

# 项目3 简易跑表

## 3.1 目标、要求

### 3.1.1 目标

设计一款基于单片机的简易跑表（简称“简易跑表”），计时范围为0~99s，精度为10ms级，控制核心为AT89C51/52。本项目内容对以MCS-51为内核的单片机（如AT89S51/52、STC89C51/52RC）同样适用。

### 3.1.2 要求

- ① 为使简易跑表达到10ms级精度，要求使用单片机定时器/计数器、中断等资源。计时、计时暂停、计时清零等功能通过常用按键控制。
- ② 原理图设计成功后，要求仿真测出1s、10s、30s、60s、90s的走时误差。
- ③ 实际制作成功后，要求实际测出1s、10s、30s、60s、90s的走时误差。
- ④ 使用Proteus进行设计。整个项目包括从原理图设计、程序设计、仿真、调试、PCB设计，直至制作成实际产（作）品。

## 3.2 ISIS设计

### 3.2.1 原理图设计

#### 1. 原理图

在ISIS设计平台中设计原理图。图3-1（a）为简易跑表的原理图、仿真片段；其所用元器件的列表如图3-1（b）所示，左侧为元器件的英文名，右侧为对应的中文解释。晶振的频率为12MHz。

本项目采用系统默认图纸大小（A4，长×宽为10in×7in）进行设计。待设计完成后，若要改变图纸大小，可单击菜单System→Set Sheet Size，调整图纸到希望的大小。

本项目取名为ZXM03.DSN。

#### 2. 选取元器件

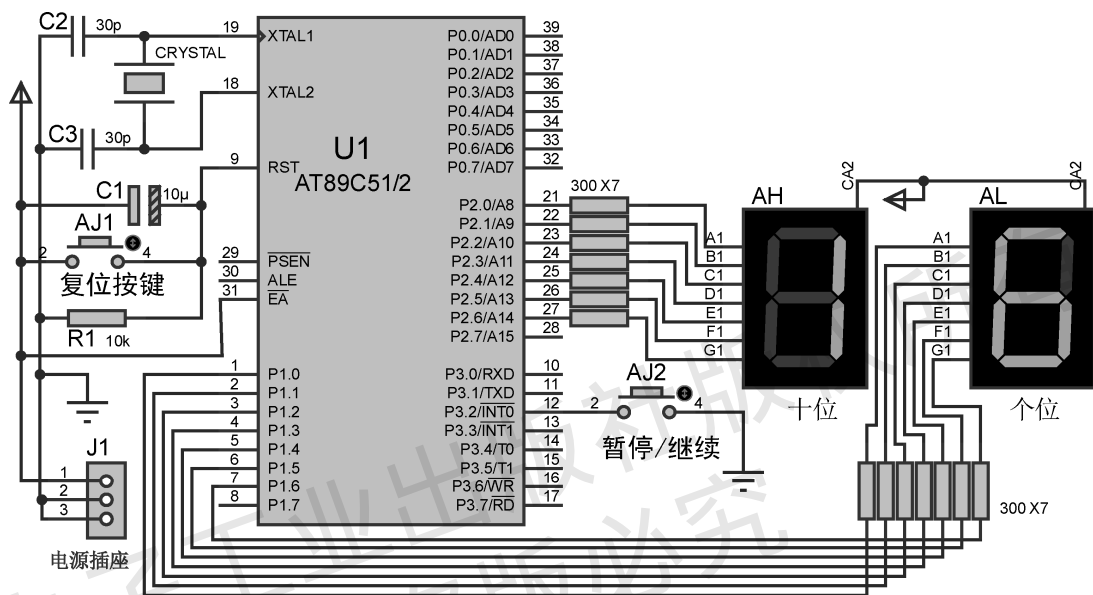
根据图3-1（b），将所需元器件从库中一一选入ISIS的对象选择器中。选入时应特别注意以下事项。

① 参与仿真的元器件必须是有仿真模型的元器件。本项目中仅电源插座J1是无仿真模型的元器件。若元器件无仿真模型，则要在仿真之前，打开其编辑属性对话框，参考图2-2（b），

