

A2. 分享

2021年7月4日 21:37

分享内容概要：

- 一、自动化拼板中画板精度注意事项；
——在画板时的注意事项，避免在画板时就引入制板误差
- 二、气体喷洒指示灯中静电及射频干扰问题分析；
——分享在静电及射频问题中原因及解决方案的分析过程
- 三、MATLAB在电路分析中的运用；
——引入一种工具来帮助进行电路、算法及数据的分析

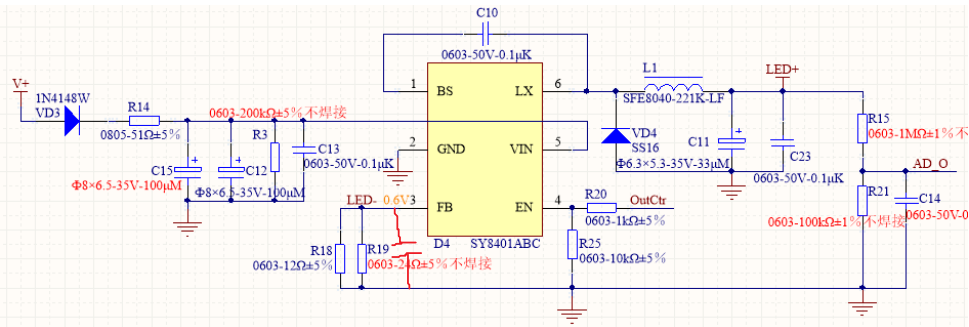
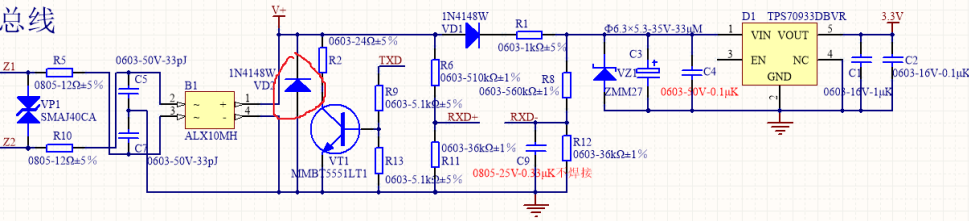
一、自动化拼板中画板精度注意事项：

- 1) 板子外形定义要注意，好的板子外形定义，中心应该落在画线的中间，制板商会以线中间切板。
 - a、老板图升级时注意板子外形；
 - b、绘制板外形的线一定要形成闭合的捕获。
- 2) 板子外形画线宽度5mil, 最大限度减小可能由线宽带来的误差；
- 3) 画板外形时捕捉问题，在【板子参数选项】里面设置；
- 4) 板型相同的PCB，拼板方式应该保持一致，包括工艺边宽度、连接筋位置。



二、气体喷洒指示灯中静电及射频干扰问题分析；

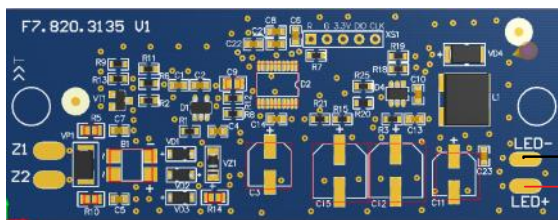
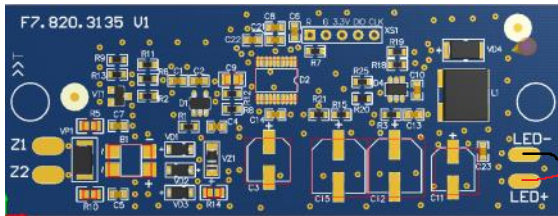
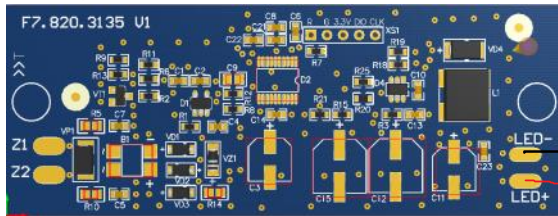
总线



