

手把手教你用51单片机DIY音乐频谱显示

文/张孝虎

以前，每当看到家里的音箱功放上的几排小灯，随着播放的音乐如波浪般跳跃，或者在电脑上打开千千静听这个音乐播放软件时，看到那动感的频谱跟随音乐节奏优美的舞动着时，不禁思绪万千，要是自己某天能亲手用普通的单片机DIY这么一个东东，那将是多么有成就感的事情，至少对我们电子爱好者来说，这是许多人曾经梦想过的。伴随音箱里传来的美妙音乐，原本只能“听”的音乐，现在却还能“看”，给人带来视觉上的炫酷享受。

本文将带领你动手探究其中的奥秘，你会发现她不再神秘，只要跟着我一步一步动手做，你一定可以成功做出来。让我们给音乐赋予一个完美的视听极限，你还在等什么？马上动手吧！

一、理论篇

下面我们将讲解与制作有关的基础理论知识，我将力求使音乐频谱显示的原理通俗易懂。当然，你如果实在等不及了，或者暂时没时间弄明白也可以快速跳过本节，在制作篇直接跟我动手开始做。

(一) 理论基础概述

正在阅读本文的你可能会问，制作这么神奇的东西，她到底是什么原理，需要用到些什么理论吗？我能明白其中的原理吗？

1) 你问：什么是频谱？什么是频谱显示？

我答：说到“频谱”这个词，它好像是一种很神秘的东西，究竟什么是“频谱”？这便要从波形本身的来源说起。我们知道，一切声音都是由振动产生的。声音之所以千变万化各不相同，是因为它们的振动各不相同。看看琵琶，吉他或者其他弦类乐器，可以发现它的每一根琴弦的直径都是不一样的。琴弦越细，音调也就越高。反之则越低。显然粗的弦就不如细的弦振动得快或者说是振动的频率高。产生音调高低的不同，就是由于振动的频率不同。很显然频率越高，音高也就越高。

频率的单位是赫兹（简称为Hz），赫兹（1857-1894），是德国物理学家，他发现了电磁波，为了纪念他，人们用它的名字来做为频率的单位。所谓的一赫兹，就是一秒钟振动一次。那么440Hz呢，当然就是每秒振动440次，这个声音就是音乐中的标准A音，是乐器定音的标准。而钢琴中央C的频率则是261.63Hz。

我们人的耳朵能够听到的频率范围，是20Hz到20000Hz。低于这个频率范围的声音叫次声波，而高于这个频率范围的声音叫做超声波。所以我们能听到的音乐的频率，即都在人耳可听到的这个范围之内，约从20Hz到20KHz。

比如，下面的表格中是我们常见的一些人声的基频范围

- 男低音 80 - 320Hz
- 男中音 96 - 387Hz
- 男高音 122 - 488Hz
- 女低音 145 - 580Hz
- 女高音 259 - 1034Hz

根据傅立叶分析，任何声音可以分解为数个甚至无限个正弦波，而它们往往又包含有无数多的谐波分量。而它们又往往是时刻在变化着。所以一个声音的构成其实是很复杂的。将声音的频率分量绘制成曲线，就形成了频谱。

对频谱进行分析的仪器就是频谱分析仪，早期频谱仪都是模拟分析的。频谱仪的原理就是将声音

信号通过一系列不同中心频率的模拟带通滤波器。每个带通滤波器相当于一个共振电路，其特性由中心频率（步进的）、频带宽度及响应时间表示。在声音信号通过滤波器后，经过平方检波器，并进行平均之后，在每个频率上测定所传输的功率，从而得到信号的频谱。然而，传统的频谱仪受到滤波器性能的制约，因为模拟电路本身的特性所局限，滤波器的带宽和响应时间成反比，也就是说模拟滤波器的频率分辨力与时间分解能力之间存在矛盾。因为频谱仪所测量的往往都是非稳态声，一般来说，都是使用若干个滤波器来覆盖整个频率范围，并将信号同时并联地输入到这些滤波器上去。或者使用中心频率能够从低到高连续变化的滤波器。

随着科学技术的不断进步，现在我们所使用的基本不再是那些笨重而不准确的模拟仪器的频谱仪，取而代之的是基于处理器的软件分析法。它分析的数据来源其实是经过了 ADC（模数转换器件）转换后得出的数字信号，所以频谱仪软件所测量的信号准确度，很大程度取决于数模转换电路的性能。比起模拟滤波器，数字滤波器应该要更加迅速和精确。

2) 你问：那什么是均衡器和音乐频谱显示？在我们欣赏音乐的过程中有什么样的作用？

我答：对于录音棚等专业级别的音乐制作来说，通常都需要对录制的音频信号进行频谱分析来辅助音频的加工制作。当然对于我们普通的爱好者来说，不需要那么精确的频谱分析，我们更多的是用来略显音乐节奏，美化环境，增添气氛。于是，大多数的音乐播放软件(如图一)，手机，mp3，高档的音箱设备等，都有了音乐频谱显示。



图一 千千静听播放音乐时的频谱显示

当高级组合音响在工作时，你可以看到其中有一个部件上有一排一、二十个高高低低的推式调钮按频率高低自左而右排列，显示一种频率分布图；或者是还有一、二十个高高低低的光柱在不断地上下跳，时而中间起一个高峰，时而并排突起两、三个高峰，而且与音乐的高低和强度同步，高音越是强，高峰就偏右边；低音越是强，高峰就偏左边，这就是均衡器及其联结的频谱显示。有时，也用一排色彩不同的条纹来显示，条纹的色彩亮度随声音的强弱而同步地改变。