将其设置为不参与仿真，否则不能进行仿真。

② 参与 PCB 设计的元器件应尽量有合适的封装（提示：虽然有的元器件有封装，但可能不合适）。图 3-1 (b) 显示电源插座 J1 有封装，封装名为 SIL-156-03，但不适合实际的电源插座。2.5.1 节制作了名为 Z3PIN-POW 的封装，并已存入封装库中，所以只需选用封装是 Z3PIN-POW 的电源插座即可。“按键”也无封装，参考 3.5.1 节制作名为 Z4PIN-BUT 的封装，并存入封装库中，再将封装指定给“按键”即可。指定封装的操作参考 2.3.1 节。

③ 参与 PCB 设计的元器件一定要有标注。



(a) 原理图、仿真片段

7SEG-COM-AN-GRN	数码管
AT89C51	单片机
BUTTON	按键
CAP	电容
CAP-ELEC	电解电容
CRYSTAL	晶振 (12MHz)
PULLDOWN	下拉电阻R1
RESISTOR	电阻
SIL-156-03	电源插座

(b) 元器件列表

图 3-1 简易跑表的原理图、仿真片段及元器件列表（实际制作时下拉电阻 R1 应为 10kΩ 的实际电阻）

### 3. 布置元器件、电气连线

将选取的元器件从对象选择器调入编辑区中，将电源、地等终端也调入编辑区中，合理安排它们的位置，再根据原理图进行电气连线。

#### 3.2.2 程序设计

##### 1. 汇编语言程序设计

Proteus ISIS 菜单栏中的 Source（源代码）项提供了源程序编辑器（SRCEDIT）、汇编代码生成器（ASEM51）等工具，支持 CALL、JMP 指令，为程序设计提供了方便。这里提供下

列程序供参考，并对重要指令进行了注释，文件名取为 ZXM03.ASM。

```

;*****
;    简易跑表程序          *
;    ZXM03.ASM            *
;    晶振 12MHz          *
;*****
    ORG    00H
    SJMP   STAR
    ORG    03H
    AJMP   INT0S          ;转 INT0 中断服务程序
    ORG    1BH
    SJMP   T1S           ;转 T1 中断服务程序
    ORG    30H
;*****
STAR:  MOV    R2,#0        ;计时初值
        MOV    R4,#20     ;定时中断溢出计数器 R4 初值为 20
        MOV    IE,#89H    ;开 T1、INT0 中断
        SETB   IT0
        MOV    TMOD,#10H  ;T1 方式 1
        MOV    TH1,#3CH   ;定时初值
        MOV    TL1,#0B0H  ;定时初值
        SETB   TR1        ;启动 T1
        ACALL  DIS        ;调用显示子程序
        SJMP   $
;*****
;    定时器 1 中断服务程序      *
;*****
T1S:   MOV    TH1,#3CH    ;中断程序
        MOV    TL1,#0B0H  ;重装初值
        DJNZ   R4,T1S1    ;定时 1s 到否
        MOV    R4,#20     ;☆到 1s, 重置 R4=20
        INC    R2
        CJNE   R2,#99,T1S0 ;☆#10,#30,#60,#90
        CLR    TR1        ;计时满 99, 关定时器
T1S0:  ACALL  DIS        ;调显示
T1S1:  RETI             ;中断返回
;*****
;    外中断 0 服务程序          *
;*****
INT0S: CPL    F0
        JNB    F0,INT0A
        CLR    TR1
        RETI
INT0A: SETB   TR1
        RETI
;*****
;    十六进制转为 BCD 字程序      *

```

```

;*****
DIS:   MOV     A,R2           ;单字节十六进制
      MOV     B,#10         ;数转为十进制数
      DIV     AB
      ACALL   SEG7
      MOV     P2,A          ;显示十位
      MOV     A,B
      ACALL   SEG7
      MOV     P1,A          ;显示个位
      RET     ;子程序返回
;*****
; 查表得数码管显示码子程序 *
;*****
SEG7:  INC     A             ; (A) ← (A) + 1
      MOVC   A,@A+PC        ;取显示断段
      RET
      DB 0C0H,0F9H,0A4H,0B0H ;0~3 的共阳型显示码
      DB 99H,92H,82H,0F8H   ;4~7 的共阳型显示码
      DB 80H,90H,88H,83H   ;8~B 的共阳型显示码
      DB 0C6H,0A1H,86H,8EH  ;C~F 的共阳型显示码
;*****
      END

