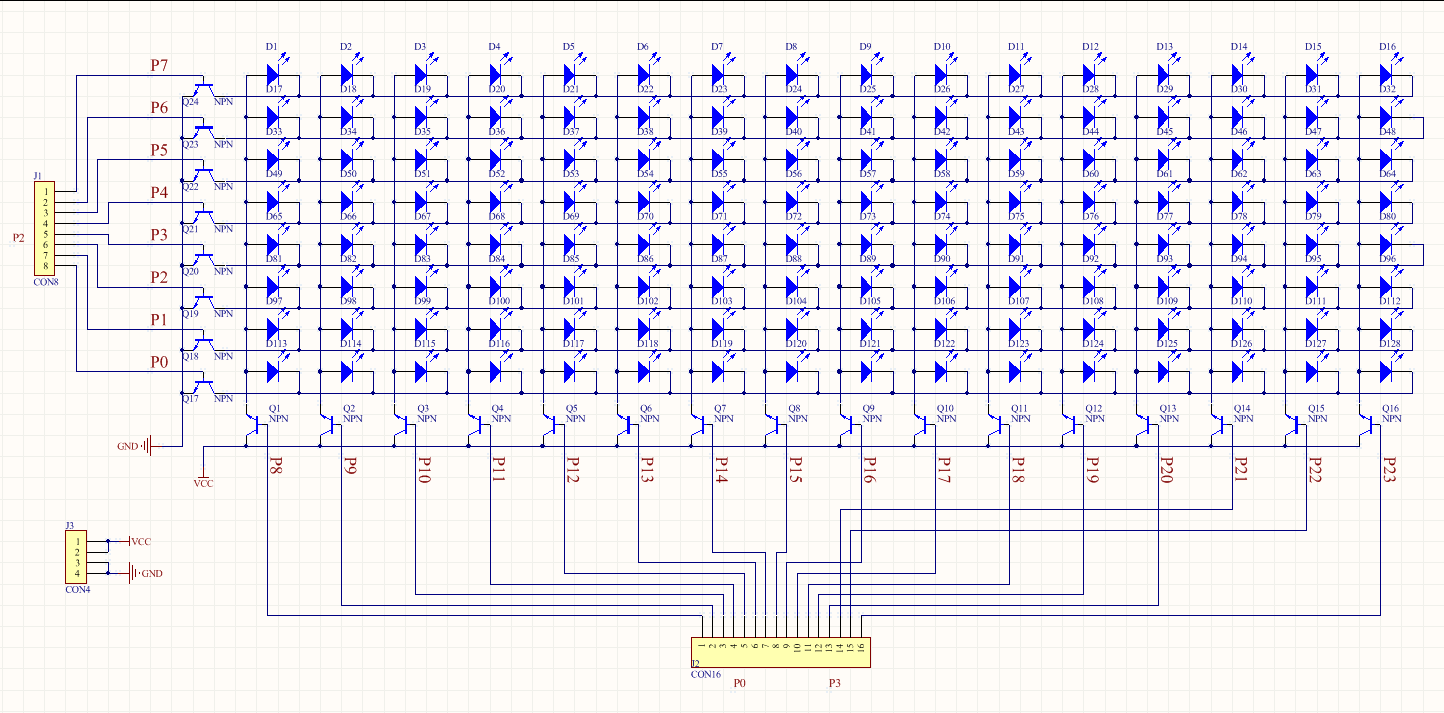
**带你玩51单片机频谱显示**

**——Angleあ辉**

**——2017.5.6**

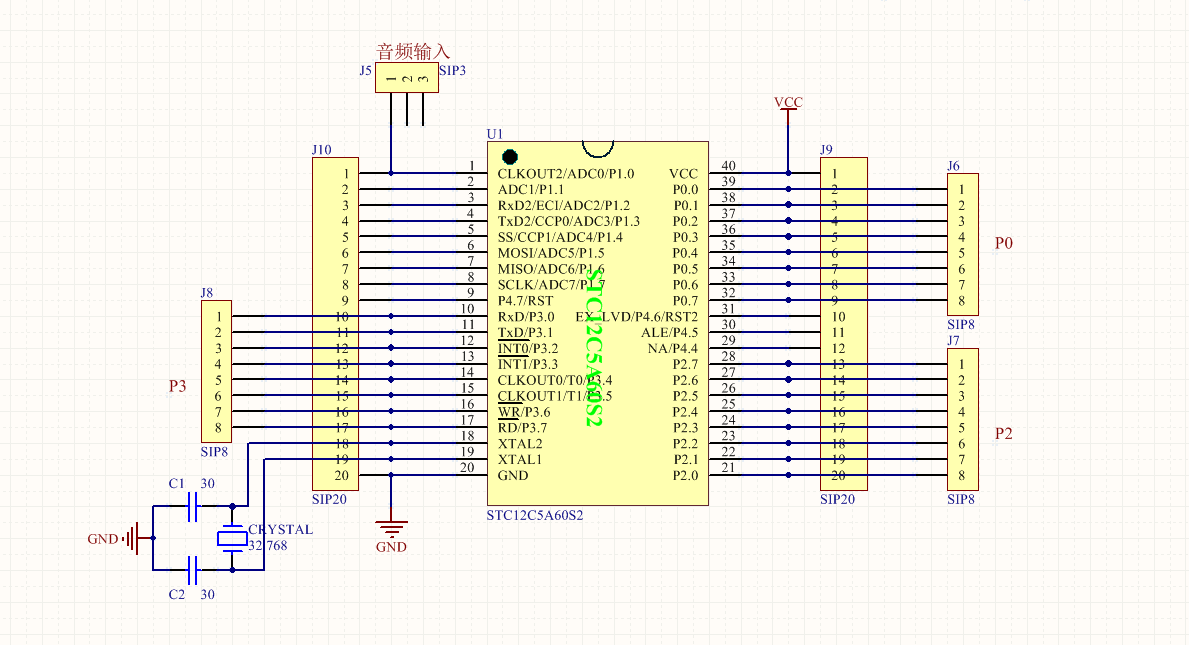
1. **原理图：**