我们知道，实际上每一种音乐声，包括哪怕是一个人的独唱，一件乐器的独奏，都包括了一系列从低到高不同频率的声波，更不要说是乐队演奏了。如果声音中低频成分多，听起来就较厚实；如果是高音成分多，听起来就比较明亮。有的人喜欢听低音成分多些，有些人喜欢听高音成分多些，各不相同。因此，以前的收录机上就有一个调节高低音的旋钮。把旋钮指向高音部位，高音就比较强；把旋钮指向低音部位，低音就比较强，这就是一个简单的均衡器。

好的声音效果，其各段频率成分应该有一定的比例，录音的时候，录音师操作调音台就可以使各段频率的成分得到调整。由于各种乐器的基频高低是不同的，所以，也可以使各种乐器之间的声音比例得到调整，常见的是把频率由低至高分成 5 段或 7 段、10 段或 15 段，有经验的录音师或音乐家能听出哪里（哪个频率段）“空”了，即这个频段弱。哪里“鼓了个包”，即这个频段过强。通过均衡器可以把这些予以弥补。又因为每个人对音乐中频率分布的欣赏标准是不同的，因此，各位录音师掌握的尺度也不同，显示出各自的风格。

录音师也可以在录音时做些“手脚”，例如，有的录音机低音部分不太好，于是在录音时就把低音部分加强，这样，在这台录音机上播放时，低音就得到了补偿，因此，你播放购买收录机随机奉送的“试机带”时，听起来常常总是非常动人的，但在播放正常的磁带时，往往就不行了。所以在正式试

验整机和评比设备、磁带、唱片时，是不准用均衡器而且要用标准试机带的。由于每个人的欣赏和爱好的不同，有人喜欢低音“浓”些，有人则崇尚自然，也有人有时要强调某种乐器。因此在高级的放音设备上也有均衡器，可以根据每人不同的爱好去随心所欲地调整频率响应。同时，这也可以弥补录音带质量上的某些缺陷，例如，除了频响不平衡外，可以抑制某个频段上突出的噪声等。

3) 你问：我们制作音乐频谱显示需要用到什么基础知识吗？

我答：首先，我们采用是基于微处理器的软件分析法，微处理器是本制作的核心部分，因此，选择一个合适的性能与配置的微处理器是关键，我选择增强型的 8051 单片机便能够满足本次制作的要求。软件上的分析法核心是快速傅立叶算法，简称 FFT。

4) 你问：什么是 FFT？

我答：FFT（Fast Fourier Transform）即离散傅立叶变换的快速算法。在数字信号处理中常常需要用到离散傅立叶变换(DFT)，以获取信号的频域特征。尽管传统的 DFT 算法能够获取信号频域特征，但是算法计算量大，耗时长，且要求相当大的内存，不利于计算机实时对信号进行处理，限制了 DFT 的应用。直到 Cooley & Tukey 在 1965 年提出的快速离散傅立叶计算方法被发现，快速傅立叶变换算法才在实际的工程中得到广泛应用。需要强调的是，FFT 并不是一种新的频域特征获取方式，而是 DFT 的一种快速实现算法，可用来将一个信号从时域变换到频域。多数的复杂信号在进行频域变换之后，变换的目的实际上是从频域里来看同一个信号，从而容易分析出其信号的特性。这也是很多信号分析采用 FFT 变换的一个重要原因。另外，FFT 能将一个信号的频谱提取出来，这在频谱分析方面也是经常使用的。

5) 你问：我们的制作中如何使用 FFT？

我答：很简单，首先，我们用 ADC 去采样一个模拟信号之后，使之变为数字信号。根据采样定理，因音频信号的最高频率约为 20KHz，若 adc 的采样频率设置为 40KHz，既有采样频率大于信号频率的两倍，就能基本满足要求。其次，将采样得到的数字信号，送入 FFT 进行变换处理。通常，若我们取 N 个采样点，经过 FFT 运算之后，就可以得到 N 个点的 FFT 结果。但通常为了方便进行 FFT 运算，通常 N 取 2 的整数次方。设 F_s 为 ADC 的采样频率，N 为傅立叶变换的点数。则有最小分辨率 $f = F_s/N$ ，因此频谱显示的最低频率就是 f Hz，以后每向右移一个点，频率值将增加 f Hz。由于 FFT 结果的对称性，通常只使用前 N/2 个采样点的结果。