```

## 2. C51 程序设计

在 Keil 中进行 C51 程序设计。这里提供下列程序供参考，并对重要指令进行注释，文件名取为 ZXM03.C。

```

// 简易跑表程序
// ZXM03.C,晶振 12MHz
#include <reg51.h>
#define uchar unsigned char
#define Fosc 12000000
#define t50ms (65536-Fosc*50/12000)
uchar data chuzhi=0;
uchar data a=20; //定时 1s , 20*50ms
uchar code tab[]={0xC0,0xF9,0xA4,0xB0, 0x99,0x92,0x82,
0xF8,0x80,0x90,0x88,0x83,0xC6,0xA1,0x86,0x8E} ;
void dis(uchar cct)
{ P1=tab[cct/10] ;
  P2=tab[cct%10] ;
}
void int0f(void) interrupt 0 using 1
{ F0=!F0;
  if(F0==0)
    TR1=1;
  else
    TR1=0;
}

```

```

void t1f(void) interrupt 3 using 0
{
    TH1=t50ms/256;
    TL1=t50ms%256;
    a=a-1;
    if(a==0)
    {
        a=20;
        chuzhi=chuzhi+1;
        if(chuzhi==100)
            EA=0;
        else
            dis(chuzhi);
    }
}

void main()
{
    TMOD=0X10;
    EA=1;
    ET1=1; EX0=1;
    TR1=1; IT0=1;
    TH1=t50ms/256;
    TL1=t50ms%256;
    dis(chuzhi);
    for(;;);
}

```

### 3. 程序汇编（编译）、下载

#### (1) 汇编程序


在 ISIS 中，完成程序编辑、汇编、生成目标代码文件 ZXM02.HEX 等操作。打开单片机编辑属性对话框，在 Program File 栏中填入（下载）目标代码文件 ZXM03.HEX，再在 Clock Frequency 栏中设置时钟频率为 12MHz。

若采用高版本 Proteus，只要源程序汇编（Build All）通过，就会自动将最后的目标代码文件填入单片机编辑属性对话框中。若采用较低版本 Proteus 或采用其他软件（如 Keil）汇编（编译）生成目标代码文件（\*.HEX 文件），则需用户将最后的目标代码文件填入单片机编辑属性对话框中。

#### (2) C51 程序

在 Keil 中建立名为 ZXM03.C 的文件，编辑程序并编译生成目标代码文件 ZXM03.HEX。打开单片机编辑属性对话框，在 Program File 栏中填入（下载）目标代码文件 ZXM03.HEX，再在 Clock Frequency 栏中设置时钟频率为 12MHz。

### 3.2.3 仿真、调试

上述各步操作正确完成后，单击仿真工具按钮中的进行仿真。仿真片段如图 3-1（a）所示。按要求操作各按键，可以看到计时、暂停、清零等时间实时显示情况。

若出现异常，可先检查电路设计，排除、修改可能的电路设计错误，直到正确；再检查程序设计是否有错误，可进入仿真调试状态，进行设置断点、查错、修改等操作，直到达到

项目设计目标。

运行正确后，进入调试状态，测试出 1s、10s、30s、60s、90s 的走时时间及走时误差（参考 3.5.2 节）。

## 3.3 ARES 设计


### 3.3.1 设计准备

#### 1. 查看元器件、封装等信息

在 ISIS 中仿真成功后，要进行下列操作。

① 检查原理图中哪些元器件参与 PCB 设计，哪些元器件不参与 PCB 设计。本项目有 25 个元器件，都参与 PCB 设计。

检查参与 PCB 设计的元器件是否都有标注。如果没有，则补上。将复位、暂停/继续键分别标注为 AJ1、AJ2。将十位、个位数码管分别标注为 AH、AL。

② 单击 ISIS 工具栏中的设计浏览器按钮 ，打开如图 3-2 (a) 所示设计浏览器，查看元器件、封装等信息。本项目有三个元器件 (AJ1、AJ2、J1) 标有无封装或未指定封装的红色 missing 标识 (见图 3-2 (a))，即两个按键和一个电源插座无封装或未指定封装，应在进行 ARES 设计前解决好它们的封装问题，如图 3-2 (b) 所示。

