

# 1 项目简介

简易数字示波器项目包括单片机电路的设计与开发、信号调理电路的计算、人机交互的设计以及外壳模型的设计等内容，结合了模拟电路、单片机设计、电路与PCB设计等综合知识。作为一款入门级数字示波器项目，采用核心板搭配底板设计，适用于电子入门实训以及单片机入门学习。



## 1.1 项目特点

- 采用核心板加扩展板设计理念，采用插件器件；
- 核心板选用国产GD32为主控，同时兼容同类型其他款式开发板；
- 项目综合程度高，实用性强，设计完成后可作为桌面日常仪表使用；
- 项目学习资料丰富，包括电路设计教学、PCB设计以及代码编程的学习。

## 2 电路原理

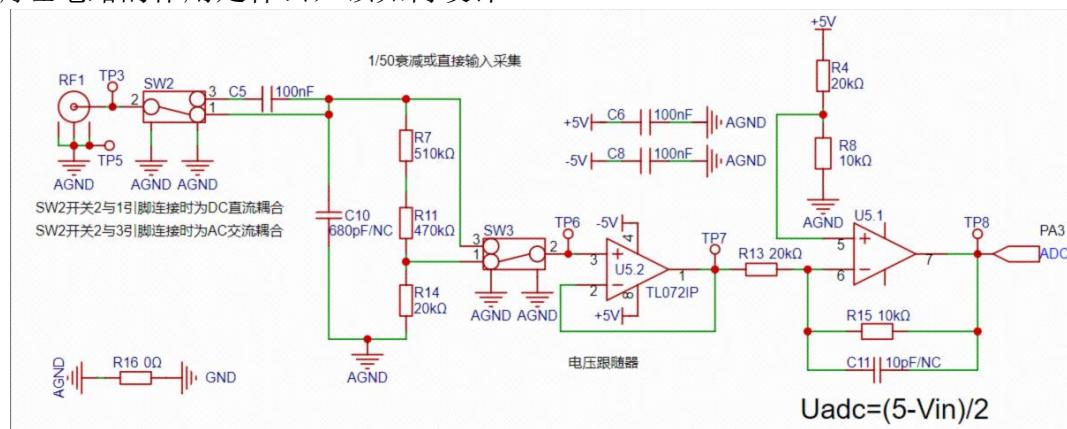
数字示波器是一种用于显示电信号波形的仪器，主要由模拟前端处理电路、单片机电路、电源电路、控制电路、触发电路、校准电路等电路组成。由于该项目为示波器入门项目，在电路设计上选择了一些核心电路，帮助初学者更好地了解示波器的原理和设计方法，主要包括了以下电路：

- (1) **模拟前端处理电路**：负责将输入的检测模拟信号进行处理后给单片机进行识别，具体电路包括了交直流耦合选择电路、电压衰减电路、信号处理电路以及频率检测电路，是电路的核心；
- (2) **电源电路**：负责给运放提供正负电源以及系统供电，是保障电路正常运行的基础；
- (3) **单片机电路**：给系统提供控制核心，负责对输入信号的采集与处理输出工作；
- (4) **人机交互电路**：用于控制示波器功能，包括按键、旋钮、LED灯、显示屏以及其他输入输出接口，为示波器功能的开发提供基础；

## 2.1 模拟前端处理电路

在整个示波器电路设计过程中，模拟前端处理电路是最为重要的，其中大量应用了模拟电路的知识，其中包括输入交直流耦合切换电路、输入信号衰减电路，以及信号调理电路，在讲解模拟输入采集电路前，我们来思考以下几个问题：

- 为什么要加这个电路，直接将输入信号接到单片机引脚不行吗？
- 输入信号处为什么要加一个电容，有什么作用？
- 信号调理电路的作用是什么，该如何设计？



## 2.1.1 交直流耦合切换电路

信号类型可以分为直流信号和交流信号，现实中的信号往往都是都不是理想波形。比如直流电源信号应该是一条水平的直流信号，但都会存在电源纹波（交流信号）；在采集交流信号时也可能混入直流信号对波形的峰峰值造成影响。为了保障对输入交流信号的准确测量，利用电容通交隔直的特性，将电容串联到电路中就可以过滤到信号中的直流分量，这就是交流耦合的概念。而直流耦合就是不对输入信号做任何处理。

举个例子：一个电压为1V的直流电平上叠加了一个1Vpp的正弦波信号，如果使用直流耦合模式，测试看到的是一个向上抬升1V的正弦波信号，但如果使用交流耦合模式，直流信号被滤除，则输出正弦波基准电压为0V。

电路中通过一个拨动开关SW2对输入交直流耦合信号进行切换，当开关2与1接到一起时为直流耦合，当开关2接到3时为交流耦合。那么问题来了，这里的电容该如何取值呢？是不是任意一个电容都行？

表1 不同容量瓷片电容的自谐振频率

电容值	自谐振频率 (MHz)	电容值	自谐振频率 (MHz)
1uF	1.7	820pF	38.5
0.1uF	4	680pF	42.5
0.01u	12.6	560pF	45
3300pF	19.3	470pF	49
1800pF	25.5	390pF	54
1100pF	33	330pF	60

交直流耦合切换电路

瓷片电容自谐振频率表 (源于网络, 仅供参考)

要解决这个问题，那就需要了解电容的频率特性，理想状态下隔直电容的选择应该越大越好，但是由于不同容值的自谐振频率不同，低于自谐振频率时电容呈现容性状态，高于该频率时存在感性状态。电容越大，其自谐振频率越低，简单来说就是大电容通低频，小电容通高频。如上右图可知当瓷片电容为0.1uF (100nF) 时，其自谐振频率为4Mhz，一般要求电容的截止频率 $f_c=0.2 \cdot f_0$ ，其中 $f_0$ 为电路中工作频率。由此可知在该项目中100nF是足够应用的，但如果输入信号频率更高，就应该选择更小的电容。

## 2.1.2 输入信号衰减电路

信号经过交直流耦合选择电路后由开关SW3选择两个通道，开关2和3接到一起时，输入信号直接流入后级的电压跟随器电路；当开关2和1接到一起时，输入信号经过R7、R11、R14三个电阻构成的电阻分压网络后将信号衰减到了1/50倍，即  $\frac{20k}{510k+470k+20k} = \frac{1}{50}$  结合后面的信号调理电路计算可知：