简而言之，使用 A/D 转换器对输入的音频信号进行采样以后，经过 FFT 变换，然后取某些频率项的幅值，量化显示，驱动 LED 点阵，点亮相应的 LED 灯。

限于篇幅，我们暂且不对 FFT 的数学原理进行过深的原理分析，以免打消大家的积极性，即使不很清楚 FFT 原理，按照我的步骤也可以做出漂亮动感的音乐频谱显示来。

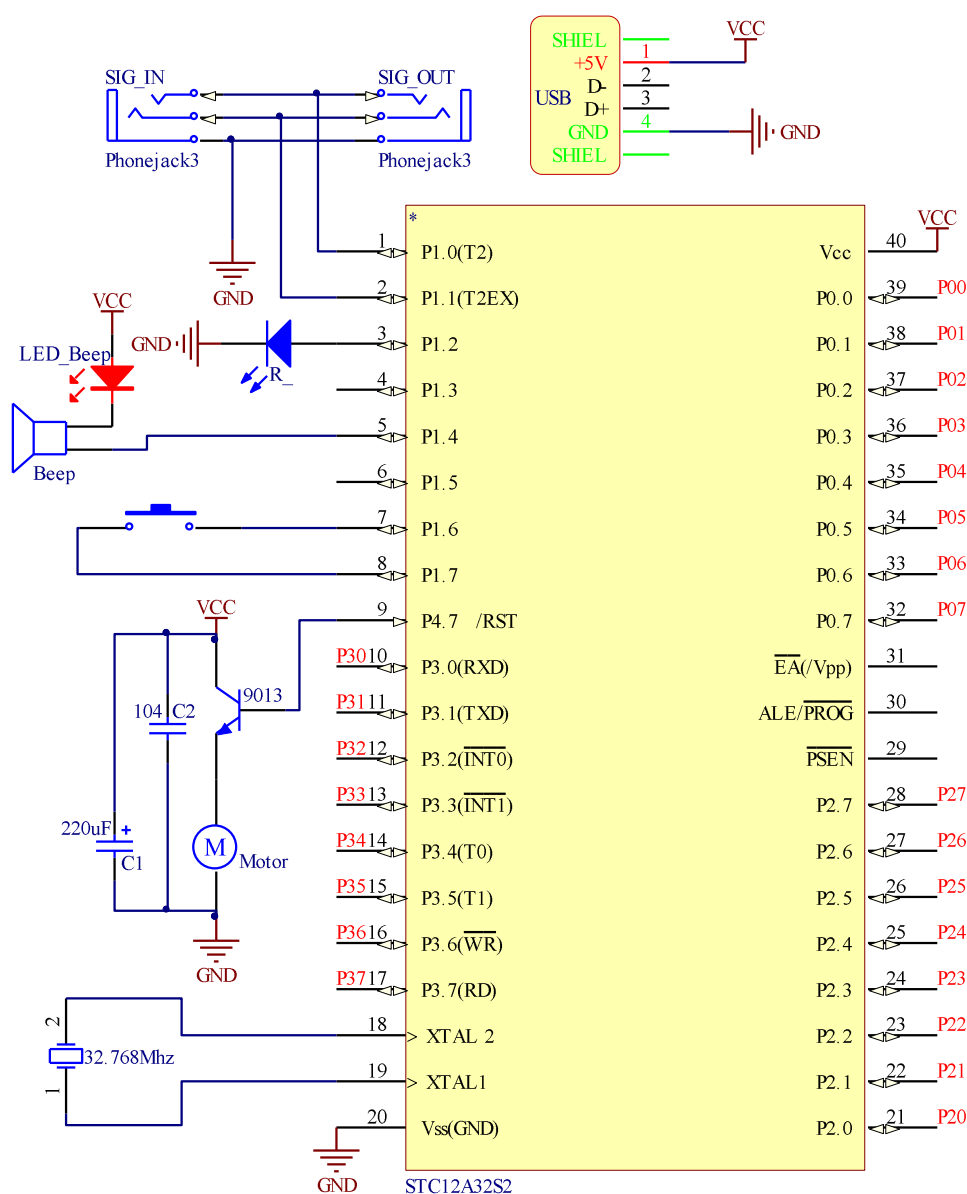
小贴士 (1):

根据 STC12A32S2 单片机的资源情况，最多只有 1280B RAM，我们取 64 点的 FFT 就可以满足要求，还达到更好的视觉效果，更高的刷新频率，以及为以后软件扩展留下余地。