2. **LED点阵原理图：**

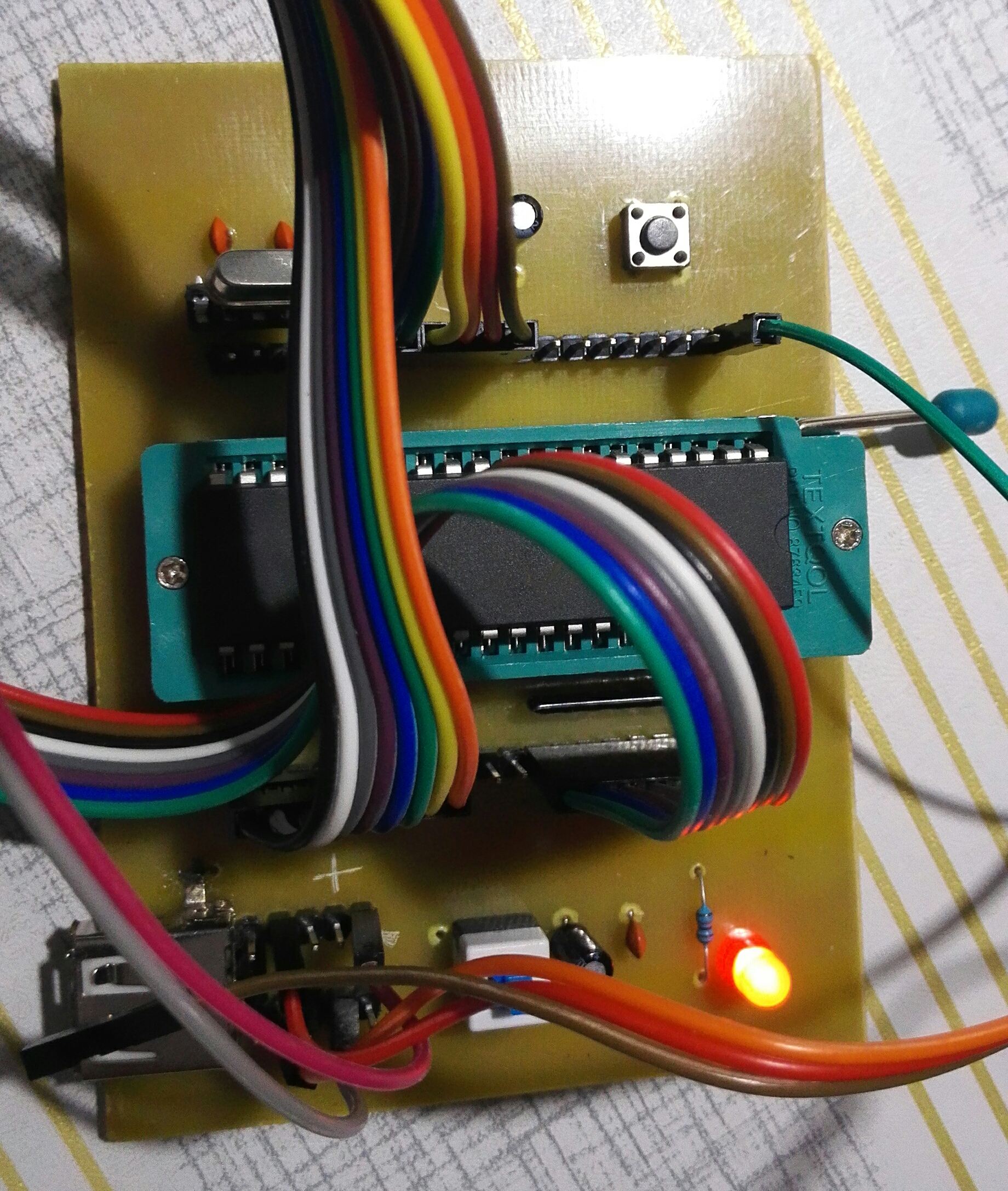
**用到128个蓝色LED（颜色可以根据自己喜爱而选），纵有8列，横有16排。用P2 I/O控制纵向，用P0,P3 I/O控制横向，三极管用NPN S9013或其他。**