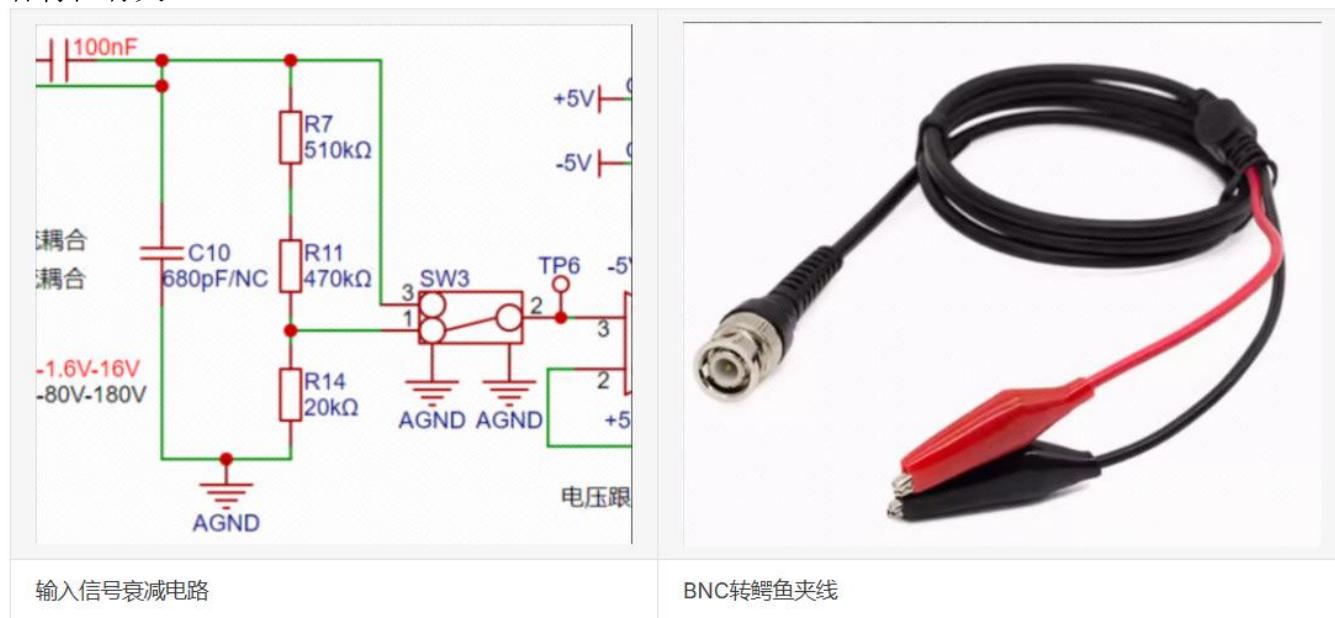
- 当SW2开关2和3接到一起时，可以测量的输入信号幅值为-1.6V~5V；
- 当SW2开关2和1接到一起时，可以测量的输入信号幅值为-80V~250V；

由此可知，当输入信号幅值较小时，可优先选择低压档位，如果测量时不确定输入信号幅值可先用高压档位测量后如满足低压范围内，可用低压档位测量以得到更为精准的测量结果。

### 补充说明

该项目中使用的探头为BNC转鳄鱼夹探头，非专业示波器探头，仅对电路进行简单分析，此处不进行展开说明更多关于探头选择与阻抗说明。如使用专业示波器的无源探头进行测量，由于探头上有一个x1和x10的档位选择，当档位为x10时可以通过调整探头上的补偿电容或者是

电路中的C10进行匹配已得到准确的测量效果，其中C10一般为可调电容，实际大小与各板间电容特性有关。



### 2.1.3 信号调理电路

在信号调理电路中包含了一个电压跟随器以及由运放构成的信号放大电路，在分析该这部分电路时需要掌握运放的虚断与虚短原理。

#### 虚断

理想运放的输入阻抗是无穷大，然而真实的运放输入阻抗却是有限的。如果给运放的输入端加一个电压，然后测量该输入端的电流会发现电流读数接近为0，感觉运放内部断开，没有电流流入一样，但实际又是连接的，这种现象称为虚断。也可以用欧姆定律 $U=I \cdot R$ 来理解，当电压一定时，电流与电阻成反比，电阻无限大那电流也就无限小接近为0。

#### 虚短

在运放处于深度负反馈时会出现虚短现象，使两个输入端的电位相等，就好像两个输入端短接到一起，可以近似为 $V_+ = V_-$ 。在负反馈中，运放的输出信号的一部分被取出并反馈到输入端。这种反馈作用使得运放的两个输入端（正输入和负输入）的电压差趋近于零，两个输入端的电压几乎相等。因为尽管运放的两个输入端在电气上并没有被直接短路，但由于负反馈的作用，两个输入端的电压却几乎相等，就好像它们被短路了一样，故称为虚短。

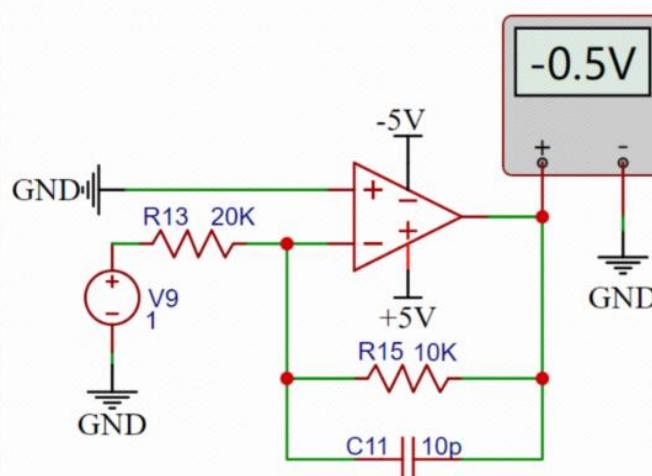
接下来我们用虚短和虚断的知识分析以下电路：

#### (1) 电压跟随器电路

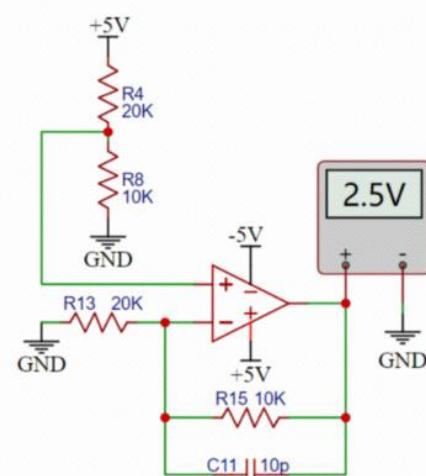
在电路U5.2芯片中运放反向输入引脚2脚接到运放输出引脚1脚，结合运放的虚短特性， $V_+ = V_- = V_{out}$ 。根据虚断可知运算放大器的输入阻抗比较大，所以运算放大器正向输入电流很小，运放输出阻抗小所以输出的电流很大，说这里的电压跟随器起着一个阻抗匹配的作用。