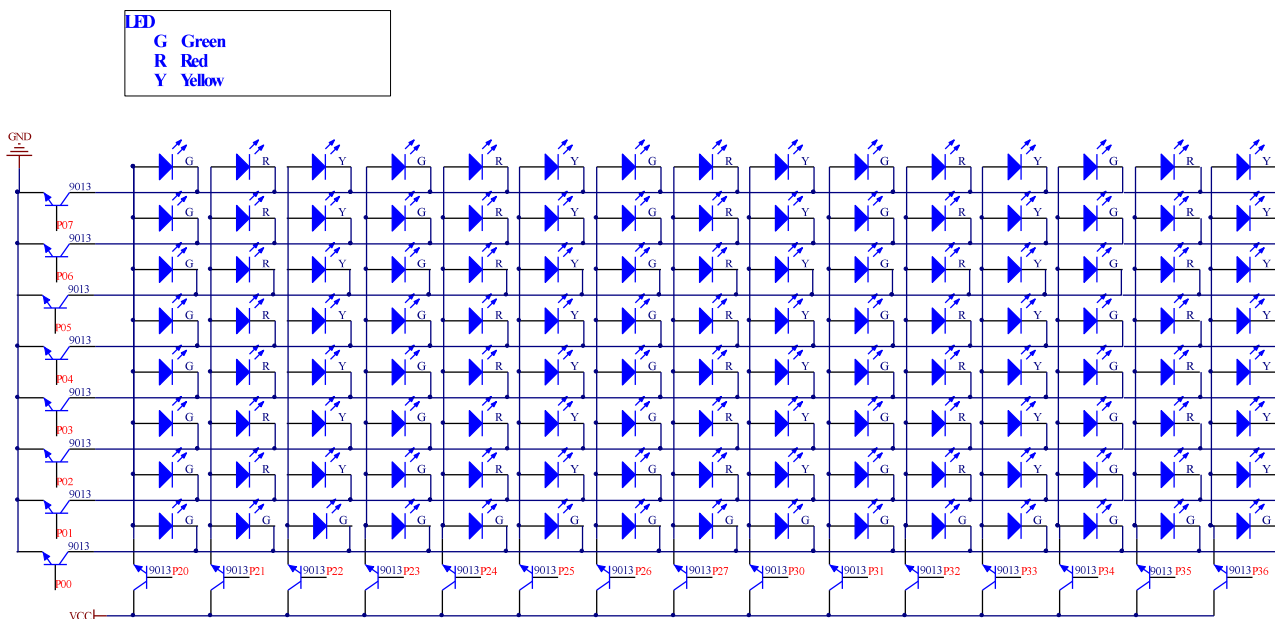
(二) 电路原理图

我们制作这个音乐频谱显示只是用来娱乐，所以没必要追求很高的精确度和工业级别稳定性，所以，我对电路进行了一系列的简化处理，这样既可以节省元器件，也可以减少 DIY 的难度，提高我们的兴趣。在不影响正常使用的情况下，我对电路进行了如下精简设计，但也相应的加入了不少亮点：

- 采用 USB 接口供电，并且对 USB 接口进行了扩展。在没有额外占用电脑主机 USB 接口情况下，随时随地给系统供电；
- 加入了触摸键设计，以及震动反馈。当触摸键响应时有震动反馈，及声光提示，如今很多触屏手机也有这种时尚设计；
- 加入 ThinkPad 笔记本上的经典呼吸灯指示设计。如夏日里的萤火虫，一闪一闪亮晶晶，不仅有趣还能指示系统工作状态；
- 省去了 MCU 的复位电路，以及晶振上的 2 个起振瓷片电容，但一般不影响系统正常工作；
- 双音频插孔设计。不影响正常的音频输出，省去了音频分线器；
- 去掉音频输入处理的电平移动电路，简化电路，并加入软自动增益控制，实现自动根据音乐的声音大小调节显示幅度；
- 采用食人鱼 LED 来显示，不仅特别的亮，而且比普通的草帽状 LED 效果更美，耐电流也更大，还省去了部分限流电阻，食人鱼 LED 有四个引脚，便于手工焊接，大量减少跳线。