1、静电：
1) 开始做静电试验时，发现一打-8kV，R5或R10就烧坏，VD2(1N4148W)发生短路：
分析原因因为-8kV打入线路板地平面，VD2承受反向-8kV电压而击穿短路，从而总线电流加大烧坏电阻。
解决方案：VD2更换为反向耐压更高的1N4007后解决。
1N4148W--反向耐压75V；
1N4007--反向耐压1KV。

2) 多次静电后D4(SY8401)发生损坏（引脚FB对引脚GND阻抗变小，从基本无穷大到具有小阻值，最后烧坏）：
分析原因因为静电通过灯板打入LED-，从而打坏芯片，在FB引脚增加0.1uF对GND的电容后解决。

2、射频干扰:



- 1) 在不通电的情况下射频干扰能点亮灯板:
 - 2种情况:
 - 1) 干扰导致D4打开, 电容暂存的电使灯板亮;
 - 2) 磁生电, 产生感应电流使灯板点亮。
- 对讲机测试复现, 发现能持续点亮灯板, 如果是情况1)的话应该只亮1下, 电放完后不会亮了, 所以情况2)的可能性较大, 后续的测试也能确定就是情况2)。
- LED-和LED+之间加10nf/1nf, 甚至直接短接LED-和LED+均无任何改善, 这就充分说明了电流的产生与线路板基本无关系, 而是灯板自身环路产生了电流。

- 1) 减小环路面积:
采用双绞线, 减小环路面积:
第1部分采用双绞线后打第1部分不再亮, 但打第2部分还会亮; 第2部分也弄成双绞线后不亮。

- 1) 减小环路面积:
灯板加10nf电容
- 2) 为什么加电阻基本没效果

只要穿过**闭合电路**的磁通量发生变化, 闭合电路中就会产生**感应电流**。这种利用磁场产生电流的现象称为**电磁感应**, 产生的电流叫做**感应电流**。

电磁感应现象的产生条件有两点(缺一不可)。

- 1 闭合电路。
- 1 穿过闭合电路的磁通量发生变化。

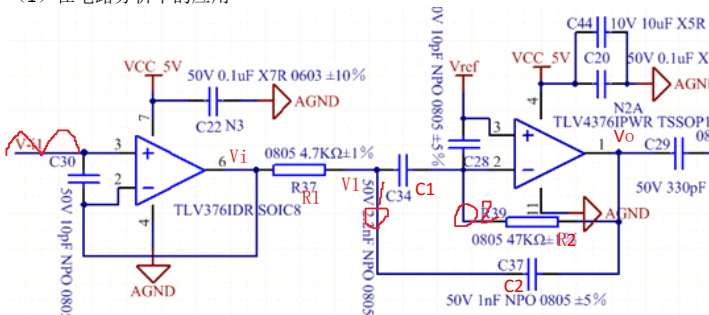
一、感应电动势的大小计算公式

(1) $E=n\Delta\Phi/\Delta t$ (普适公式) {法拉第电磁感应定律, E : 感应电动势(V), n : 感应线圈匝数, $\Delta\Phi/\Delta t$: 磁通量的变化率}。

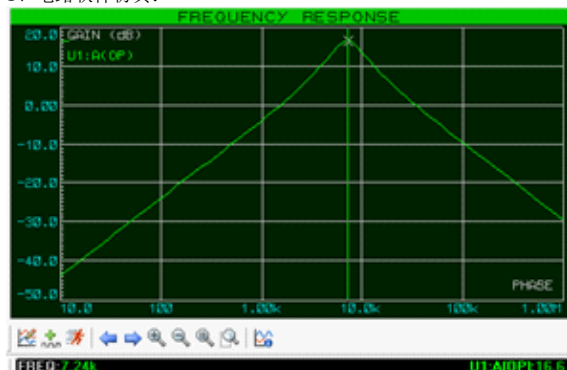
二、磁通量 $\Phi=BS$, 其中 Φ : 磁通量(Wb), B : 匀强磁场的磁感应强度(T), S : 正对面积(m^2) 计算公式 $\Delta\Phi=\Phi_1-\Phi_2$, $\Delta\Phi=B\Delta S=BLV\Delta t$ 。

三、MATLAB在电路分析中的运用。

(1) 在电路分析中的应用



1、电路软件仿真:



2、Matlab分析过程:

```
S1 = (Vi-V1)/R1 - s*C2*(V1-Vo) - s*C1*(V1-Vref); %节点1的KCL
S2 = s*C1*(V1-Vref) - (Vref-Vo)/R2; %节点2的KCL
```

```
V1 = solve(S2, V1); %解V1
S = subs(S1); %带入V1, 更新方程式S1得方程S
```

```
Vo = solve(S, Vo); %解Vo
Vo = simple(Vo); %化简Vo
```

得到,

$$Vo = -\left(\frac{C1 * R2 * s}{C1 * C2 * R1 * R2 * s^2 + (C1 * R1 + C2 * R1) * s + 1}\right) * (Vi - Vref) + Vref$$

```
Hs = -C1*R2*s/(C1*R1*s + C2*R1*s + C1*C2*R1*R2*s^2 + 1); %由Vo得传递函数Hs
```

%带入数据求幅频特性

```
R1 =4700; R2 =47000; C1 =2.2*10^-9; C2 =10^-9;
```

```
f = 1:1:100000;
```

```
s = j*2*pi*f;
```

```
Hf = subs(Hs);
```

```
Hf_w = 20*log10(abs(Hf));
```

```
semilogx(f, Hf_w); % x坐标轴是对数坐标系
```

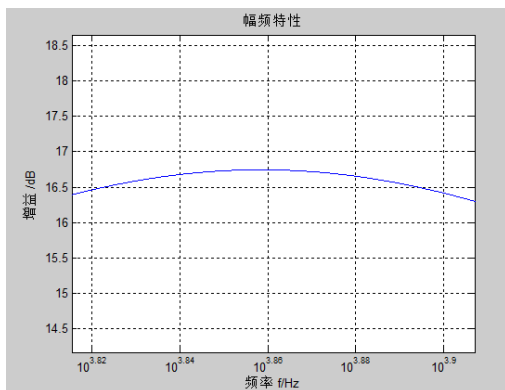
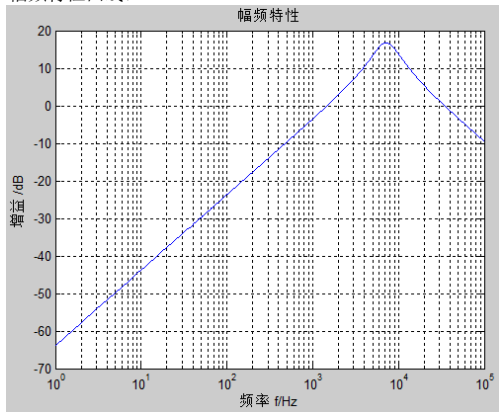
```
grid on;
```

```
xlabel(' 频率 f/Hz');
```

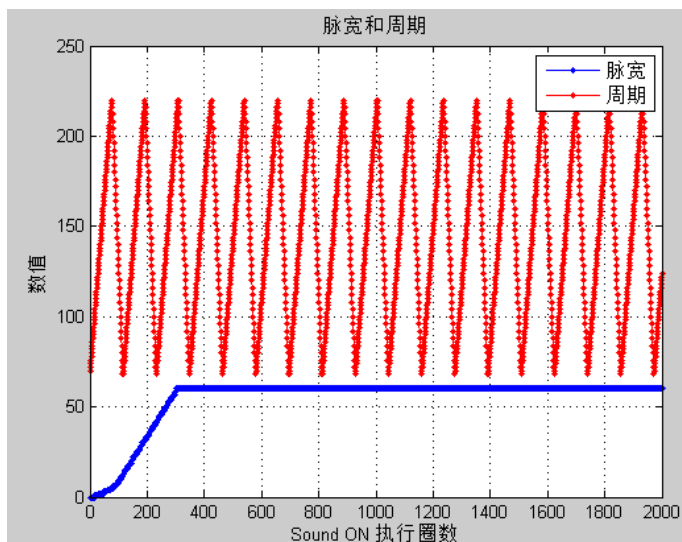
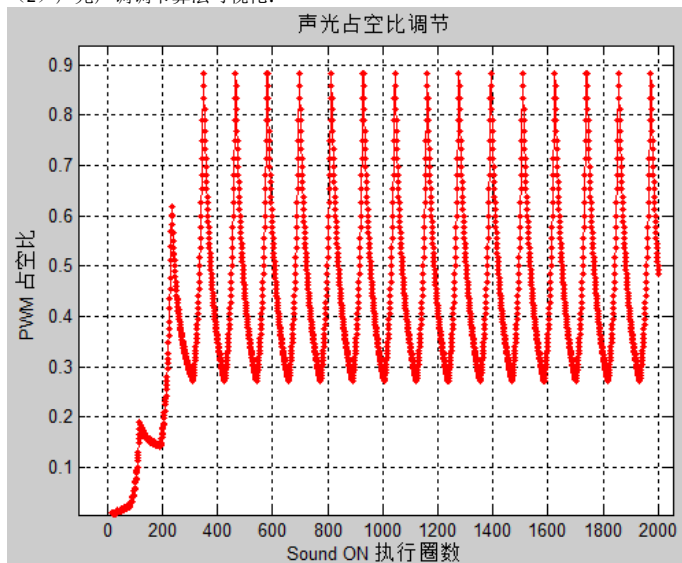
```
ylabel(' 增益 /dB');
```

```
title(' 幅频特性')
```

