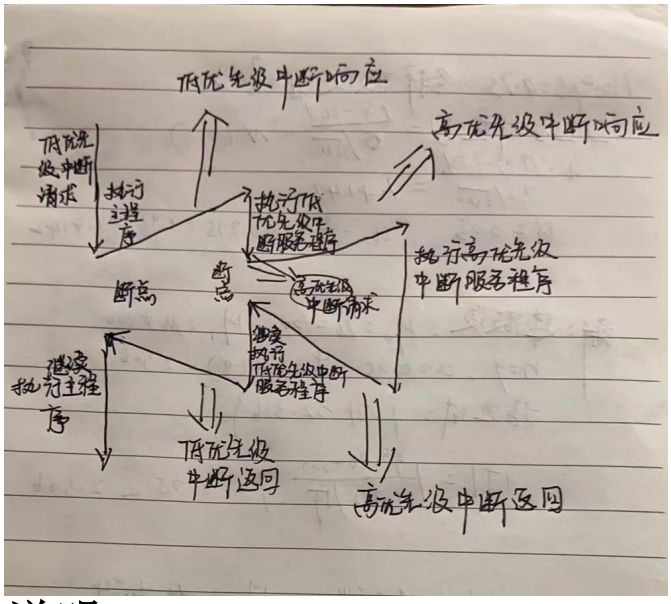
单片机中断

笔记本:	My Notebook		
创建时间:	2023/12/4 21:02	**更新时间:**	2023/12/4 22:58
作者:	u0iwp2ud		
URL:	https://yiyan.baidu.com/		

一、



说明：如果多个中断源同时提出了中断请求，先响应高优先级中断源，后响应低优先级中断源。属于相同优先级的中断源，则根据其内部中断查询顺序，先查询的先响应，后查询的后响应；如果一个中断源提出了中断请求，已经转去执行其中断服务程序了，期间又有一个中断源提出了中断请求，CPU 的处理原则是，如果新的中断优先级与当前正在处理的中断是同级的，则不子响应，待当前中断服务程序执行完成后，再响应。

中断的优势：

1、提高处理器的效率：使用中断可以有效降低CPU对于外设的轮询时间，释放CPU的大量时间用于执行其他任务。
2、提高系统的实时性：当有事件需要立即响应时，通过中断可以使CPU立即处理该事件，并且保证处理完成之后立即返回原来的执行任务，从而提高了系统的实时性。
3、简化程序结构：中断可以将程序分为两部分，一部分是中断服务程序，另一部分是主程序。当有中断发生时，CPU会暂停主程序的执行，转去执行相应的中断服务程序。完成后再返回主程序继续执行。这样可以实现并行处理多个任务，提高系统的灵活性和响应能力。

中断函数与普通函数的区别：

1、入口与出口：中断函数与普通函数在入口和出口上有区别。中断函数在入口处需要屏蔽一些中断，以防止在处理重要工作中又被中断重入，导致处理异常。同时，中断函数还要注意在入口保存重要的寄存器状态，在出口处恢复，防止中断函数结束后给正常程序带来异常。而普通函数则没有这些要求。

2、关联性与服务对象：中断函数一般是和硬件相关联，在一定条件下才跳转进入的函数，而普通函数则主要是为了执行特定的计算或操作任务，两者属于主从关系。主程序需要子程序时就去调用子程序，并把调用结果带回主程序继续执行。 中断服务程序与主程序两者一般是无关的，不存在谁为谁服务的问题，两者是平行关系。