图二 MCU 部分的原理图

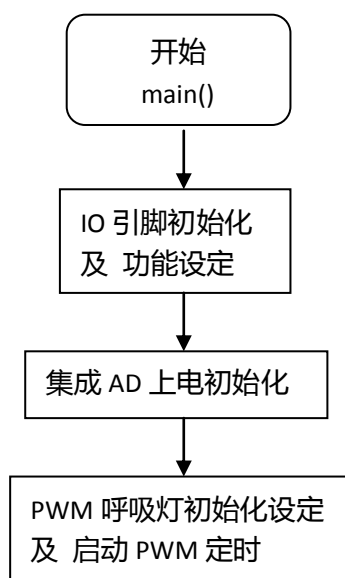


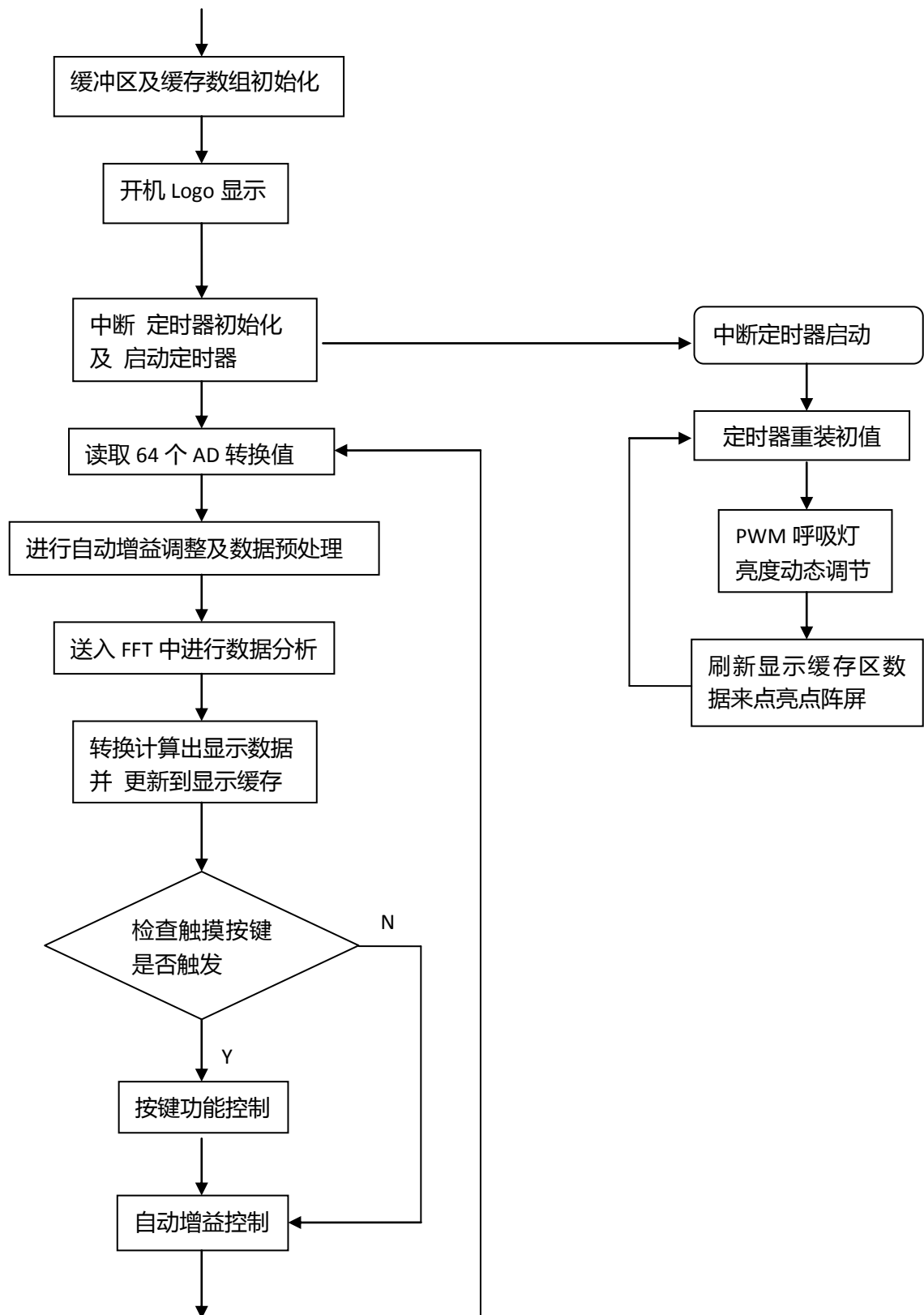
图三 点阵驱动电路图

(三) 编程思路

- 1) 在主函数中，单片机通过 AD 对音频数据采样，然后存放到数据缓存区进行预处理，完成 AD 滤波处理，自动增益控制信息扫描以及其他信息处理。接着，将缓存区数据送入 快速傅立叶变换 (FFT) 处理子函数进行运算。处理完后，从缓存区取出运算结果，根据得到幅值计算出点阵的显示数据，并存储到显示缓存区。
- 2) 在中断函数中，根据显示缓冲区的内容对点阵显示屏进行实时刷新点亮。

(四) 程序流程图





(五) 程序摘讲

- a) 我们使用函数“IOINIT();”对输入输出进行初始化,及设定好 IO 模式寄存器,我们需要把相应的 IO 配置成所需的引脚模式。根据 STC12A 系列的 Datasheet, IO 口可以设置成如下图所示的四种模式:

P1口设定 <P1.7, P1.6, P1.5, P1.4, P1.3, P1.2, P1.1, P1.0口>(P1口地址: 90H)

P1M1 [7 : 0]	P1M0 [7 : 0]	I/O 口模式 (P1.x 如做A/D使用, 需先将其设置成开漏或高阻输入)
0	0	准双向口(传统8051 I/O 口模式), 灌电流可达20mA, 拉电流为230 μ A, 由于制造误差, 实际为250 μ A~ 150 μ A
0	1	推挽输出 (强上拉输出, 可达20mA, 要加限流电阻)
1	0	仅为输入 (高阻), 如果该I/O口需作为A/D使用, 可选此模式
1	1	开漏(Open Drain), 如果该I/O口需作为A/D使用, 可选此模式

图四 IO 模式寄存器

特别需要注意的是要将 P1.0, P1.1 口设置成输入高阻态, 用作 ADC, 语句“P1ASF =B(00000011);”将 P10, P11 的 IO 设置为模拟输入功能;

小贴士 (2):

这里有个特别的宏用法, 把这段语句加到你程序的第一行即可:

```
#define BinToHex(n)
(((n>>21)&0x80)|((n>>18)&0x40)|((n>>15)&0x20)|((n>>12)&0x10)|((n>>9)&
0x08)|((n>>6)&0x04)|((n>>3)&0x02)|((n)&0x01)) //请把这条宏语句写在同一行。
#define B(n) BinToHex(0x##n##l)
```

举个例子:

原本给单片机 IO 赋值的语句“P0=0x37”;我们现在就可以写成“P0=B(00110111);”这样, 很大程度上方便我们修改 IO 口的状态。

- b) 函数“InitADC();”用来初始化单片机内部集成的模数转换器。在使用 ADC 转换时, 要记得提前上电, 待 ADC 的电源稳定后才使能转换。
- c) Timer_INT();//定时器初始化设定, 定时器用于 LED 屏的刷新及频谱数据显示;
- ```
for(i=0;i<34;i++)
LEDBuf[i]=0x00; //初始化显示存储缓冲区;
FontDisp();//显示 Logo 标志, 或者其他开机动画;
```
- d) for(i=0;i<64;i++)//读取 64 次 ADC 转换结果,
- ```
{
    dd[i].real=(GetADCResult(0)+GetADCResult(1))<<gain;
    //读取 ad 结果并放大 2 的 gain (变量) 次方, 然后给 FFT 算法的缓存区赋值;
}
```

你也许会问“...<<gain;”是用来做什么呢? 且听我慢慢道来:

我们知道, 如“x=0x11<<1;”即将 0x11 乘以 2, 再赋给 x, “x=0x11<<4;”就是将 0x11 乘以 2 的四次方, 再赋给 x, 我想你已经明白了吧。在很多场合, 我们用左移 (<<) 或右移 (>>) 来代替积商的运算, 能够提高程序的执行速度。那么在此处, “...<<gain;”便是用来调节增益的。下面的代码是自动增益的控制块:

```

Cgain=dd[2].real/32;
if(7<Cgain<=8)
{
    gain=4;
}
else if(4<Cgain<=6)
{
    gain=5;
}
else if(2<Cgain<=4)
{
    gain=6;
}
else
{
    gain=7;
}

```

小贴士 (3):

自动增益是什么?