****

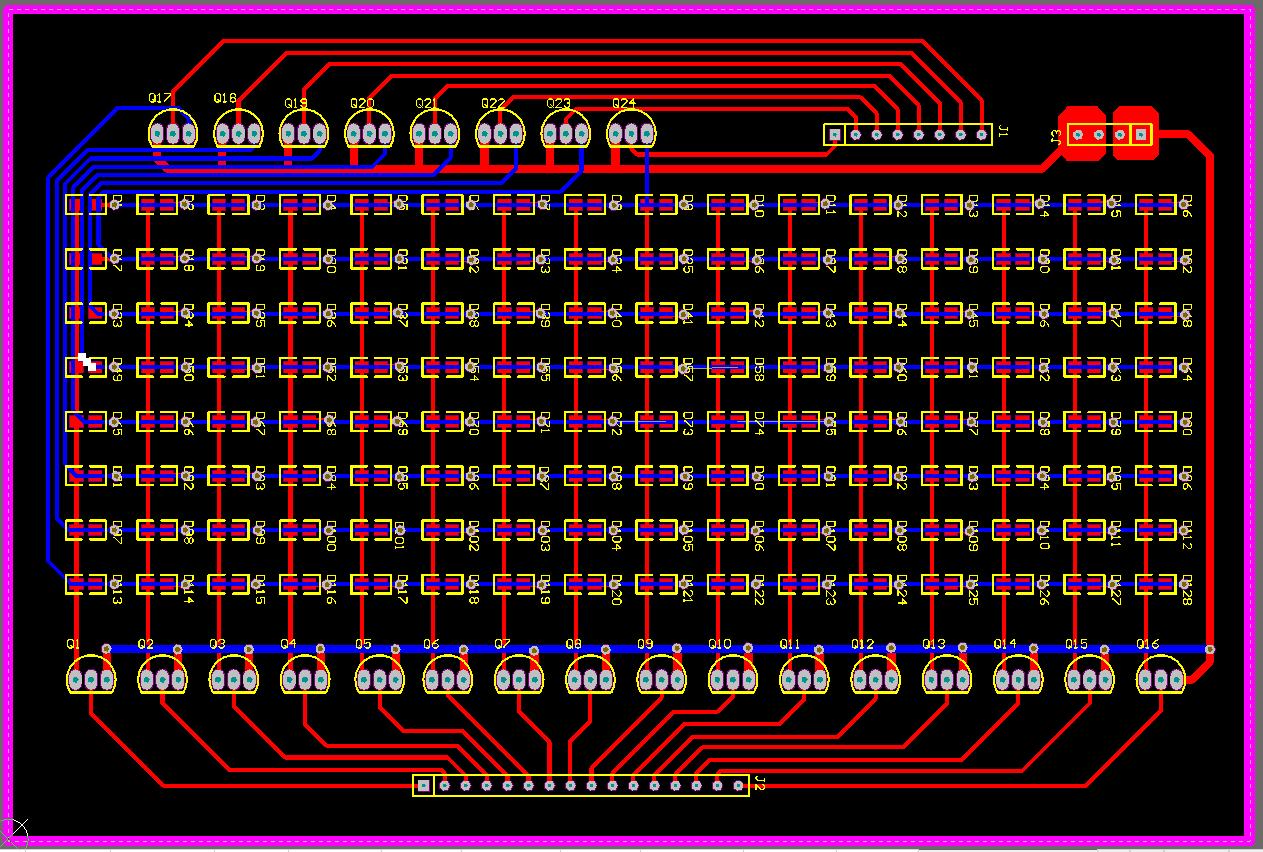
1. **控制系统原理图：**

**用到的芯片是STC12C5A60S2，音频输入这里只用一个声道输入采集**

****

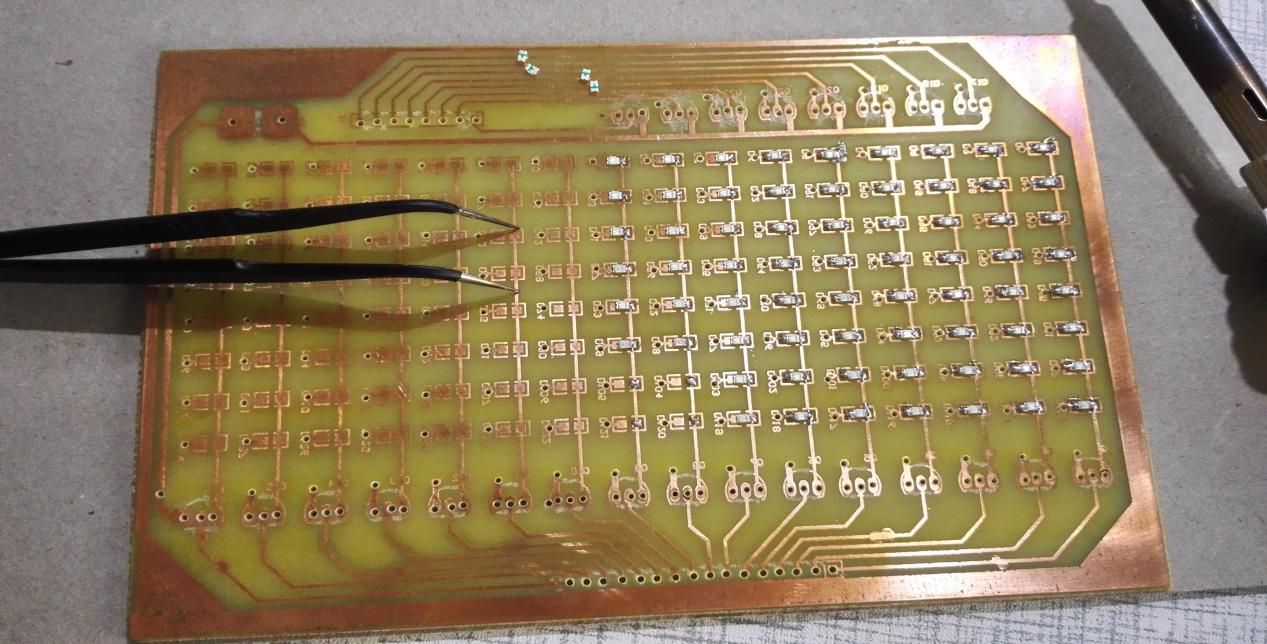
****

**3）PCB图：**