3、嵌套与优先级：普通函数嵌套可实现若干级，嵌套的最多级数由计算机内存开辟的堆栈大小限制。而中断嵌套级数主要由中断优先级数来决定，一般优先级数不会很大。

4、处理方式：主程序调用子程序过程完全属于软件处理过程，不需要专门的硬件电路。而中断处理系统是一个软、硬件结合系统，需要专门的硬件电路才能完成中断处理的过程。

中断的注意事项：

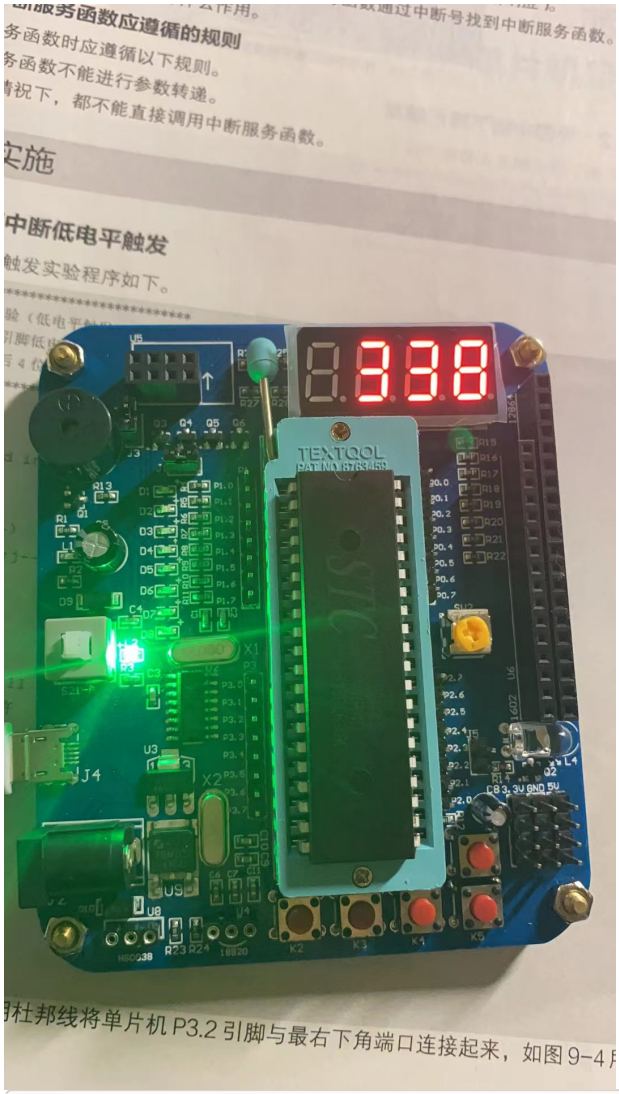
1、理解中断优先级：中断优先级是单片机的核心概念之一，指当多个中断源同时触发时，单片机会优先处理哪个中断源。理解中断优先级的原理和设置方法，以便正确地处理多个中断源的优先级关系。

2、避免优先级反转：在有高优先级中断源的中断服务程序中，不应再调用低优先级中断的中断服务程序，否则将无法保证系统的正常运行。

3、避免重复中断：在中断服务程序中不应再产生相同的中断，否则将导致系统运行混乱。

4、注意寄存器的保存和恢复：在中断服务程序中需要使用到寄存器，因此需要在进入中断服务程序之前保存寄存器的状态，在退出中断服务程序时恢复寄存器的状态，以避免干扰主程序的正常运行。

二、




```
#include<reg52.h>
#define duan P0
/*变量定义*/
sbit wei2 = P2^5;//定义第二位LED显示器
sbit wei3 = P2^6;//定义第三位LED显示器
sbit wei4 = P2^7;//定义第死位LED显示器



//硬件延时函数
void delay(unsigned int xms)
{
        unsigned int i,j;
        for(i=xms;i>0;i--)
                for(j=112;j>0;j--);
}
          void low() interrupt 0
          {
                  delay(10);
                }
//主函数实现LED显示自己的学号
void main()
{
        while(1)
        {
                duan = 0x4f;
                wei1 = 0;
                wei2 = 1;//第二位显示器工作
                wei3 = 0;
                wei4 = 0;
                delay(1);//延时1ms
```

```
                duan = 0x4f;
                wei1 = 0;
                wei2 = 0;
                wei3 = 1;//第三位显示器工作
                wei4 = 0;
                delay(1);//延时1ms
                duan = 0x7f;
                wei1 = 0;
                wei2 = 0;
                wei3 = 0;
                wei4 = 1;//第四位显示器工作
                delay(1);//延时1ms


        }
}
```