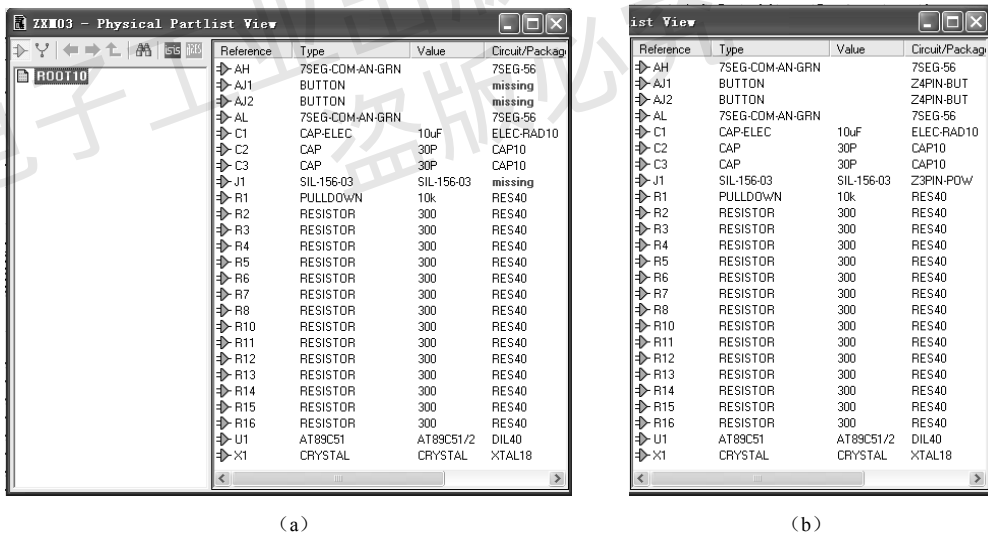


图 3-2 打开设计浏览器，查看元器件、封装等信息

#### 2. 指定封装

##### (1) 指定数码管的封装

因在 ISIS 中选取元器件时，已选用有封装 7SEG-56 的数码管 7SEG-COM-AN-GRN，所以只需按 2.3.1 节、2.5.2 节指定并分配引脚即可。

(2) 电源插座 J1 的封装在 2.5.1 节中已制作且存盘，名为 Z3PIN-POW。参考 2.3.1 节、


2.5.2 节指定封装。

### (3) 制作按键的封装

按键产品多种多样, 现有封装库中没有符合 AJ1、AJ2 要求的封装, 要自己设计封装。请按 3.5.1 节制作、存盘, 再指定封装。

## 3.3.2 导出网表, 进入 ARES

单击 ISIS 中的“生成网表并切换到 ARES”工具按钮 (或单击菜单 Tools→Netlist to ARES), 进入 ARES 设计窗口 (见图 1-5)。

在 ARES 窗口中, 单击按钮, 保存 PCB 文件, 文件名自动为“ZXM03.LYT”, 表示该文件是简易跑表的 PCB 文件; 接着参考图 1-28 设置板界, 可先设置得稍大些, 如宽为 82mm, 高为 100mm。


## 3.3.3 布局

一般, 先手动布局, 再自动布局, 然后手动调整到满意为止。


### (1) 手动布局

将占用板面积较大的元器件 (如 U1、AH、AL 等) 及需要特定位置的元器件 (如 AJ1、AJ2 等) 布好。

### (2) 自动布局

单击菜单 Tools→ Auto Placer..., 打开自动布局对话框 (见图 2-6)。本项目采用默认设置, 单击 OK 按钮, 尚未布局元器件所对应的封装被自动转移到编辑区板界内进行自动布局。

### (3) 调整布局

手动布局、自动布局后, 若有不满意之处, 可再手动布局进行调整。单击工具按钮, 右击选中需要调整的元器件封装, 在弹出的菜单中选择相关命令, 进行移动、转向等调整操作。图 3-3 所示为经手动布局、自动布局、手动调整后的布局情况。