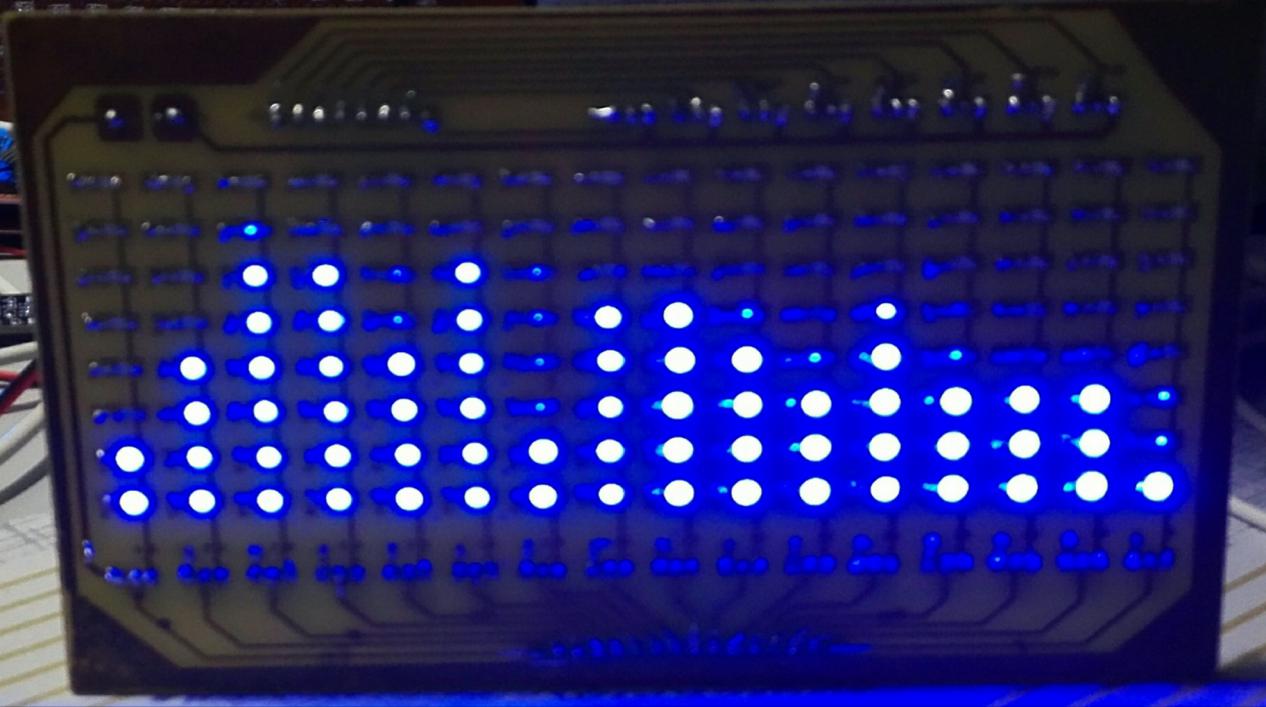
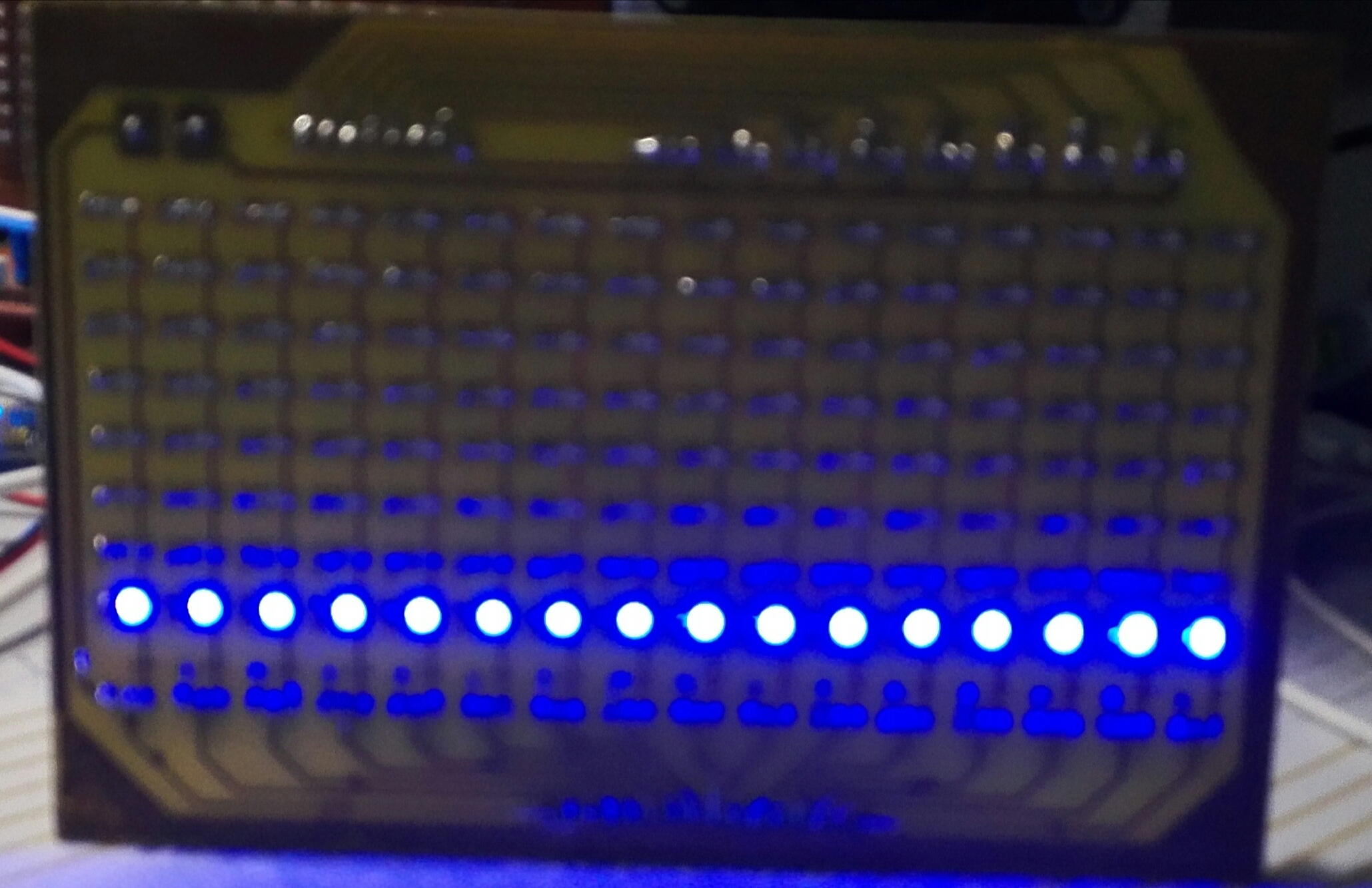
****

**4）制作的效果图：**

**PCB图是画的双面板（因为条件有限只能做单面板，另一面用跳线）**

****

1. **完成效果显示：**

****

1. **音频代码：**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\* 频谱代码 \*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Angleあ辉（修改） \*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ———2017.5.6 \*/**