当输入的音频信号的音量变大或变小时, 采样的 AD 值幅度也会有相应趋势的变化, 于是根据这个值自动调节 AD 值放大的倍数, 维持在一定的范围内, 从而使显示的效果最好, 不会出现满屏或者空屏的现象。

- e) 如何制作呼吸灯呢? 想让你的 LED 像夏日里的萤火虫般一闪一闪的吗? 快把这段语句加到你的中断程序中去吧:

```

if(flag)
{
    if(++x==0xFF)          //MINOUTPUT
    {
        flag=0;
    }
}
else
{
    if(--x==0xCF)          //MAXOUTPUT
    {
        flag=1;
    }
}
PWM0_set(x);              //设置 PWM 占空比

```

小贴士 (3):

不要忘记在主函数中初始化 PWM 寄存器噢, 即 “PWM_init();”!

二、制作篇

1) 请根据下面的元器件清单来准备好原材料。

序号	器件	型号规格	数量	备注
A	集成 AD 的单片机	STC12A32S2	1	此系列的其他单片机亦可
B	音频接口座	3.5mm	2	用来输入/输出音频信号
C	拨动开关		2	起开关或选择作用
D	USB 母头接口		2	USB 供电以及 USB 延伸扩展
E	食人鱼 LED	绿色	51	
		黄色	35	
		红色	36	
F	石英晶振	32.768Mhz	1	
G	震动电机		1	可从废旧手机上拆下
H	贴片三极管	9013	24	
I	蜂鸣器		1	作为声音提示
J	贴片钽电容	227J	2	滤波，稳定电源
K	瓷片电容	104pF	2	
L	导线		若干	
M	万能板	28*50	1	

2) 当然，你还得有电烙铁，焊锡丝，助焊剂，镊子，钳子，万用表等常用工具。

单片机可选择 STC12A 系列的其他型号，不过建议选 ROM 空间稍大点的，以便以后续升级程序功能，但附件提供的程序 ROM 资源消耗量不到 5K。另外，若读者一时没法找到食人鱼 LED，也可以用普通的草帽型 LED 代替，此时就可以省略 9013 了。因为食人鱼 LED 封装是正方形的，且相邻的 LED 之间不会有空隙，而且亮度很好，故强烈推荐大家用食人鱼 LED 来制作。

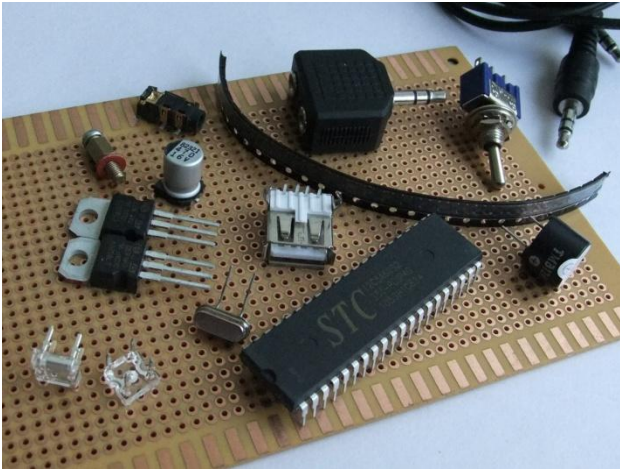


图 A 准备好元器件

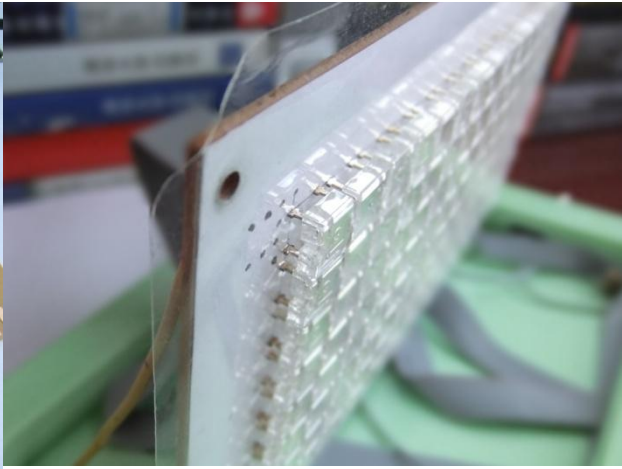


图 B LED 的优美排列

3) 准备焊接,在 28*50 孔的万能板上插好食人鱼 LED，请大家一定要注意不要把正负极装反了，有时间的

话尽量先测试下 LED 灯，不同颜色的 LED 正负极可能是不同的！在万能板上铺上一张白纸和一张透明薄膜，可以用万能表笔打孔，白纸可以提高光的反射程度，薄膜也可以防尘防水。