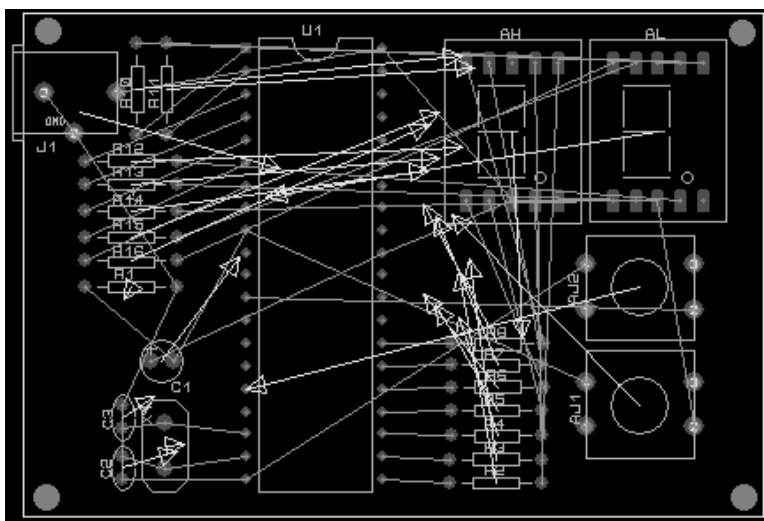


图 3-3 经手动布局、自动布局、手动调整后的布局情况

### 3.3.4 布线

布线有手动布线、自动布线、手动调整。一般都采用三者相结合的布线方式。

本项目的布线与 2.3.4 节的相同。只改变 Net Classes 选项卡中电源线宽、信号线宽的设置，分别为 T40、T30，也采用双面板设计，标注文字、绘制图符、调整板界尺寸操作也相同。