**#include <STC12C5A60S2.h> //头文件**

**#include<intrins.h>**

**#define SAMPLE\_NUM 128**

**#define LOG 6**

**#define TIME 2048 //中断**

**#define FFT\_OUT\_MIN 0**

**#define FFT\_OUT\_MAX 7**

**#define OFF L1=L2=L3=L4=L5=L6=L7=L8=L9=L10=L11=L12=L13=L14=L15=L16=0; //对P0,P3赋值**

**#define ADC\_POWER 0x80 //ADC power control bit**

**#define ADC\_FLAG 0x10 //ADC complete flag**

**#define ADC\_START 0x08 //ADC start control bit**

**#define ADC\_SPEEDLL 0x00 //420 clocks**

**//#define ADC\_SPEEDL 0x20 //280 clocks**

**#define ADC\_SPEEDH 0x40 //140 clocks**

**//#define ADC\_SPEEDHH 0x60 //70 clocks**

**//采样存储序列表**

**unsigned char code BRTable[SAMPLE\_NUM] ={0, 64, 32, 96, 16, 80, 48, 112,**

**8, 72, 40, 104, 24, 88, 56, 120,**

**4, 68, 36, 100, 20, 84, 52, 116,**

**12, 76, 44, 108, 28, 92, 60, 124,**

**2, 66, 34, 98, 18, 82, 50, 114,**

**10, 74, 42, 106, 26, 90, 58, 122,**

**6, 70, 38, 102, 22, 86, 54, 118,**

**14, 78, 46, 110, 30, 94, 62, 126,**

**1, 65, 33, 97, 17, 81, 49, 113,**

**9, 73, 41, 105, 25, 89, 57, 121,**

**5, 69, 37, 101, 21, 85, 53, 117,**

**13, 77, 45, 109, 29, 93, 61, 125,**

**3, 67, 35, 99, 19, 83, 51, 115,**

**11, 75, 43, 107, 27, 91, 59, 123,**

**7, 71, 39, 103, 23, 87, 55, 119,**

**15, 79, 47, 111, 31, 95, 63, 127};**

**char code sin\_tabb[SAMPLE\_NUM] = {0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 59, 65, 70, 75, 80, 85, 89, 94, 98, 102,**

**105, 108, 112, 114, 117, 119, 121, 123, 124, 125, 126, 126, 126, 126, 126, 125, 124,**

**123, 121, 119, 117, 114, 112, 108, 105, 102, 98, 94, 89, 85, 80, 75, 70, 65, 59, 54,**

**48, 42, 36, 30, 24, 18, 12, 6, 0,**

**-6, -12, -18, -24, -30, -36, -42, -48, -54, -59, -65, -70, -75, -80, -85, -89, -94, -98, -102,**

**-105, -108, -112, -114, -117, -119, -121, -123, -124, -125, -126, -126, -126, -126, -126, -125,**

**-124, -123, -121, -119, -117, -114, -112, -108, -105, -102, -98, -94, -89, -85, -80, -75, -70,**

**-65, -59, -54, -48, -42, -36, -30, -24, -18, -12, -6};**

**char code cos\_tabb[SAMPLE\_NUM] = {127, 126, 126, 125, 124, 123, 121, 119, 117, 114, 112, 108, 105, 102, 98, 94,**

**89, 85, 80, 75, 70, 65, 59, 54, 48, 42, 36, 30, 24, 18, 12, 6, 0, -6, -12, -18, -24, -30, -36, -42, -48, -54, -59,**

**-65, -70, -75, -80, -85, -89, -94, -98, -102, -105, -108, -112, -114, -117, -119, -121, -123, -124, -125, -126, -**

**126, -126, -126, -126, -125, -124, -123, -121, -119, -117, -114, -112, -108, -105, -102, -98, -94, -89, -85, -80,**

**-75, -70, -65, -59, -54, -48, -42, -36, -30, -24, -18, -12, -6, 0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 59, 65, 70,**