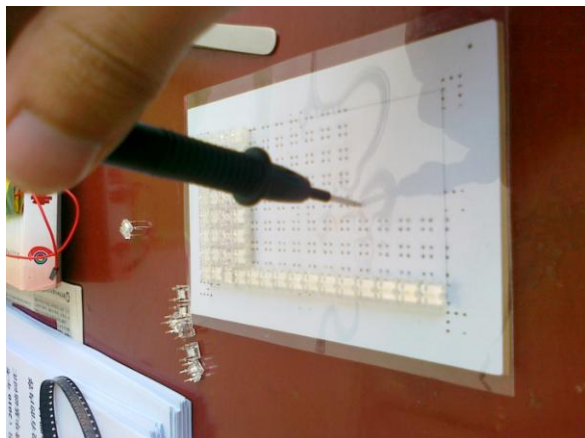


图 C 正在顺序安装 LED

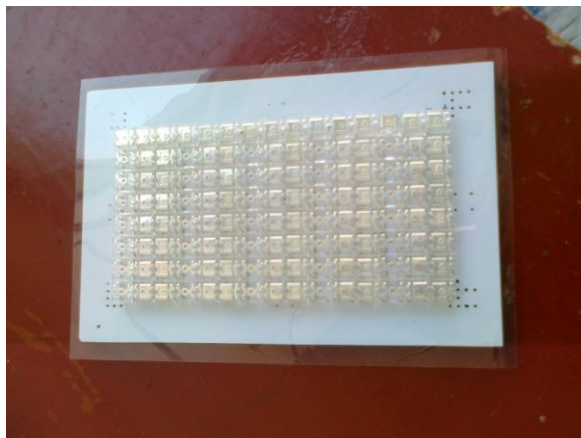
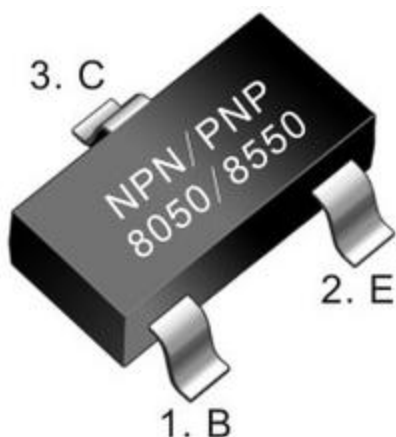


图 D 插好的 LED 屏

- 4) 插好 LED 后，在万能板上焊接好贴片 9013,如果没有贴片的也没关系，就用普通的也可以，注意不要短路了，再按从左到右的顺序依次接上 15 根行和 8 根列的排线，把 24 根排线分别接到三极管的基极（引脚如下图四所示），并把电源线，地线接到三极管上并引出。贴片类的三极管器件耐热性能不够好，如果长时间加热可能会让贴片元器件性能降低甚至损坏，所以烙铁温度不能太高，焊接时间不能过长。



图四 普通贴片三极管的引脚示意图

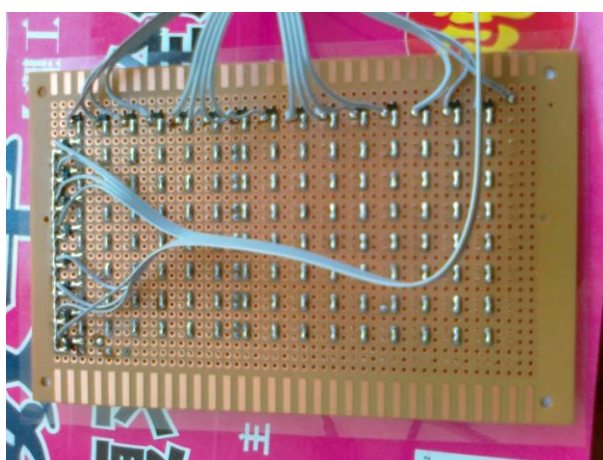


图 E 焊接贴片三极管和排线



图 F 当 LED 屏邂逅台历外壳

- 5) 把 LED 点阵屏装到外壳里，我是用的一个日历的外壳，这个外壳可以自己就地取材，大小合适就好。

然后用笔做好需要钻孔的标记，不要画偏了，拿一个闲置不用的电烙铁，加热钻好孔，并用小刀整平。取几个铜柱，拧好螺丝，装上初步做好的 LED 点阵屏，并固定好。当然你也可根据自己的爱好加装些别的元器件。

需要注意的是这种音频插孔有五个引脚，我们只用外围的三个引脚，如下图五所示:标号为 1 的引脚是 GND，2 与 3 分别为左右声道。



图五 音频插孔引脚，

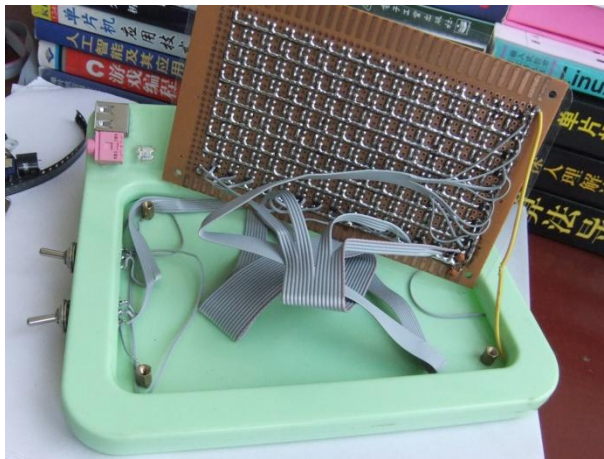


图 G 焊接好的 LED 屏放入外壳中



图 H 用铜柱将 LED 屏固定在外壳中

- 6) 装上电源开关，以及 USB 电源接口，音频插孔等。完成最后的组装，把各个器件的连线接到单片机的引脚上，把点阵屏的 列排线分别按顺序接到单片机的 P0 口，P2 口，以及将行选排线接到 P3 口上，不要粗心大意接错线了，并接好其他的零星的连线，最主要的是电源引线，尤其是需要连接到地的引线比较多。为了避免我们辛苦制作的成果因短路而烧毁，上电测试前，请用万能表测量系统的电源与地之间的电阻，如果发现测量电阻值过小，那就应该检查下是否有短路发生，如果是用电脑的 USB 供电，短路将有可能损坏电脑。