#### (2) 比例放大电路

在对U5.1运放构成的电路进行分析时，可以将其拆解为一个同相比例放大电路和一个反相比例放大电路进行单独分析后合到一起。



反相比例放大电路



同相比例放大电路

假设运放正向输入端接地，构成反相比例放大电路，由运放的虚断特性可知反向输入引脚流入运放的电流为0，可以R13和R15可以看做串联，所以流经电流相同。由虚短特性可知 $V_+ = V_- = 0$ ，由此可得：

$$\frac{V_i - 0}{R_{13}} = \frac{0 - V_o}{R_{15}}, \quad V_o = -\frac{R_{15}}{R_{13}} * V_i$$

假设运放反向输入端接地，构成正相比例放大电路，由运放的虚断特性可知运算放大器两个输入引脚流过的电流为0，即流过R4的电流为0， $V_{in} = V_+$ 。可以看做R13和R15串联，所以流过的电流也相同。由虚短特性可知两个输入引脚的电压相同，即 $V_+ = V_-$ ，所以有 $V_{in} = V_+ = V_-$ 。 $V_o$ 与R13和R15构成回路，则流过的电流为：

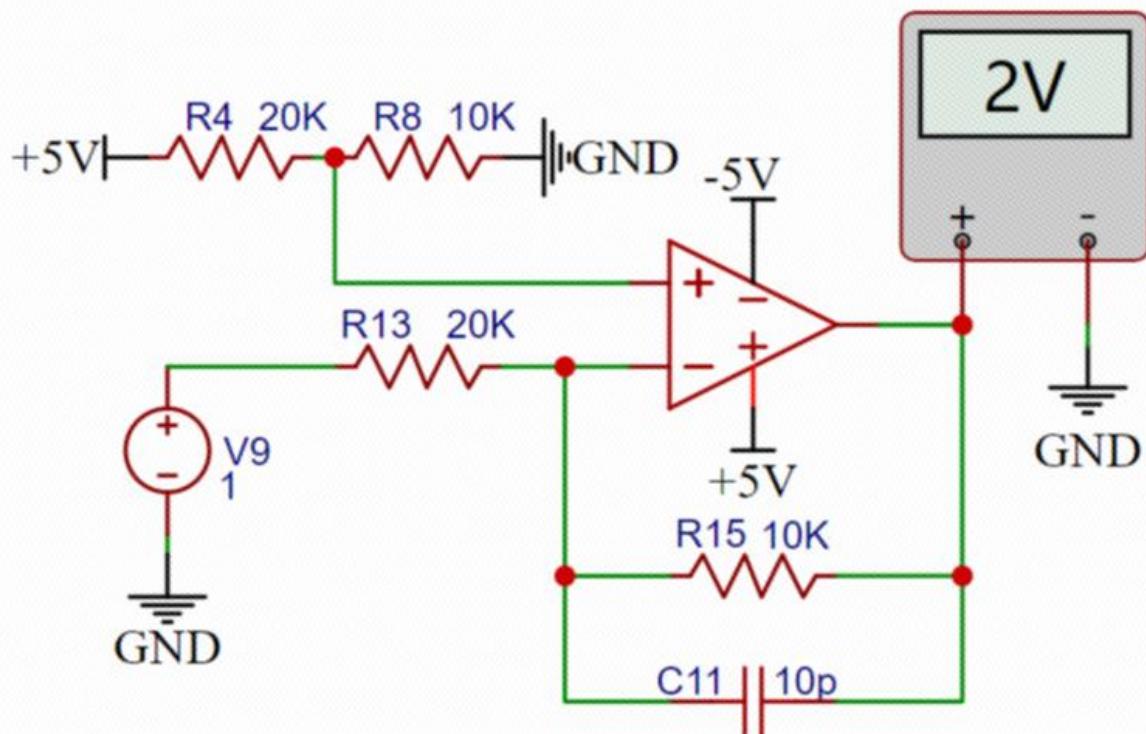
$$I = \frac{V_o}{R_{13} + R_{15}}$$

单独对R13来看，流过的电流也等于它两端的电压除以它的电阻值，代入 $V_{in} = V_+ = V_-$ 得到以下公式：

$$I = \frac{V_-}{R_{13}} = \frac{V_+}{R_{13}} = \frac{V_{in}}{R_{13}}$$

$$\text{合并两公式可得出: } V_o = V_{in} * (1 + \frac{R_{15}}{R_{13}})$$

将以上两个公式合到一起，将 $V_{in} = \frac{5}{3}V$ ,  $R_{13} = 20K$ ,  $R_{15} = 10K$ 带入得 $V_o = 2.5V$ ，与仿真结果一致。



将以上同相比例放大电路与反相比例放大电路结合起来得到以下公式  $V_o = (5 - V_{in})/2$ , 将  $V_{in} = 1$  代入公式得  $V_o = 2V$ , 与仿真结果保持一致。

$V_o$  信号将直接接到单片机的 ADC 引脚, 由于单片机采集到 ADC 的电压范围值为 0~3.3V, 由此可以计算出该示波器输入电压范围。

当输入信号  $V_{in}$  不衰减时, 将  $V_o = 0$ 、 $V_o = 3.3V$  分别代入公式得:

$$0 = \frac{5 - V_{in1}}{2}, V_{in1} = 5V \quad 3.3 = \frac{5 - V_{in2}}{2}, V_{in2} = -1.6V$$

当输入信号  $V_{in}$  衰减  $\frac{1}{50}$  时, 将  $V_o = 0$ 、 $V_o = 3.3V$  分别代入公式得:

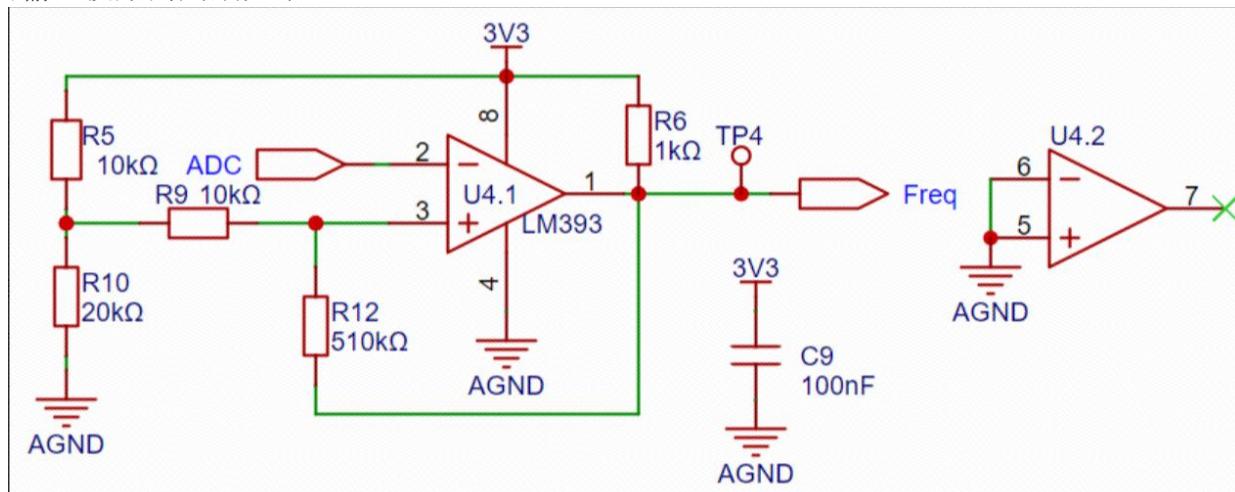
$$0 = \frac{5 - 1/50 V_{in1}}{2}, V_{in1} = 250V \quad 3.3 = \frac{5 - 1/50 V_{in2}}{2}, V_{in2} = -80V$$