**75, 80, 85, 89, 94, 98, 102, 105, 108, 112, 114, 117, 119, 121, 123, 124, 125, 126, 126};**

**xdata unsigned char result[128];**

**xdata unsigned char temp[128];**

**xdata unsigned char num[128];**

**unsigned char anum,timernum,timernum2,lednum3,Ltime,t; //用于分离**

**unsigned char lednum[]={0x01,0x03,0x07,0x0f,0x1f,0x3f,0x7f,0xff}; //0-7的显示数组 P2组控制**

**int xdata FftReal[SAMPLE\_NUM];**

**int xdata FftImage[SAMPLE\_NUM];**

**unsigned char code musicdisplay[]={ //显示内容数组（可以根据自己的编写更显示内容）**

**0x00,0x00,0x00,0x00,0x00,0x00, //缓冲区**

**0x00,0x00,0x00,0x00,0x00,0x00,**

**0x00,0x00,0x00,0x00,0x00,0x00,**

**0x21,0x71,0xF9,0x71,0x71,0x71, // <==**

**0x71,0x01,0x01,**

**0x89,0xF9,0x89,0x01, // -I-**

**0x01,**

**0xF9,0x09,0x09,0x01, // -L-**

**0x71,0x89,0x71,0x01, // -O-**

**0xF1,0x09,0xF1,0x01, // -V-**

**0xF9,0xA9,0xA9,0X01, // -E-**

**0x01,**

**0xE1,0x19,0xE1,0x01, // -Y-**

**0x71,0x89,0x71,0x01, // -O-**

**0xF9,0x09,0xF9,0x00, // -U-**

**0x00,0x00,**

**0x78,0x84,0x42,0x21,0x42,0x84,0X78, //--心--**

**0x00,0x00,0x00,0x00,0x00,0x00,**

**0x00,0x00,0x00,0x00,0x00,0x00,**

**0x00,0x00,0x00,0x00,0x00,0x00,**

**};**

**#define LED P2**

**#define pwm 9**

**/////////// 定义列扫描端口;,用到P0，P3 I/O口**

**sbit L9=P3^0;**

**sbit L10=P3^1;**

**sbit L11=P3^2;**

**sbit L12=P3^3;**

**sbit L13=P3^4;**

**sbit L14=P3^5;**

**sbit L15=P3^6;**

**sbit L16=P3^7;**

**sbit L1=P0^0;**

**sbit L2=P0^1;**

**sbit L3=P0^2;**

**sbit L4=P0^3;**

**sbit L5=P0^4;**

**sbit L6=P0^5;**

**sbit L7=P0^6;**

**sbit L8=P0^7;**

**void delay(unsigned char a) //延时函数**

**{**

**unsigned int i;**

**while(--a)**

**for(i=0;i<50;i++);**

**}**

**void off() //初始化P0,P3 I/O口**

**{**

**LED = 0x00;**

**OFF**

**}**

**void display(unsigned char i,unsigned char dat) //显示数组扫描函数**

**{**

**P2=dat;**

**switch(i)**

**{**

**case 1:L1=1;break;**

**case 2:L2=1;break;**

**case 3:L3=1;break;**

**case 4:L4=1;break;**

**case 5:L5=1;break;**

**case 6:L6=1;break;**

**case 7:L7=1;break;**

**case 8:L8=1;break;**

**case 9:L9=1;break;**

**case 10:L10=1;break;**

**case 11:L11=1;break;**

**case 12:L12=1;break;**

**case 13:L13=1;break;**

**case 14:L14=1;break;**

**case 15:L15=1;break;**

**case 16:L16=1;break;**

**}**

**delay(pwm);**

**off();**

**}**

**void show\_musicdisplay() //显示延时函数**

**{**

**unsigned char a,b,c;**

**for(c=0;c<70;c++)**

**for(b=20;b>0;b--)**

**for(a=0;a<35;a++)**

**display(a,musicdisplay[a-1+c]);**

**}**

**void FFT\_process() //下落迟滞**

**{**

**unsigned char i;**

**for(i=0;i<16;i++)**

**{**

**if(result[i] < temp[i])**

**{**

**num[i]++;**

**if(num[i] == 1)**

**{**

**if(temp[0]==0&&temp[1]==0&&temp[2]==0&&temp[3]==0&&temp[4]==**

**0&&temp[5]==0&&temp[6]==0&&temp[7]==0&&temp[8]==0&&temp[9]==**

**0&&temp[10]==0&&temp[11]==0&&temp[12]==0&&temp[13]==0&&temp[14]==**

**0&&temp[15]==0&&temp[16]==0);**

**else**

**{**

**result[i] = --temp[i];**

**num[i] = 0;**

**}**

**}**

**}**

**else**

**num[i] = 0;**

**}**

**}**

**void disp()**

**{**

**timernum++;**

**if(timernum == 17)**

**timernum = 1;**

**OFF**

**FFT\_process();**

**switch(timernum) //频谱显示扫描函数**

**{**

**case 1:anum = result[7];L1=1;break;**

**case 2:anum = result[6];L2=1;break;**

**case 3:anum = result[5];L3=1;break;**

**case 4:anum = result[4];L4=1;break;**

**case 5:anum = result[3];L5=1;break;**

**case 6:anum = result[2];L6=1;break;**

**case 7:anum = result[1];L7=1;break;**

**case 8:anum = result[0];L8=1;break;**

**case 9:anum = result[8];L9=1;break;**

**case 10:anum = result[9];L10=1;break;**

**case 