有 LED 电源指示灯，有上拉电阻的安装位，有电源输入口。

IO 端口功能介绍

一般我们拿到一个嵌入式任务，想到的第一个问题就是打通任督二脉，这任脉就是诸如配置字，IO 端口，ADC，中断，时钟等等归类为输入、输出的功能；这督脉就是诸如 UART，IIC，IIS，PWM，网络、USB 等等归类为通信的功能。有了这任督二脉做为基础，后边的逻辑和算法就相对顺利了。那么我们首先从 C 语言直接编程的角度一一介绍 PIC32 这些基本功能。后续我们会引入 Harmony 库的介绍。

IO 端口有哪些特性

PIC32 的端口的控制说难也难，你需要了解诸如漏极开路、阻抗匹配、端口驱动能力等等概念。说简单也简单，设置一个方向寄存器，即可对端口进行开关控制，我们试图从最简单的方式入手，由浅入深介绍每一个 PIC32 典型外设的特性和使用方法。

本节介绍的功能使用那块开发板进行实战？

设置端口的高低很简单，首先请允许我介绍一块开发板，PIC32 USB Starter Kit II



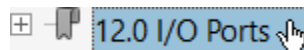
此开发板可以进行 IO 端口，中断，USB 主、从、OTG，时钟配置，在线调试，输入捕捉，输出比较，实时时钟，内存配置，Bootloader 等功能。该板卡的配置文件，购买信息等在如下链接：<http://www.microchip.com/DM320003-2>，其中 DM320003-2 是该板卡的型号信息。在此页面下有一个 Documents 选项卡，其中有一个 PIC32 USB Starter Kit II User's Guide 的文档下载链接，下载并且打开该文档，查阅到板卡的原理图，在 FIGURE A-10 我们找到关于 IO 的线路配置，发现三盏 LED 等分别由 RD0~RD2 控制，三个输入开关分别由 RD6，RD7，RD13 输入。

如何设置 IO 端口的输入和输出？

在正式对芯片进行编程和寄存器配置之前，我们有必要聊聊怎么去方便的下载和查阅芯片有关的资料，如上述章节所述，microchip 的网站有个很好的功能，那就是您可以在网站的域名后边加上您感兴趣的内容，如果内容和网站存储的格式相符，那么就会直接跳转到相对应的页面了。那么，我们首先需要下载主芯片 MCU 的数据手册。

<http://www.microchip.com/PIC32MX795F512L>

打开页面之后在页面的下方有一个 Documentation 的子项，打开该子项之后就会发现数据手册了。下载之。开启数据手册，在 Acrobat 右边有导航栏：



点击之后进入 IO 端口的章节，我们仔细查看该章节发现了一个问题，该章节并未对 IO 端口进行非常详尽的描述，只是做了一个概述，这是为啥？答案在章节的一段话，摘录如下：

Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 12. “I/O Ports”** (DS60001120) in the *“PIC32 Family Reference Manual”*, which is available from the Microchip web site (www.microchip.com/PIC32).

该段 Note 告诉了我们如果需要详细的 IO 端口的细节信息，您还需要下载另一个文档，也就是 Section 12.”I/O Ports” (DS60001120)才可以，那么这个文档在那里呢？还在我们刚才下载数据手册的地方，向下拉可以找到 Reference Manual 的子项，在该子项中有 Section 12.”I/O Ports” (DS60001120)文档。下载之。

如何读入和写出 IO 端口的状态

一般来讲，在默认状态下的端口我们直接设置它的方向寄存器便可。换句话说，我们通过查阅数据手册得知，针对 RD0 的输出功能，我们只需要设置方向寄存器 TRISDbits.TRISD0=0; 即可；针对 RD6 的输入功能我们只需要设置方向寄存器 TRISDbits.TRISD6=1;即可。运用 IO 端口笔者写了一段交通灯的例程，例程定名为 IO_Traffic_Light.rar 读者可以自行参考。