得到以下结论:

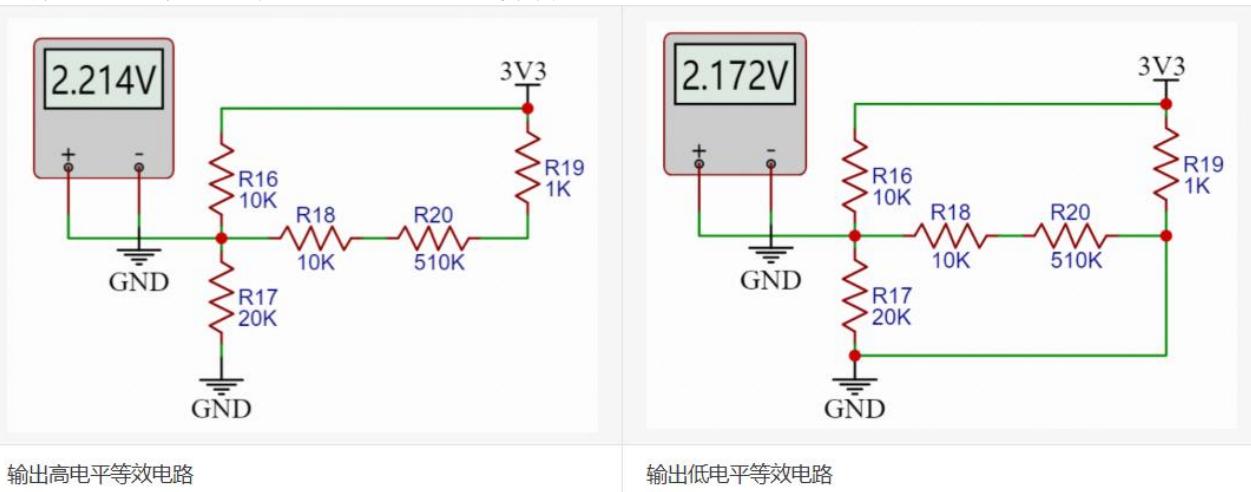
低压档位测量范围: -1.6V~5V, 高压档位测量范围: -80V~250V

## 2.1.4 比较器测频电路

为了实现频率检测的功能, 将 ADC 输入信号通过一个滞回比较器对输入信号进行比较, 实现频率的测量功能。滞回比较器是属于电压比较器中的一种, 常规的电压比较器是一个单限比较器, 电路中只有一个阈值电压, 但在输入电压在阈值附近有微小变化时都会引起输出电压的跃变。为了增强电路的抗干扰能力, 在单限比较器的基础上引入了正反馈, 保障了在一定范围内信号的稳定性。通过滞回比较器电路后输出一个方波信号, 使用单片机的定时器捕获功能计算出输入波形的周期大小。



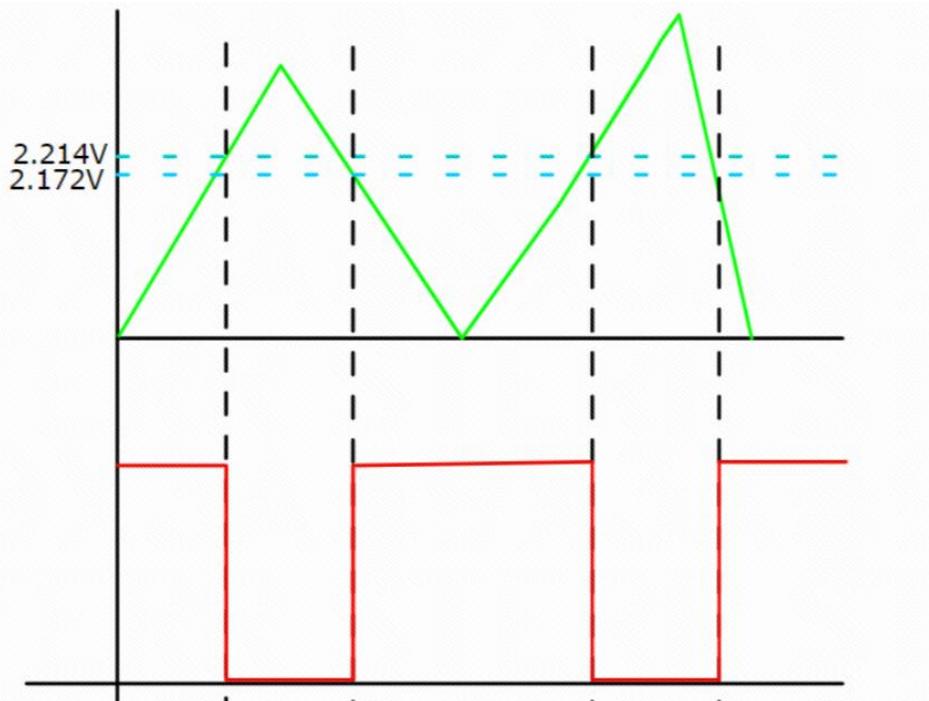
滞回比较器电路的阈值电压需单独对运放输出结果进行分析, 当输出为高电平时, 输出端上拉到高电平, 这时等效电路如下左图所示, 算得  $U_{th} = U_+ = 2.214V$ 。当运放输出为低电平时, 输出端接地, 等效电路如下右图所示, 算得  $U_{tl} = U_- = 2.172V$ 。



输出高电平等效电路

输出低电平等效电路

下图绿色线代表信号输入电压变化情况，从0电位上升，初始输出状态为高电平，输入电压达到2.214V时，输出信号变成低电平，直到输入信号低于下限阈值2.172V时输出变为高电平。可以根据比较器当前的输出状态来确定下一个变化电平的阈值，当输出为高电平时使用的是高阈值 $U_{th}$ ，输出为低电平时使用低阈值 $T_{tl}$ 。之所以将阈值设置接近是为了避免信号干扰造成的误识别。