11:anum = result[10];L11=1;break;**

**case 12:anum = result[11];L12=1;break;**

**case 13:anum = result[12];L13=1;break;**

**case 14:anum = result[13];L14=1;break;**

**case 15:anum = result[14];L15=1;break;**

**case 16:anum = result[15];L16=1;break;**

**}**

**if(anum >= 8)**

**anum = FFT\_OUT\_MAX;**

**LED = lednum[anum];**

**}**

**void ADC\_init()**

**{**

**P1ASF = 0x01; //Open 8 channels ADC function**

**ADC\_RES = 0; //Clear previous result**

**ADC\_RESL= 0;**

**ADC\_CONTR = ADC\_POWER | ADC\_SPEEDH;**

**delay(5); //ADC power-on and delay**

**}**

**float ADC\_read()**

**{**

**ADC\_CONTR = ADC\_POWER | ADC\_SPEEDH | ADC\_START;**

**\_nop\_(); //至少要延时4个时钟周期**

**\_nop\_();**

**\_nop\_();**

**\_nop\_();**

**\_nop\_();**

**while (!(ADC\_CONTR & ADC\_FLAG));//Wait complete flag**

**ADC\_CONTR &= ~ADC\_FLAG; //关闭ADC**

**return (ADC\_RES\*4+ADC\_RESL); //Return ADC result**

**}**

**short sqrt\_16(unsigned long M)**

**{**

**unsigned int N,i;**

**unsigned long tmp,ttp;**

**if( M == 0 )**

**return 0;**

**N = 0;**

**tmp = (M >> 30);**

**M <<= 2;**

**if( tmp > 1 )**

**{**

**N++;**

**tmp -= N;**

**}**

**for( i=16; i>0; i-- )**

**{**

**N <<= 1;**

**tmp <<= 2;**

**tmp += (M >> 30);**

**ttp = N;**

**ttp = (ttp<<1)+1;**

**M <<= 2;**

**if( tmp >= ttp )**

**{**

**tmp -= ttp;**

**N ++;**

**}**

**}**

**return N;**

**}**

**void FFT() //傅叶函数**

**{**

**register unsigned char i,bb,j,k,p;**

**register short TR,TI,temp;**

**unsigned long ulReal,ulImage;**

**unsigned char max;**

**for(i=0;i<SAMPLE\_NUM;i++)**

**{**

**FftReal[BRTable[i]] = ADC\_read();**

**FftImage[i] = 0;**

**}**

**for(i=1;i<=LOG;i++)**

**{**

**bb=1;**

**bb <<= (i-1);**

**for(j=0;j<=bb-1;j++)**

**{**

**p=1;**

**p <<= (LOG-i);**

**p = p\*j;**

**for(k=j;k<SAMPLE\_NUM;k=k+2\*bb) //傅叶运算**

**{**

**TR=FftReal[k];TI=FftImage[k];temp=FftReal[k+bb];**

**FftReal[k]=FftReal[k]+((FftReal[k+bb]\*cos\_tabb[p])>>7)+((FftImage[k+bb]\*sin\_tabb[p])>>7);**

**FftImage[k]=FftImage[k] - ((FftReal[k+bb]\*sin\_tabb[p])>>7)+((FftImage[k+bb]\*cos\_tabb[p])>>7);**

**FftReal[k+bb]=TR-((FftReal[k+bb]\*cos\_tabb[p])>>7)-((FftImage[k+bb]\*sin\_tabb[p])>>7);**

**FftImage[k+bb]=TI+((temp\*sin\_tabb[p])>>7)-((FftImage[k+bb]\*cos\_tabb[p])>>7);**

**FftReal[k] >>= 1;**

**FftImage[k] >>= 1;**

**FftReal[k+bb] >>= 1;**

**FftImage[k+bb] >>= 1;**

**}**

**}**

**}**

**for(i=0;i<16;i++)**

**{**

**ulReal = FftReal[i+1];**

**ulReal \*= ulReal;**

**ulImage = FftImage[i+1];**

**ulImage \*= ulImage;**

**result[i] = sqrt\_16(ulReal+ulImage);**

**}**

**while(result[i] == FFT\_OUT\_MIN)**

**{**

**for(i=0;i<16;i++)**

**{**

**result[i] =result[i] +1;**

**}**

**}**

**for(i=0;i<16;i++)**

**{**

**if(result[i] > FFT\_OUT\_MAX)**

**{**

**result[i] = FFT\_OUT\_MAX;**

**if(result[i] > max)**

**max = result[i];**

**}**

**}**

**}**

**void init() //初始化函数**

**{**

**P2M0=0XFF;**

**P2M1=0X00;**

**P3M0=0XFF;**

**P3M1=0X00;**

**P1M0=0XFE;**

**P1M1=0X01;**

**P4SW=0x70;**

**ADC\_init();**

**AUXR |= 0x80; //定时器时钟1T模式**

**TMOD=0x01;**

**TH0=(65535-TIME)/256;**

**TL0=(65535-TIME)%256;**

**EA=1;**

**ET0=1;**

**show\_musicdisplay(); //显示数组内容调用函数**

**TR0=1;**

**}**

**void main() //主函数**

**{**

**init();**

**while(1)**

**{**

**FFT();**

**}**

**}**

**void timer0() interrupt 1 //定时子程序**

**{**

**unsigned char i;**

**t++;**

**if(t == 16)**

**{**

**for(i=0;i<16;i++)**

**temp[i] = result[i];**

**t = 0;**

**}**

**disp();**

**TH0=(65535-TIME)/256;**

**TL0=(65535-TIME)%256;**

**}**