完成布线后的简易跑表 PCB 版图如图 3-4 所示。

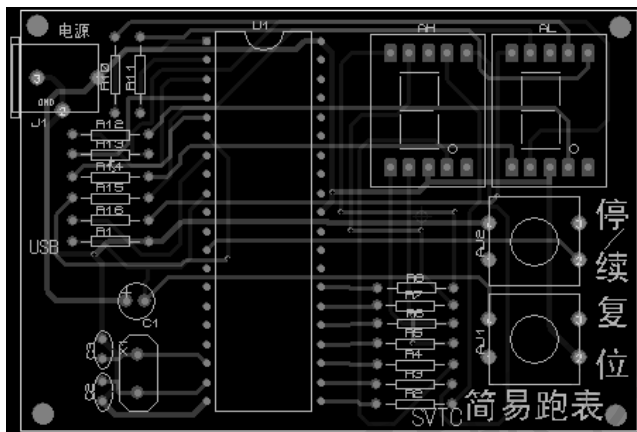


图 3-4 完成布线后的简易跑表 PCB 版图

### 3.3.5 3D 预览

确认 PCB 版图无误后，单击菜单 Output→3D Visualization，可进行 PCB 的 3D 预览，如图 3-5 所示。

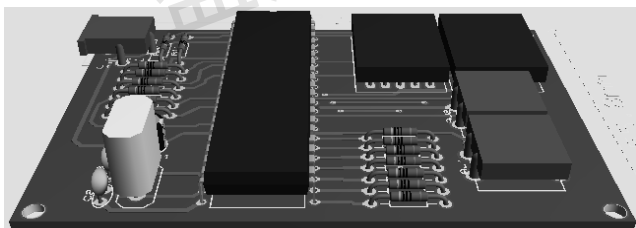


图 3-5 PCB 的 3D 预览

### 3.3.6 图纸输出

PCB 设计、3D 预览满意后，可输出图纸（参考 2.3.6 节）。

本项目输出可送制板厂制板的压缩文件 ZXM03（“Gerber”格式），如图 3-6 所示。

名称	类型
ZXM03 - CAD/CAM	WinRAR ZIP 压缩
ZXM03	PCB Layout
Backup Of ZXM03.LBK	LBK 文件

图 3-6 压缩文件 ZXM03（“Gerber”格式）



## 3.4 实际制作

### 1. 硬件制作

将 PCB 压缩文件送制板厂，加工出 PCB，如图 3-7 所示。

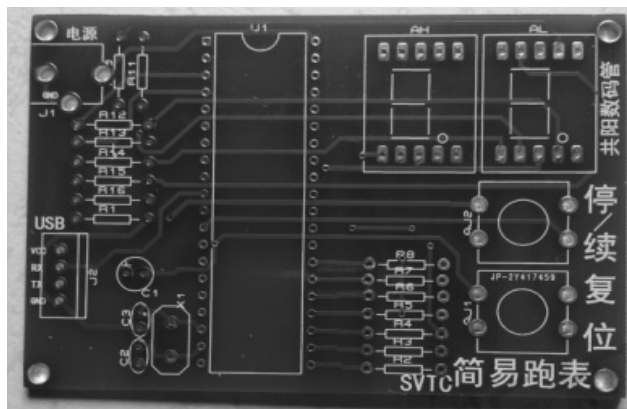


图 3-7 简易跑表的 PCB

仔细检查电路板，是否有缺线、断线、搭连等现象；必要时，要使用万用表、示波器等工具检查，直至无误。

正确选择、使用元器件，细心安装焊接。

### 2. 下载目标代码文件

可使用编程器将项目相应目标代码文件（ZXM03.HEX）下载到单片机 AT89C51/52、AT89S51/52 中，也可使用 ISP 下载器将项目相应目标代码文件（ZXM03.HEX）下载到单片机 AT89S51/52、STC89C51/52RC 中。

### 3. 实际运行

上述操作完成无误，通电运行。分别操作“复位”（清零重开始）按键、“暂停/继续”按键，完成计时、暂停、继续、清零等功能测试。

若出现异常，可先检查电路设计，排除、修改可能的电路设计错误，直到正确；再检查程序设计是否有错误，可进入仿真调试状态，进行设置断点、查错、修改等操作，直到达到项目设计目标。

图 3-8 是简易跑表运行时的照片。

设计测时方案，测出简易跑表 1s、10s、20s、40s、80s 的走时时间及走时误差。

**提示 1：**也可根据 PCB 版图在实验 PCB 上制作本项目。

**提示 2：**PCB 左侧的 USB 接口是为了使用计算机来供电、通信而后加的，学生设计时可忽略。

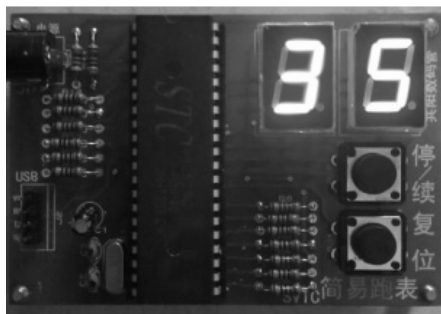



图 3-8 简易跑表运行时的照片

## 3.5 技术链接

### 3.5.1 制作四脚按键的封装并存入封装库

四脚按键 (BUTTON) 实物照片如图 3-9 (a) 所示。设其封装取名为 Z4PIN-BUT, 其外形、尺寸如图 3-9 (b) 所示。

#### 1. 放置焊盘

① 单击圆形穿孔焊盘模式按钮 ，在对象选择器中就会列出各种圆形穿孔焊盘，单击其中的 C-100-60，如图 3-10 所示。

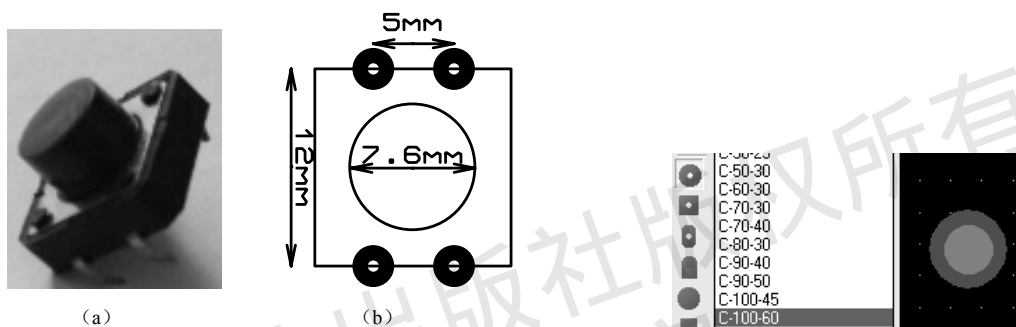



图 3-9 四脚按键 (BUTTON) 实物照片及其封装外形、尺寸

图 3-10 选择焊盘类型并放置

② 单击工具栏中  按钮，将坐标单位设置为公制。在要放置焊盘处按键盘字母“O”键，单击放置焊盘，焊盘的中心便是伪原点 (0,0)，如图 3-11 (a) 所示，

③ 右击焊盘选择 Replicate 命令，按照图 3-11 (b) 进行复制参数设置。单击 OK 按钮，结果如图 3-11 (c) 所示。在空白处单击退出复制后的选中状态。