幅频特性曲线:



(2) 声光声调调节算法可视化:



3、中心频率推导

$$H(j\omega) = \frac{-C_1 R_2 j\omega}{C_1 C_2 R_1 R_2 (j\omega)^2 + (C_1 R_1 + C_2 R_2) j\omega + 1}$$

$$= \frac{-1}{C_2 R_1 j\omega - \frac{1}{C_1 R_2 \omega} j + \frac{C_1 R_1 + C_2 R_2}{C_1 R_2}}$$

$$= \frac{-1}{(C_2 R_1 \omega - \frac{1}{C_1 R_2 \omega}) j + \frac{C_1 R_1 + C_2 R_2}{C_1 R_2}}$$

$$\Rightarrow |H(j\omega)| = \frac{1}{\sqrt{(C_2 R_1 \omega - \frac{1}{C_1 R_2 \omega})^2 + (\frac{C_1 R_1 + C_2 R_2}{C_1 R_2})^2}}$$

$$\Rightarrow \text{当 } C_2 R_1 \omega = \frac{1}{C_1 R_2 \omega} \text{ 时 } |H(j\omega)| \text{ 最大}$$

即中心频率点为

$$\omega_0 = \frac{1}{\sqrt{C_1 C_2 R_1 R_2}}$$

while(round_cnt<2000)

%% 声光声调控制程序算法

% Begin

if((ZKB_count < ZKB_max))

ZKB_TIME_count = ZKB_TIME_count + 1;

end

if ZKB_TIME_count >= 4

ZKB_TIME_count_count = ZKB_TIME_count_count+1;

ZKB_TIME_count = 0;

end

if (ZKB_count < 3)

ZKB_Change_Time = 4;

elseif ZKB_count < 5

ZKB_Change_Time = 3;

elseif ZKB_count < 8

ZKB_Change_Time = 2;

else

ZKB_Change_Time = 1;

end

if ZKB_TIME_count_count >= ZKB_Change_Time

ZKB_TIME_count_count = 0;

ZKB_count = ZKB_count + 1;

end

%每16ms定时进行1次升调

if sound_INC_DEC_flg == 0

% 升调

if sound_PRD >= PRD_MAX

sound_INC_DEC_flg = 1;

% 升到最大后再准备减调

else

sound_PRD = sound_PRD + 1;

PRDBL=(sound_PRD*2);

DTBL=ZKB_count;

end

%每8ms定时进行1次声调

else

% 降调

if sound_PRD <= PRD_MIN

sound_INC_DEC_flg = 0;

else

sound_PRD = sound_PRD - 2;

% hzj:低电平时间降快点

PRDBL=(sound_PRD*2);

% 只有8位 小心溢出

DTBL=ZKB_count;

end

end



tx3307_v10

```

% End

%% 声调控制过程可视化
round_cnt = round_cnt + 1;          % Sound_On执行次数
H_array(round_cnt) = ZKB_count;      % 脉宽
PRD_array(round_cnt) = sound_PRD*2; % 周期
ZKB = ZKB_count/(sound_PRD*2);      % 占空比
ZKB_array = [ZKB_array ZKB];
PWM = [ones(1,ZKB_count) zeros(1,sound_PRD)];
end

```