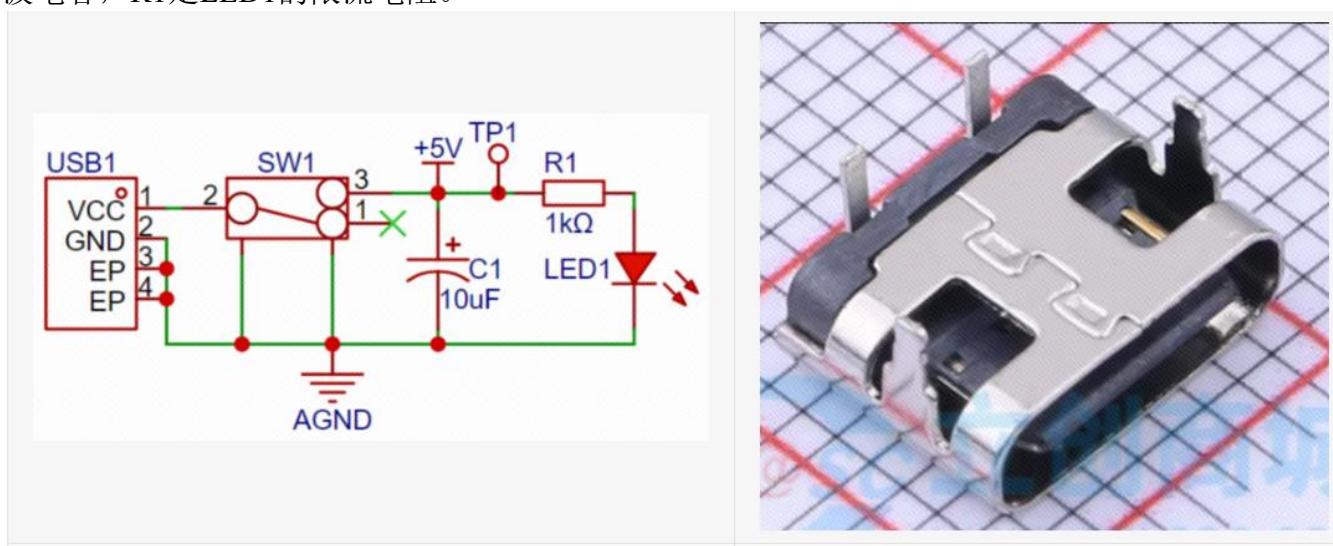


### 注意事项

此处阈值比较器运放正向输入信号为固定电平，若使用带有DAC输出的单片机，可自由配置该点电位大小，从而改变阈值电压，实现对触发模式的设置。

## 2.2 电源控制电路

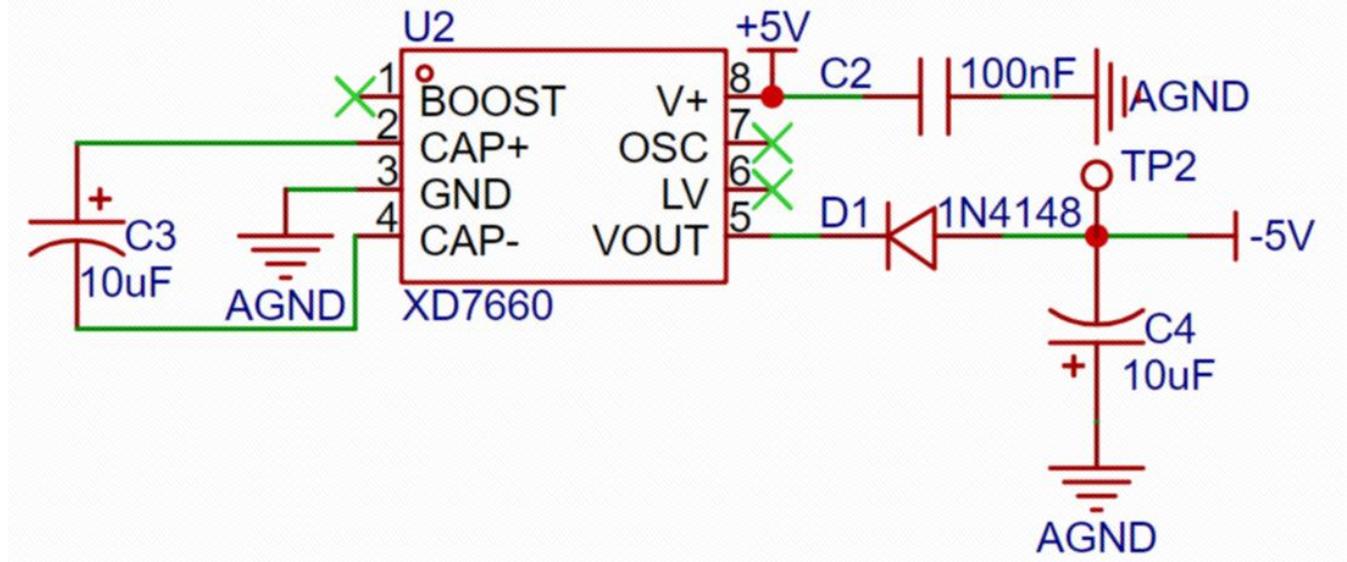
该项目使用GD32最小系统板为核心，板载了5V转3.3V降压电路，所以在设计扩展板时只需要设计一个5V电源输入电路即可，这里选用了主流的Type-C为输入接口，且该接口只有两根线，插件封装，方便新手焊接学习，但需要注意的是这个Type-C接口仅用于供电，不能传输数据，如果需要使用传输数据，可以使用核心板上的Type-C接口。SW1为电源总开关、C1为输入滤波电容，R1是LED1的限流电阻。