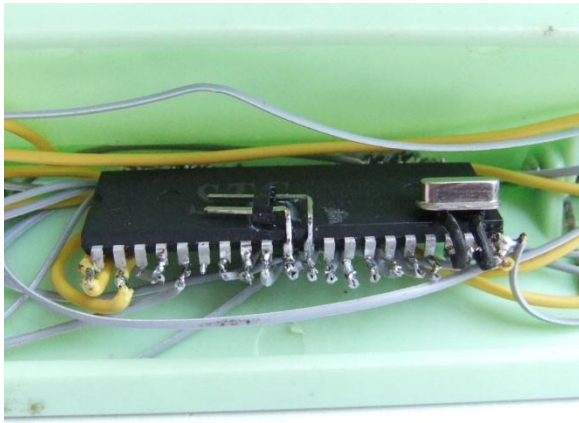


图 I 将 LED 屏的排线顺序焊接到单片机上



图 J 用导线连接好其他器件的引脚



图 K 将滤波电容接到电源上



图 L 将震动电机接好并固定

- 7) 检查好硬件后，用 keil uVision4 打开音乐频谱显示源代码的工程文件（如下图六所示），点击编译按钮



，编译生成 hex 文件（如图七所示），可以看到程序体积小于 5K，并下载好单片机固件程序(程序源码及简单注释在附件里)，接上音频输入及 USB 电源即可。上电后的 2 秒内，将运行 LED 屏的测试程序，如果发现有个别的 LED 灯不亮,可能需要检查下你的线路连接，接下来请把音频延长线插到频谱显示的音频输入插孔，另一端接到正在播放音乐电脑或者 mp3 上，调节音量到合适的输出大小，如果没有错误，你将可以看到跃动的柱状频谱，呼吸灯也早已经开始“呼吸”。待调试好了以后，一切正常工作，就可以用 502 胶水把元器件粘在外壳上固定好，这样既美观又牢固。不过不要在测试之前把元器件粘贴起来，否则如有元器件的损坏，不便更换。如果你想一览为快，具体的视频效果请移步我的优酷空间观看：<http://u.youku.com/mhlyjay>。效果如下图：



图六 源代码中的工程文件

```
Program Size: data=132.0 xdata=537 code=4842
creating hex file from ".\Hex\Prj"...
".\Hex\Prj" - 0 Error(s), 2 Warning(s).
```

图七 编译信息



图 M 组装好的成品一览



图 N 上电测试 LED 屏



图 O 开机滑动显示 Logo



图 P 跳跃的柱状频谱

- 8) 另附赠基于杜洋 mini1608 硬件结构的音乐频谱显示的 hex 固件，需要使用 32.768Mhz 的无源晶振，有

感兴趣的读者可以马上下载试试。

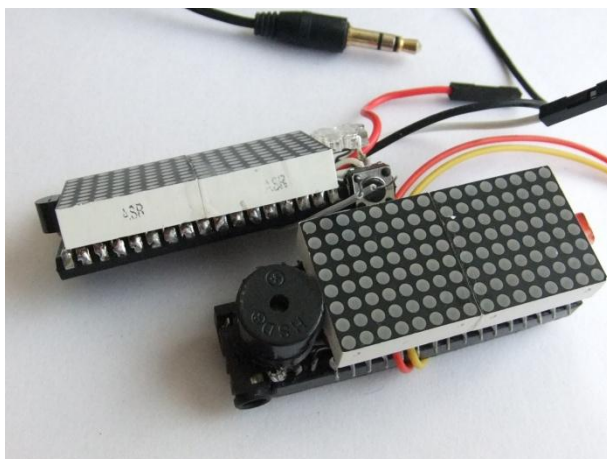


图 Q 自制类 mini1608 的硬件

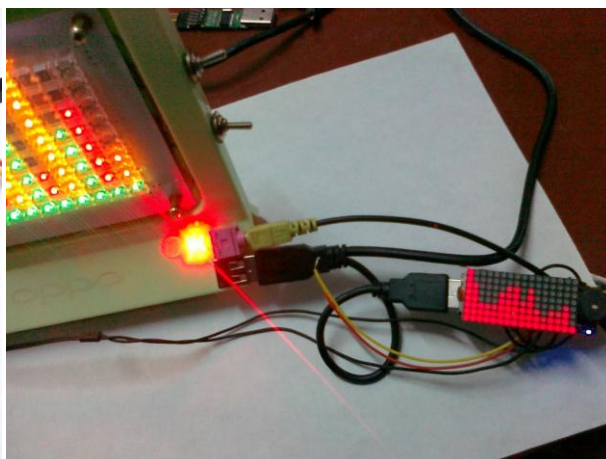


图 R 基于 mini1608 接法的另类频谱显示

<End 2011/4/25>

基于“手把手教你用 51 单片机 DIY 音乐频谱显示_稿件 - V2.docx”，修改了“**制作篇**”。