小任务：弄一块 DM320003-2 的板子，可以买，可以借，可以自己做，然后把 IO_Traffic_Light.rar 运行之，调整红绿灯变化的速度。

外设引脚选择（PPS）

一般来讲，每一个逻辑功能端口（例如 UART0_TX）对应的物理端口应该是唯一的（例如 29pin）；但是，有的时候我们并不希望如此，我们希望一个逻辑端口可以映射到不同的物理端口，比如，UART0_TX 可以有时候是 29pin，38pin，45pin 等等，如果你希望如此，则外设引脚选择功能可以帮助您实现这个功能。对于 PPS 这个功能，输入和输出正好是两个相反的概念

输入捕捉和输出比较

ADC 数模转换功能介绍

如何用轮询的方式读取 ADC

如何用中断的方式读取 ADC

ADC 与 DMA

UART 通信功能介绍

如何用简单的方式进行 UART 通信

如何用中断的方式进行 UART 通信

UART 与 DMA

IIC 通信功能介绍

如何用软件模拟的方式进行 IIC 通信

如何用中断的方式进行硬件 IIC 通信

编译器的基本功能

Harmony 库介绍

Harmony 库是 Microchip 公司针对于 PIC32 系列单片机做的一款多功能软件开发包，不

同一般的“库”的定义，Harmony 是采用了可以配置源代码生成的方式进行软件的二次开发，这就大大方便了用户对项目的开发和管理。在实际使用上，Harmony 库最好配合 Git 和 SVN 进行项目管理，另外也可以利用 Source Insight 进行代码浏览。

Harmony 库的下载和安装

如

如何将 Harmony 库的源码独立运行

CAN 通信功能介绍

如何建立一个简单的 CAN 通信链路

以太网通信功能介绍

建一个简单的以太网通信链路需要的资源

通过直连的方式和计算机进行通信

通过路由的方式和计算机进行通信

定时中断

如何写一个简单的定时中断的程序

如何切换不同的定时中断源

定时中断有那些常见的高级功能

看门狗计时器

如何写一个简单的看门狗程序

如何切换看门狗程序的计数源

Flash 读写功能介绍

如何读写内部 **Flash**

如何读写外部 **Flash**

如何利用自带的 IIS 搭建一个简单的声音输出系统

需要那些资源

如何搭建一个简单的 U 盘，SD 卡播放器实例

如何搭建一个简单的蓝牙播放器实例

如何搭建一个简单的 USB 耳机播放器实例

如何搭建一个简单的 USB 麦克风实例

如何利用 PIC32MZ 搭建一个蓝牙数据传输系统

需要那些资源

如何搭建一个简单的蓝牙数据传输系统实例

如何使用 **PIC32MZ** 的 **Bootloader** 引导程序

需要那些资源

如何搭建一个 **Bootloader** 引导程序的实例

如何进行图形界面的开发

进行图形界面的开发需要考虑那些问题

从哪里找到这些资源

常用的屏有那些，常用 **Touch** 控制器有那些

如何搭建一个图形界面的实例

如何换屏，如何换 **Touch**

如何利用 **Pic32MZ** 进行马达控制

如何搭建一个马达控制的实例

如何构建一个 RTOS 实例

如何搭建一个 **embos**、**freertos**、**openrtos**、**threadx**、**uCOSII**、**uCOSIII** 实例

以 **freeRTOS** 为例，如何创建、停止、挂起一个和多个任务？

以 **freeRTOS** 为例，如何进行进程间的通信

如何构建一个 USB 通信实例

如何搭建一个 **USB** 主通信实例

如何搭建一个 **USB** 从通信实例

如何搭建一个 **USB** 通过 **Hub** 通信的实例

写作目的：

1. 以实战为目的，为的是让工程技术人员快速上手
2. 尽量少些或者不写源代码，以流程，资源的详细介绍为主，告诉读者从那里去下载这些资源，怎么去调配和使用这些资源，而不是让读者照着书本敲代码。
3. 以具体的应用角度进行串联，我要实现这些应用需要那些资源，从那里找，找到之后怎么做。
4. 将嵌入式软件程序员的自我修养融入书中，软件自我修养是算法+数据结构和操作系统原理。
5. 只写干货没废话，长代码。