Type-C电源输入电路

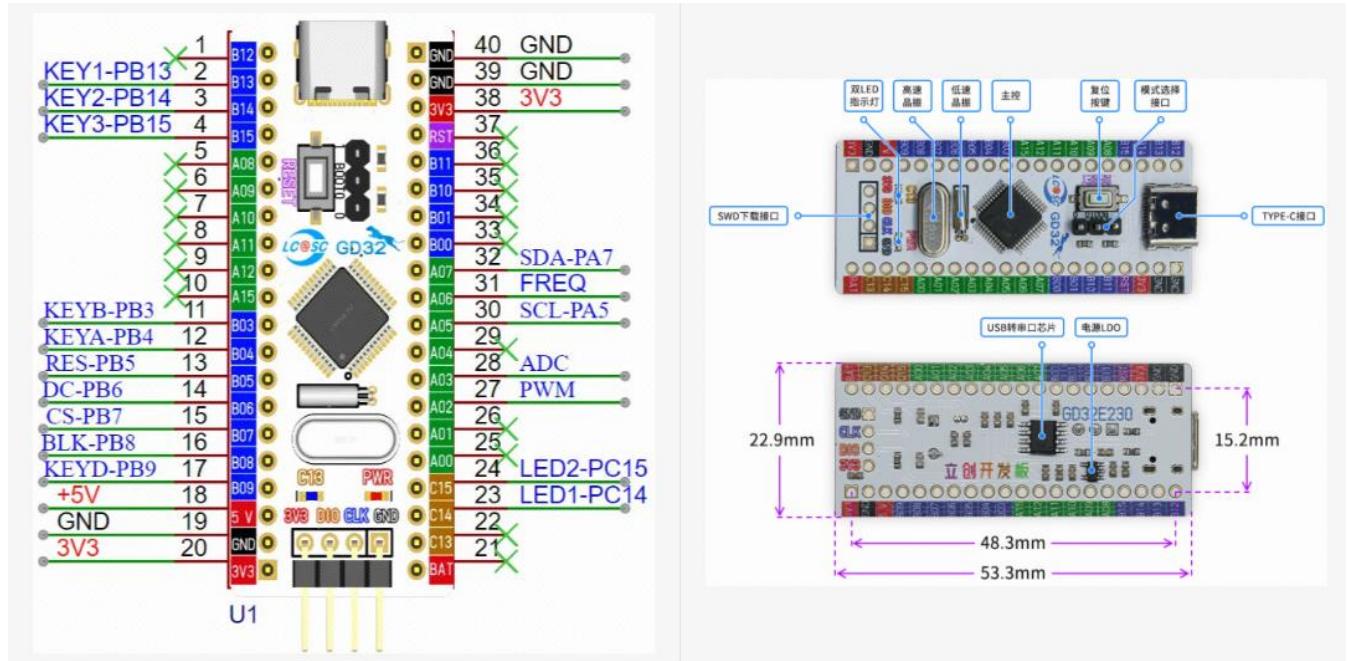
两脚Type-C实物图

除了电源输入电路外，为保障运算放大器对于负电压的测量性能，使用了XD7660负压产生电路得到一个负电压，该芯片外围电路简单，只需要两个电容和一个二极管即可工作，理论上输入电压为+5V，也可以输出一个-5V的电压，由于芯片内部存在一定压降及转换效率，实际测量负电压为-4.3V左右，也能满足运算放大器的要求。



## 2.3 单片机电路

该项目使用了由立创开发板团队推出的GD32最小系统板为主控，这款开发板是由立创开发板团队联合兆易创新推出的一款全国产的开发板，板载CH340下载芯片，只需要一根数据线就可以对板子进行烧录与串口调试，同时兼容STM32最小系统板的尺寸与引脚配置，可以直接进行替换。



## 核心板电路引脚分配图

## 核心板实物图

在配置核心板引脚与单片机功能时需结合引脚特性，比如SPI液晶屏驱动需要连接到对应的SPI引脚，ADC检测需要接到到ADC功能的引脚上，这里的引脚配置不唯一，可重新配置其他的方案，以数据手册中的引脚功能说明为准。

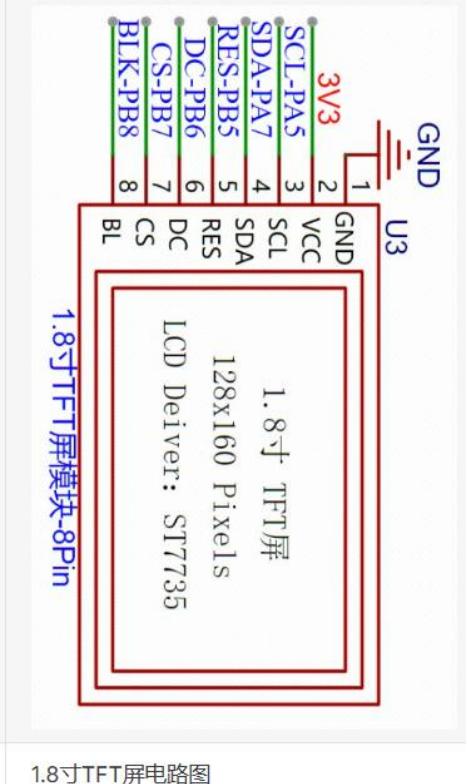
## 2.4 人机交互电路

### 2.4.1 液晶屏显示电路

1.8 TFT 是一款彩色显示屏，具有  $128 \times 160$  个彩色像素，使用四线SPI通信方式与单片机进行连接，一共有八根引脚，模块引脚说明及与单片机连接情况如下所示：

引脚序号	引脚名称	引脚说明
1	GND	电源接地引脚
2	VCC	接电源VCC引脚
3	SCL	串行时钟信号
4	SDA	SPI数据引脚
5	RES	SPI复位引脚
6	DC	SPI指令和数据选择引脚
7	CS	LCD片选信号
8	BL	背光控制引脚

1.8寸TFT液晶屏引脚说明图



1.8寸TFT屏电路图

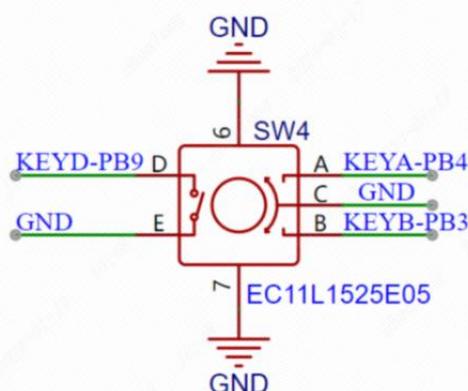
### 2.4.2 旋转编码器电路

旋转编码器属于一种特殊的按键，该项目使用的EC11旋转编码器有五个引脚，其中DE两个引脚类似于普通按键引脚，按下导通，松手断开，其余ABC三个引脚用于检测旋钮的转动方向，C脚为公共端，直接接地就行。

在旋转编码器时，A和B两个信号引脚存在相位差，也就是有一个引脚信号变化后另一个引脚信号再跟着变化，即两个引脚不同时变化，通过检测哪个引脚先变就能判断是正转还是反转功能。



