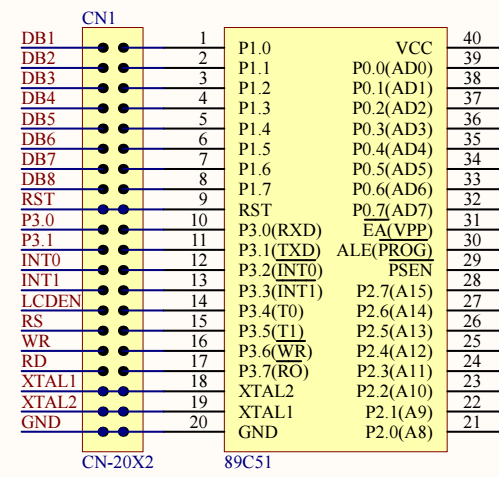
# 实验一 IO口应用

**------用独立按键模拟电子琴键盘发声**

1. **实验目的**
2. 熟悉Keil软件的使用；
3. 熟悉独立按键状态识别技巧；
4. 熟悉蜂鸣器驱动电路，及蜂鸣器发声频率控制方法。
5. **实验内容及原理**
6. 独立按键

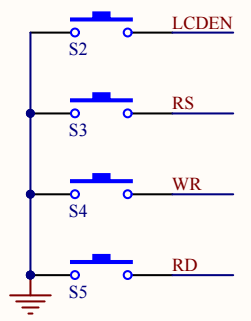
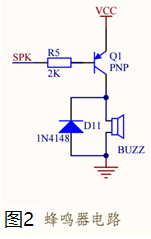


图1 独立按键硬件结构

开发板四个独立按键与单片机STP89C52连接关系如图1。可见，当S2按下时，P3.4引脚为低电平，否则为高电平；S3按下时P3.5为低电平；S4按下时P3.6为低电平；S5按下时P3.7为低电平。

1. 蜂鸣器

开发板电路使用了单片机的I/0口P2.3来控制蜂鸣器。但并没有直接用P2.3控制蜂鸣器，而是通过PNP三极管间接控制。这是因为单片机的I/0口的输入输出电流有限，不能满足蜂鸣器鸣响所需要的电流。而该三极管最大可以提供1A以上的电流，足以驱动蜂鸣器，所以使用P2.3控制PNP三极管的导通和截止，从而达到控制蜂鸣器的目的。

向P2.3写出逻辑“1”时，P2.3输出+5V，三极管的基极电流为0，三极管处于截止状态，电源+5V不能加到蜂鸣器的正极，蜂鸣器不鸣响；向P2.3写出逻辑“0”时，P2.3输出0V，三极管导通，图2中限流电阻R5取合适的值，可以使三极管处于饱和导通状态，电源+5V通过三极管的发射极和集电极加到蜂鸣器的正极，有电流流过蜂鸣器的正极和负极，有源蜂鸣器就开始鸣响。

1. 电子琴键盘音阶频率

电子琴部分音阶与频率对应关系如表1所示。当单片机P2.3输出相应频率的方波给蜂鸣器时，蜂鸣器便发出对应频率的音阶。

表一 电子琴键盘音阶频率对照表（频率单位：Hz）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 音阶 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 频率 | 261.632 | 293.672 | 329.636 | 349.237 | 392.005 | 440.010 | 493.895 |

1. **实验步骤**
2. 使用Keil编译软件，建立文件夹用来保存当前工程，文件夹以班级学号姓名来命名；新建C51工程文件\*.uvproj并保存在当前文件夹中；建立新的\*.c文件保存并将其加入到当前工程中。单片机芯片选择AT89C52。
3. 在\*.c的文件中编写程序，实现以下功能：

按下独立按键S2时，使蜂鸣器发出“do”的音；按下S3发出“re”的音；按下S4发出“mi”的音，按下S5发出“fa”的音。

独立按键优先级排列顺序依次为：S2🡪S3🡪S4🡪S5

1. 编译链接工程文件，生成\*.Hex文件；
2. 设置晶振频率为11.0125MHz。调试程序，用Keil软件中逻辑分析仪验证频率是否正确。
3. 下载\*.hex文件到单片机开发板，依次按下S2~S4，验证蜂鸣器发声频率正确。
4. **实验源代码**

/\*  
encoding:utf-8  
name:89c51四音电子琴  
author:https://www.mrskye.cn  
\*/  
#include<reg51.h>  
#include<intrins.h>  
sbit buzz = P2^3;  
unsigned char m; //持续时间变量  
void delay(unsigned int i) //消抖延迟函数  
{  
 unsigned int j  
 unsigned char k;  
 for(j=i;j>0;j--)  
 {  
 for(k=125;k>0;k--);  
 }  
}  
void delay130us(void) //do:误差 -0.055500567537us  
{  
 unsigned char a,b;  
 for(b=13;b>0;b--)  
 for(a=3;a>0;a--);  
}  
void delay146us(void) //re:误差 -0.820108967083us  
{  
 unsigned char a,b;  
 for(b=1;b>0;b--)  
 for(a=64;a>0;a--);  
}  
void delay164us(void) //mi:误差 -0.277704880817us  
{  
 unsigned char a;  
 for(a=74;a>0;a--);  
}  
  
void delay174us(void) //fa:误差 -0.271167423383us  
{  
 unsigned char a,b;  
 for(b=1;b>0;b--)  
 for(a=77;a>0;a--);  
}  
void main() //switch 改造升级  
{  
 while(1)  
 {  
 if(P3 != 0xff)  
 {  
 delay(30); //消抖  
 if(P3 != 0xff)  
 {  
 switch(P3)  
 {  
 case 0xef:  
 {  
 while(P3==0xef)  
 {  
 for(m=3823;m>0;m--)//0.130816ms\*3823=0.5s  
 {  
 buzz =~buzz;  
 delay146us();  
 }  
 }  
 break;  
 }  
 case 0xdf:  
 {  
 while(P3==0xdf)  
 {  
 for(m=3405;m>0;m--)//0.146836ms\*3405=0.5s  
 {  
 buzz =~buzz;  
 delay130us();  
 }  
 }  
 break;  
 }  
 case 0xbf:  
 {  
 while(P3==0xbf)  
 {  
 for(m=3037;m>0;m--)//0.164818ms\*3037=0.5s  
 {  
 buzz =~buzz;  
 delay164us();  
 }  
 }  
 break;  
 }  
 case 0x7f:  
 {  
 while(P3==0x7f)  
 {  
 for(m=2863;m>0;m--)//0.1746185ms\*2863=0.5s  
 {  
 buzz =~buzz;  
 delay174us();  
 }  
 }  
 break;  
 }  
 }  
 }  
 }  
 }  
}

**五、实验总结**

1. 蜂鸣器要使用无源的，对应keil 中元件可以为 speaker、sounder ，且需要为 ACTIVE 才能仿真。有源蜂鸣器需要主题调节启动电压（通常为 12V 无法达到）。
2. 频率与时间周期换算通过在线网站解决：<http://www.gowebtool.com/unit-conversion/frequency/>
3. 延迟执行函数使用工具解决：单片机小精灵v1.3
4. 目前代码实现方案比较 low ，参考网络他人作品发现可以用 switch 判断整个 P3 口的值，确定输出音效。~~有时间在升级吧~~~ 升级改进完成。