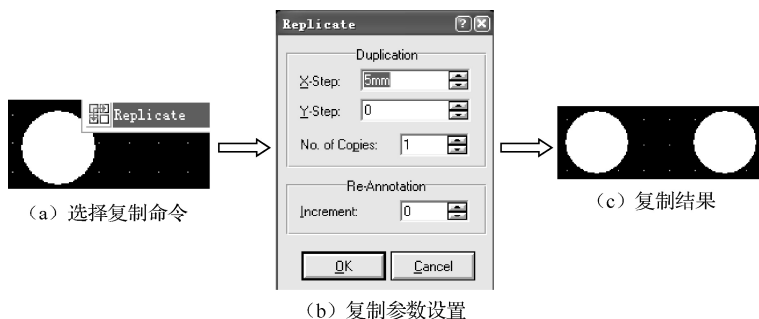
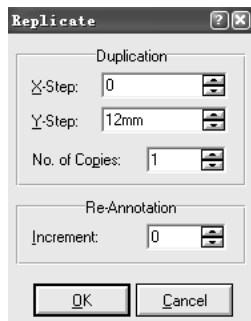


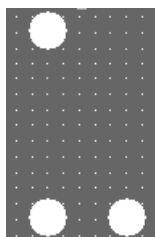
图 3-11 水平复制焊盘操作

④ 右击第一个放置的焊盘，选择 Replicate 命令，按照图 3-12 (a) 进行复制参数设置。单击 OK 按钮，结果如图 3-12 (b) 所示。在空白处单击退出复制后的选中状态。

⑤ 右击第一个放置的焊盘，选择 Replicate 命令，按照图 3-13 (a) 进行复制参数设置。单击 OK 按钮，在空白处单击退出复制后的选中状态，结果如图 3-13 (b) 所示。

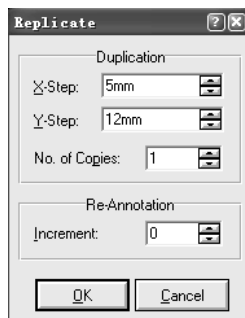


(a)

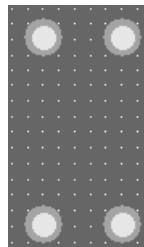


(b)

图 3-12 垂直复制焊盘




(a)

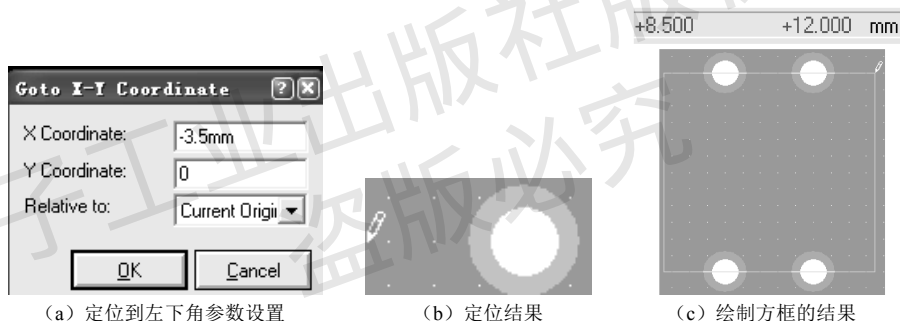


(b)

图 3-13 非正交复制焊盘

## 2. 放置元器件轮廓框

① 单击操作模式按钮, 进入放置 2D 方框模式。单击菜单 View→Goto XY, 如图 3-14 (a) 所示, 设置相对当前原点的定位坐标 (-3.5mm,0), 单击 OK 按钮, 光标定位结果如图 3-14 (b) 所示。单击, 开始绘制方框, 移动光标, 注视窗口右下方状态栏坐标变化, 当坐标为 (8.5mm,12mm) 时再单击, 如图 3-14 (c) 所示完成方框绘制。




(a) 定位到左下角参数设置

(b) 定位结果

(c) 绘制方框的结果

图 3-14 绘制封装的 2D 方框

② 单击操作模式按钮, 进入 2D 画圆模式。单击菜单 View→Goto XY, 如图 3-15 (a) 所示, 设置相对当前原点的定位坐标 (2.5mm,6mm), 单击 OK 按钮, 光标定位在圆心。单击, 开始画圆, 移动光标, 注视窗口右下方状态栏半径变化为 3.8mm (见图 3-15 (b)) 时, 再单击, 结果如图 3-15 (c) 所示, 完成画圆。