EC11实物图及旋钮帽



EC11电路连接图

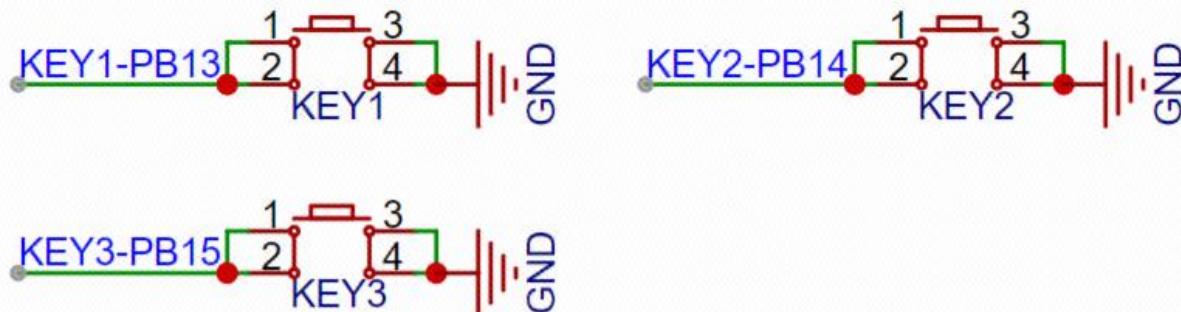
### 2.4.3 LED灯指示电路

LED指示电路设计比较简单，采用低电平驱动的方式，当单片机引脚输出为低电平时，LED两端存在电势差，LED点亮；当单片机引脚输出为高电平时，LED灯熄灭。



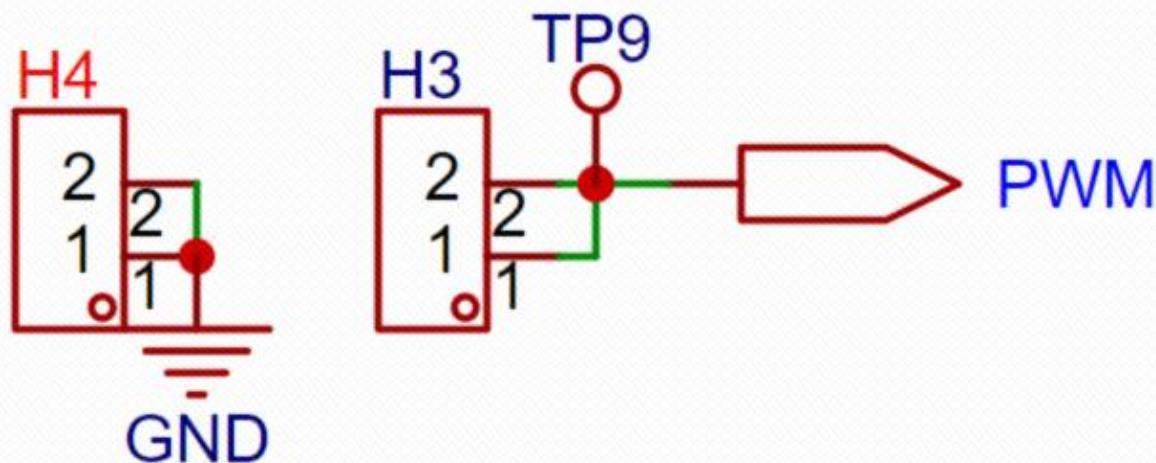
### 2.4.4 按键输入检测电路

除了旋转编码器外，该项目还使用了三个独立按键对系统进行控制，三个按键一侧直接接地，另一侧连接到单片机引脚，当单片机引脚检测到按键按下时，单片机引脚直接接到GND接地，单片机收到该引脚接地信号的反馈后去实现对应的功能，为节约硬件成本，可以在软件设计时引入消抖功能，避免机械按键抖动时的误触发。



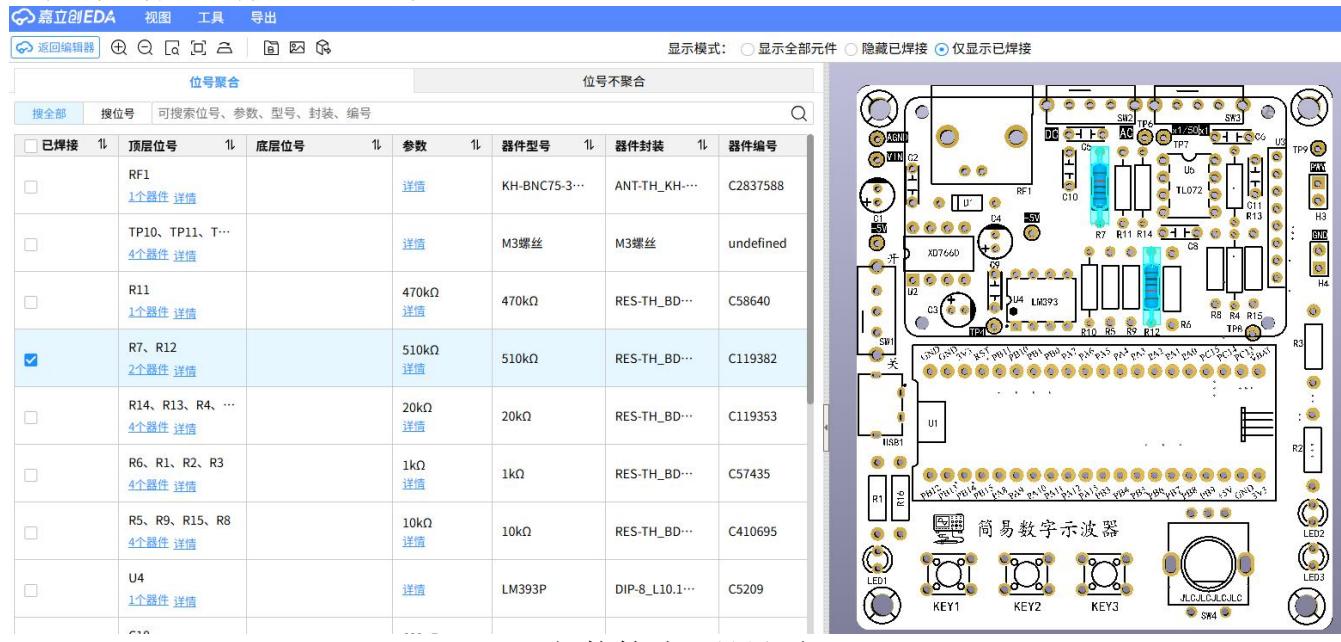
### 2.4.5 其他电路

除了示波器检测功能外，单独引出了一个PWM信号用于模拟一个简易的函数发生器功能，可以通过改变输出PWM的频率和占空比输出一个简易方波信号输出。



### 3 焊接与调试

焊接时可根据物料清单对照板子上的元器件位置进行焊接，下载焊接辅助工具在电脑端或手机端查看元器件位置辅助焊接。



焊接辅助界面

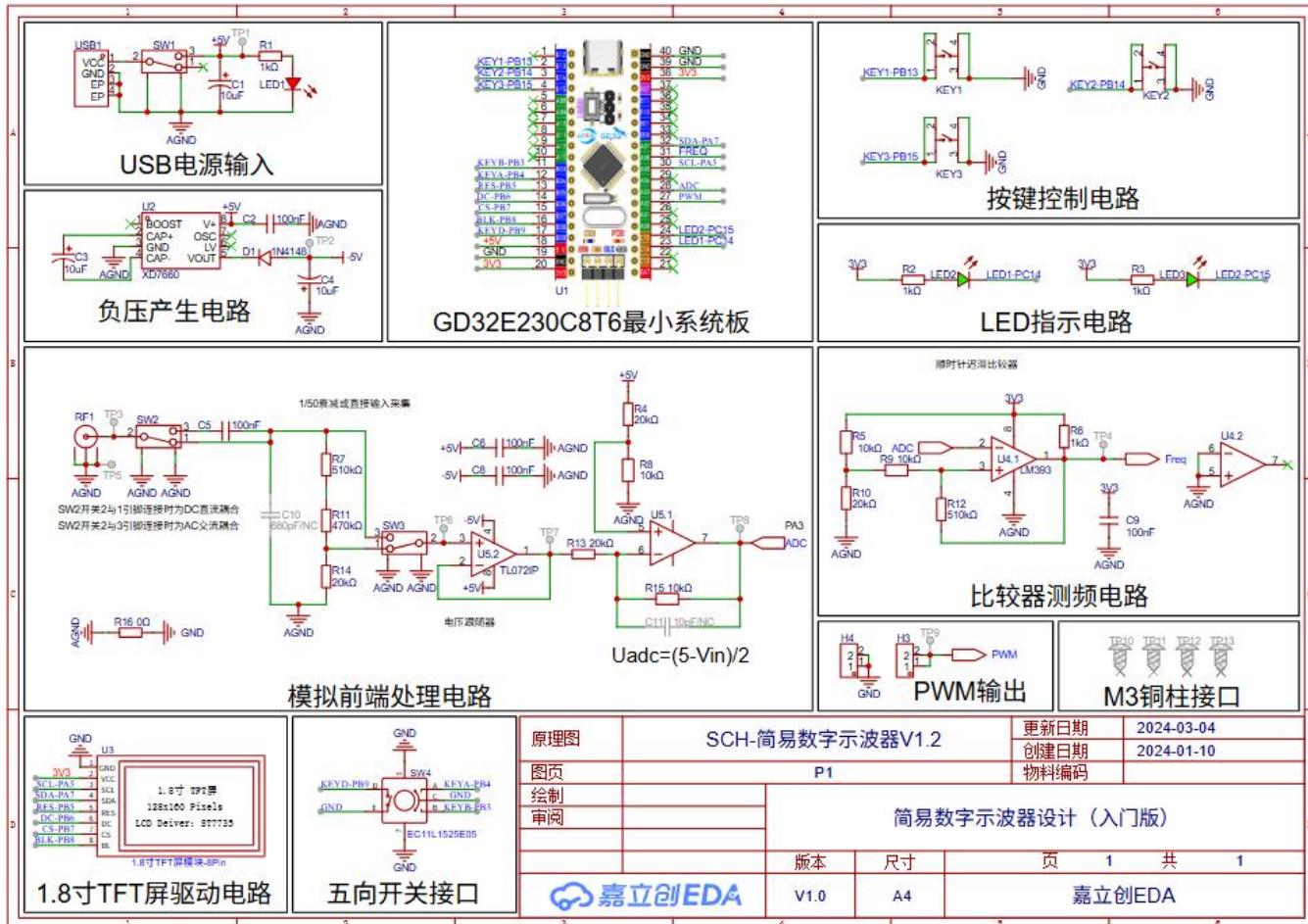
在焊接调试过程中有以下几点注意事项：

- (1) 焊接过程中注意用电安全，手不要接触到烙铁头，免得烫伤
- (2) 焊接过程中先按元器件的低到高进行焊接
- (3) 电容插装焊接垂直安装，独石电容、电解电容紧贴电路板插装焊接。安装电解电容时，要注意电容极性不能接反，否则因电流过大容易引起爆炸。
- (4) 二极管插装焊接水平卧式安装，安装时要贴紧电路板插装焊接，注意分清型号和极性。
- (5) 芯片安装时要注意分清芯片型号和方向，确认无误后再上电。

焊接时参考以下顺序依次焊接，先焊接芯片座再安装芯片，谨防芯片安反，使用万用表检查无短路，电解电容方向焊接正确方可上电，防止接错电容爆炸。

序号	器件名称	参数	位号	数量	备注
1	通用二极管	1N4148	D1	1	黑色色环一侧为负极
2	插件电阻	1k $\Omega$	R1, R2, R3, R6	4	
3	插件电阻	20k $\Omega$	R4, R10, R13, R14	4	
4	插件电阻	10k $\Omega$	R5, R8, R9, R15	4	
5	插件电阻	510k $\Omega$	R7, R12	2	
6	插件电阻	470k $\Omega$	R11	1	
7	插件电阻	0 $\Omega$	R16	1	
8	瓷片电容	100nF	C2, C5, C6, C8, C9	5	
9	瓷片电容	680pF	C10	1	
10	瓷片电容	10pF/NC	C11	1	
11	发光二极管	3mm 红色	LED1	1	长引脚一侧为正极
12	发光二极管	3mm 绿色	LED2, LED3	2	长引脚一侧为正极
13	USB连接器	TYPE-C-2P	USB1	1	
14	比较器芯片	LM393P	U4	1	先焊接DIP8芯片座再装芯片
15	运算放大器	TL072IP	U5	1	先焊接DIP8芯片座再装芯片
16	电源芯片	XD7660	U2	1	先焊接DIP8芯片座再装芯片

17	拨动开关	SK-12E12-G5	SW1, SW2, SW3	3	
18	排针	2.54*2P	H3, H4	2	
19	轻触开关	6x6x7	KEY1, KEY2, KEY3	3	
20	电解电容	10uF	C1, C3, C4	3	有白色边一侧为负极
21	同轴连接器	BNC75	RF1	1	
22	旋转编码器	EC11	SW4	1	
23	主控电路	GD32E230C8T6	U1	1	板子上焊接排母, 核心板插上
24	显示模块	1.8寸TFT	U3	1	板子上焊接排母, 屏幕插上去
25	两个M2的螺丝安装在原理屏幕的一侧, 四个M3铜柱安装在板子四个角落				



## 4 项目资料



[开源工程](#)



[线上文档](#)



[视频教程](#)



[技术支持](#)