(a) 定位圆心

(b) 半径

(c) 画圆的结果

图 3-15 绘制封装的 2D 圆

### 3. 对焊盘编号

单击菜单 Tools→Auto Name Generator, 打开如图 3-16 (a) 所示的自动命名生成器。因焊盘编号只有数字, 无需前缀, 所以 String 栏空白, Count 从 1 开始, 单击 OK 按钮, 单击左上角的焊盘, 设置焊盘编号为 1, 按逆时针单击其他焊盘, 编号结果如图 3-16 (b) 所示。

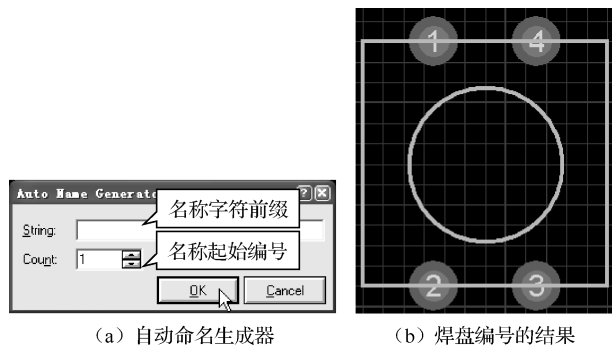



图 3-16 对焊盘编号

### 4. 放置元器件标注

单击 2D 模式栏的 , 再单击对象选择器中的 REFERENCE, 如图 3-17 所示, REF 放置在封装的上方。

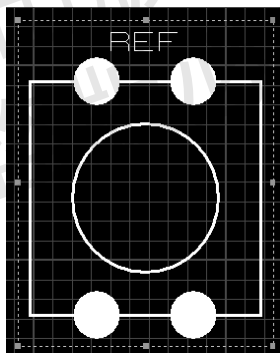



图 3-17 放置 REF 的结果

### 5. 封装命名并存入封装库

全选图 3-17 (右击按住拉出包围全部设计的框), 单击 ARES 菜单中的 , 打开如图 3-18 (a) 所示的 Make Package 对话框, 将封装命名为 Z4PIN-BUT 并存入封装库, 还可进行 3D 预览 (见图 3-18 (b))。最后, 单击 OK 按钮完成。

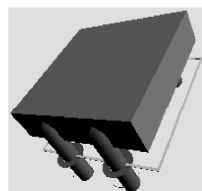
实际按键的 1、2 脚是连通的, 3、4 脚也是连通的。电气连线及 ARES 设计时要注意这点。

### 6. 封装指定

制作的按键封装 Z4PIN-BUT 已入封装库中, 如何在 ISIS 中将三脚插座的现有封装指定 (修改) 为 Z4PIN-BUT 呢? 打开三脚插座的属性对话框, 看到 PCB Package 栏为 SIL-156-03, 将其直接更改为 Z4PIN-BUT 即可。



(a) 创建封装：设置封装名为Z4PIN-BUT并存入封装库



(b) 3D预览

图 3-18 封装命名、存库

## 7. 原点说明

未设置原点的封装自动以第一个放置的焊盘中心为原点，在本项目中即为焊盘 2 的中心，即放置本封装时光标点在焊盘 2 的中心。

### 3.5.2 测时及其误差

#### 1. 仿真调试测时

本项目要求时间精度为毫秒级，设计中必须使用单片机资源，如定时/计数器及中断。

本项目要求测出 1s、10s 等的走时时间及走时误差。Proteus 提供了仿真调试测时的精确方法。

举例：测 1s 及其误差，可在汇编语言程序指令“MOV R4, #20 ;☆到 1s, 重置 R4=20”行设置断点，或在 C 语言程序中 t1f()函数的第 6 行“chuzhi=chuzhi+1;”设置断点。运行时，两次断点间运行时间会在 ISIS 下方状态栏中显示。

[U1] Digital breakpoint at time 2.0003s (1.0002s elapsed) · 1

即走时时间为 1.0002s，走时误差为 0.0002s。

类似的，也可仿真测量 10s、30s 等的走时时间及走时误差。

#### 2. 简易跑表实物测时

建议学生在教师指导下视设备情况、设计方案及测试电路，测出 1s、10s、30s 等的走时时间及走时误差。